

# Polarion 19 User and Administrator Help

POL002 • 19

# **Contents**

Getting Started	1-1
Learn Polarion basics	2-1
Create a user account and log on	2-1
Change password, email, and personal info	2-4
Disable email notifications	2-6
Understand your user permissions	2-6
Navigation and content	2-7
Navigation topics	
Search the Portal	
Create a project	
Project definition and options for creating	
Create a new project from a template	
Access projects	
Open an existing project	
Access a project's documents	
Access Project Work Items	-25
User Guide	2 4
User Guide	<b>3-</b> 1
Home	3-1
About Home pages	
Work with Home pages	
Your personal Home page	
Personalize Navigation	
Dashboard	
Work with the Dashboard	
Repository and Project Group Dashboards	-11
Project dashboard	
Edit and update the Dashboard topic	-13
Plans 3	-14
About Plans	-14
Browse Plans	-15
Review a Plan 3-	-16
Create a new Plan in a project 3-	-17
	-18
Understand Plan actions	-20
	-21
,	-21
I de la companya de	-22
· · · · · · · · · · · · · · · · · · ·	-23
	-26
Work Items overview	-26

Work Item types	. 3-26
Table view	. 3-27
Tree view	. 3-28
Live Plan view	. 3-29
Road Map view	. 3-30
Multi Edit view	
Matrix view	. 3-32
Time Sheet view	. 3-33
Work Item browser and editor panes	. 3-35
Link to another Work Item	. 3-36
Import Work Items	. 3-37
Browse Work Items	. 3-44
Search Work Items	. 3-50
Create and edit Work Items	. 3-55
Review and Approve Work Items	. 3-70
Assign a Work Item	. 3-95
Change a Work Item's status	. 3-96
Prioritize Work Items	. 3-96
Work Item Comments	3-102
Attach files to a Work Item	3-104
Voting	3-105
View Work Item History	3-105
Set and remove Watches	
Set Planning constraints	
Print Work Items	
Clipboard operations	
Link Work Items	3-108
Export Work Items	3-128
Work Item actions	3-134
Move Work Items	3-146
Use Excel Round-trip	3-149
Work Item translations	3-155
Documents and Pages	3-157
Test Runs	3-158
Baselines	3-160
About Baselines	3-160
Create a new Baseline	3-161
Modify an existing Baseline	3-161
Compare Baselines	3-162
Browse, search and report Baselines	3-163
Builds	3-164
Quality and metrics	3-164
About quality and metrics	3-164
Process dashboard	3-166
Work Items dashboard	3-167
Testing dashboard	3-167
CMMI dashboard	3-167
Reports	
About reports	3-168

	Mayingelte	2 460
	Mavin site	
	Unit test coverage	
	Unit tests	
	Javadoc	
	Xref	
	nitor	
Rep	pository Browser	
	Repository Browser overview	
	Repository operations	
Wo	rking with Documents	
	Working with Documents overview	
	Create a new Document	
	Spaces and Subspaces	3-176
	Create a new empty Document	3-185
	Add remote links to a LiveDoc	3-187
	Import Microsoft Word Documents	3-190
	RegIF Import/Export	
	Browse for Documents	
	Search for Documents	
	The Document Editor	
	Find and Replace	
	Copy and Paste	
	Spell check	
	Keyboard shortcuts	
	Work Items in Documents	
	Comments and threads	
	Images and other attachments	
	· · · · · ·	
	Insert a Cross-reference	
	Insert a Diagram	
	Insert or Edit a Formula	
	Insert a Table of Contents	
	Insert a Table of Figures	
	Insert a Table of Tables	
	Insert page breaks	
	Insert Classic Wiki macros and content	3-259
	Resolve conflicts	3-261
	Document Round-trip for Microsoft Word	3-261
	Document Properties, History, and Outline	3-271
	Document actions	
	Merge Work Items from different Documents	3-299
	Sign Documents	
Wo	rking with Pages	
	Working with Pages overview	
	Create new Pages	
	Access existing Pages	
	Delete Pages	
	Edit Pages	
	Page History	
	Export to PDF or Print a Page	
		~ ~~

Download and Reuse Pages	
Build reports	
Polarion for Requirements engineers	3-397
Polarion and requirements management	3-397
Approaches to requirements	3-397
Requirements projects	3-398
Elicitation phase	3-399
Definition phase	
Approval phase	
Common operations	
Advanced topics	
Polarion for project managers	
Planning to start a Project	
Project template for Agile process	
Project Administration	
Manage Project planning	
Dashboards and Reports	
Polarion for developers	
Access Projects	
Locate your Work Items	
Process Work Items	
Builds, tests, source code and documentation	
Polarion for testing and quality assurance	
Testing and quality assurance overview	
Test managment Projects and templates	
Create and manage test specifications	
Create and manage Test Runs	
Use Test Parameters and Iterations	
Integrate external testing tools	
Report testing status and results	
Managing Variants	
Introduction to Variant Management	
Prerequisites	
Terminology	
Process Overview	
Variant Projects and Project Templates	
Define the Feature Model	
Create a Master Specification	3-522
Create Variants and Variant Specifications	3-528
Variant management advanced topics	3-536
Appendix	3-547
Using Classic Wiki	3-547
Microsoft Project import-export	3-565
Deprecated macros	
Administrator's guide	4-1
Administration basics	4_1
Administration rights	
Administration interface	

Resolve problems and troubleshooting	. 4-7
Setup for Internet Explorer users	
Administration topic overview	. 4-8
Manage users and permissions	. 4-9
Overview - User Management	. 4-9
Users and Projects	4-10
Create user accounts	4-10
Configure user roles	4-15
Configure user permissions	4-18
Configure user Time-splitting	4-26
Integrating Polarion server with LDAP/Active Directory	4-27
Network-authenticated Users (SSO)	4-31
Configure the User Accounts Vault	
Create and manage projects	
Create a new project	
Mark an existing repository folder as a Project	
Modify existing Project properties	
Add and Remove Projects from Polarion management	
Customize Project templates	
Use external repositories (GIT, SVN)	
Configure Work Items	
Customize Work Items and related features	
Configure Work Item types	
Configure Time Points	
Configure Prioritization	
Configure Categories	
Configure Custom fields	
Configure Enumerations	
Configure Bulk Edit	
Define Template Work Items	
Customize export templates	
Customize field filters	
Configure calculated fields	
Configure Work Item Workflow	
Configure Work Records	
Configure the Work Items Table	
Configure queries	
Configure the global working calendar	
Configure Planning	
Configure Work Item linking	
Configure Auto-assignment	
	4-100
	4-100
	4-101
	4-101
3. 3	4-101
•	4-103
Configure Test Run workflow	4-110
Configure Test Run Signatures	4-114

5

	Configure Test Export Templates	4-117
	Configure manual Test Steps	4-118
	Configure cleanup of old Test Runs	4-123
Cor	nfigure Plans	
	Enable and administer Plans	
	Define Plan custom fields	
	Define Plan Statuses	
	Administer Plan templates	
	Configure Plan permissions	
	Use Plan enumeration for Work Item fields	
Coi	nfigure the Portal	
<b>.</b>	Configure Shortcuts and Favorites	
	Configure languages	
	Configure interface views	
$\sim$	nfigure Reports	
CUI	Source code Reports	
	·	
	Work Item metrics	
	Work Item Quality report	
<b>^</b>	Process Quality Report	
COI	nfiguring Documents and Pages	
	Configure Documents and Pages overview	
	Configure Document types and workflow	
	Security: Best Practices for Pages	
	Set up Document Types and Workflow	
	Configure Document Types	
	Configure Document Workflow	
	Configure Document custom fields	
	Configure Signatures	
	Configure Document merge actions	
	Configure PDF Export	
	Configure attachment size limit	
	Configure formulas	4-173
Cor	nfigure notifications	4-174
	Notifications, events and targets	4-174
	Enable E-mail notifications	4-175
	Access the configuration	4-175
	Configure settings	4-175
	Configure event targets	
	Customize the layout of notification emails	
	Find users with disabled notifications	
Cor	nfigure SSL support	
	Install and configure SSL	
	Configure SSL on Linux	
	Configure SSL on Windows	
	Additional resources	
	Import a certificate to the Java Keystore	
	Configure secure access to the repository	
Cor	nfigure Building	
JUI	Basic build management	4-182

Quick start: Simple Maven 2 project	4-183
Quick start: Maven 2 multiproject	4-183
Quick start: Ant build	4-183
Quick start: Build with Makefile	4-185
Quick start: External exe, batch or shell script	4-185
Quick start: Continuous integration	4-185
Configure Work Item Reports	4-186
Advanced Build management	
Complex and custom building	4-187
Build Artifacts	4-187
Supported Build Artifacts	4-188
Build Descriptors (definitions)	4-190
Artifacts vs. Descriptors	4-191
Build properties	
Working directory	4-194
Auto-recognition of Build Artifacts	
	4-195
· · · · · · · · · · · · · · · · · · ·	4-197
Maven-specific information	4-197
Scheduled Builds	
Integration Builds	4-200
Configure Repositories	4-200
Configure Repositories overview	
Configure external repositories (GIT, SVN)	4-201
Configure a GitLab repository	4-202
Configure a Bitbucket repository	4-204
Configure Resource Traceability	4-205
Configure Connectors	
Configure Connectors overview	4-222
Import an IBM DOORS project	4-224
DOORS import limitations and recommendations	4-227
Linked Data applications	4-228
Configure synchronization	4-235
System specific instructions	4-246
MATLAB Simulink	4-261
Configuration properties	4-267
System maintenance	4-268
Low resource notifications	4-268
Prevent server clogged with long-running transactions	4-269
Activities index cleanup	4-271
Configure the Scheduler	4-272
Backup Polarion data	4-277
	4-279
Use the Polarion Diagnostic Tool	4-282
Recover disk space	4-282
	4-285
	4-286
-,	4-286
Optimize the PostgreSQL database	4-286

Enumerations in the Polarion database	
Speed up the movement of Work Items to and from Documents	
Database compatibility (H2, PostgreSQL)	
PostgreSQL database Start-Stop-Restart	
Optimize Subversion	
Manage Polarion licenses	
Configure single sign-on (SSO)	
Remove unused Excel import configurations	
Configure the Dashboard	
Customize XML export templates	
Customize Excel export templates	
Configure OLE Object support	
Customize the Login screen	
Repository browser mime types	
Take a thread dump	
Advanced system tuning	
Advanced system tuning	<del>-</del> -505
Reference	5-1
Administration reference	5-1
Polarion folder structure	5-1
Passwords	5-3
Polarion system processes	5-3
System properties file location	
Scheduler cron information and examples	
Permissions management	
Dynamic roles and permissions	5-7
Default Enumeration configuration files	. 5-16
Notification events and targets	. 5-17
Project template properties	. 5-24
PDF Export Header/Footer Properties	. 5-25
Custom field types	. 5-27
Build configuration file format	
Work report Job configuration	. 5-30
Workflow Conditions and Functions for Work Items	. 5-31
Workflow Conditions and Functions for Documents	. 5-37
Workflow conditions and functions for Test Runs	. 5-40
Polarion log files	. 5-41
Xml export schema	. 5-42
Internally defined metrics	
Calculations and facts	
Velocity macros location	
Velocity variables for Active Wiki Pages	
Apache Lucene options	
License usage reference	
Self-signed SSL certificates	
User Reference	
User Reference overview	
Activity Stream	. 5-59

Attachment File Types       5-5         Captions       5-6         Dialog boxes       5-6	2
Import from Excel	2
Keyboard Shortcuts	
Time values	
Diagram editor	6
Documents	
Non-Latin alphabets	
Plans	
Templates	0
Test Management Reference	5
Work Items	
Mathematical formula syntax	8
Polarion SDK	.1
Glossary	1
About Bolorion	4

# **Chapter 1: Getting Started**

Users Ad	dministrators	Requirements Engineers
Create an account Explore the Portal Search the Portal Your Home page and Favorites Access Projects Create and manage LiveDocs Create and manage Work Items Create and manage Pages Release Plans Insert a diagram or formula Insert images and attachments	Administration basics Users and permissions Configure Work Items Manage Builds Configure notifications Configure repositories Configure Connectors Configure the Portal System maintenance Advanced administration	Different approaches to requirements Requirement phases Requirement Projects Locate requirements Create, access and modify requirements Link requirements Export or exchange requirements Review and Approve requirements Use the Traceability Matrix Traceability reporting
Developers Pr	oject Managers	Testing and QA
Access Projects Find Work Items Process Work Items Work with Builds Work with Tests View source code and JavaDocs Create Variants	Project templates Work Item types and link roles Workflows Electronic signatures Custom Work Item fields Automatic assignments Work Item Categories Users and permissions Notifications GANTT Project Plan Agile Project template Manage Project Planning Dashboards & Reports	Polarion for QA The QA template Import from Microsoft Office Test Cases in Documents Test Cases in the tracker Link Test Cases Specify Manual Test Steps Create and manage Test Runs Paramaterized Testing Define Iterations Use Round-trip for Excel Use external testing tools Reports and results

Polarion documentation is covers all features of the Polarion platform. However, depending on your license, some of the features described may not be available to you when you log on. Documentation generally notes if some feature or information is restricted to one or more product licenses. The following icon convention is used:

Icon	Product Name
ALM	Polarion ALM
PRO	Polarion PRO
REQ	Polarion Requirements
QA	Polarion QA
REV	Polarion Reviewer (see below)

In addition to the above, the following add-on is available:

Polarion Variants: This add-on provides a back-end server component which, when combined with capabilities built into some project templates, adds robust variant management capabilities to the above products.

If no product icon appears for a topic, the content is applicable to all products. If a topic applies only to certain products, the icon(s) appear beneath the topic main heading.

### Installation documentation

PDF versions of documentation covering server and license installation are included with all Polarion distributions.

They, along with additional Enterprise Setup, Single Sign-on and Integration documentation are available at SIEMENS' Doc Center.

### **Extensions**

If default functionality doesn't fully meet your needs, you may find a solution on the Polarion ALM **Extensions** web portal. There you will find a catalog of useful extensions for Polarion ALM solutions ranging from utilities to project templates, to complete features.

The Extensions portal is located at: http://extensions.polarion.com. You can click the Extensions link in the tool view of Navigation to visit the site.

# **Chapter 2: Learn Polarion basics**

# Create a user account and log on

To work with projects in Polarion, you need a user account, user name, and password. Polarion user IDs are case sensitive.

How you obtain an account depends on your company's business method.

If your business uses Polarion's default Subversion-based user authentication, there are two options:

- Your system administrator configures the system to allow you to create a new user account automatically or explicitly.
- Your system administrator creates your account and assigns a user name and password.

If your business uses LDAP Active Directory or other global user authentication, your administrator provides you with information for authentication and credentials. Active Directory IDs are usually case insensitive.

### **Note**

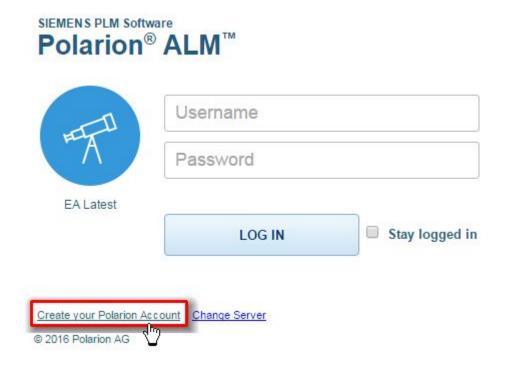
If your account is originally created using the default Subversion authentication, and your company subsequently transitions to LDAP, you may be unable to log on if you enter all lowercase characters, but the actual value stored in Active Directory contains uppercase characters. Consult your administrator if you have any problems with logon and authentication.

Unless otherwise stated, the topics below pertain to Polarion's default Subversion-based user authentication.

### Create an account explicitly

Administrators may optionally configure a Polarion installation to enable new users to explicitly create a new user account. In that case, a link appears in the logon screen that enables you to create an account.

1. Click Create your Polarion Account.



2. In the form, specify your account credentials such as user name, password, and email address for receiving notifications.

### Caution

The characters allowed in passwords depend on how your system is configured to authenticate users. The safest approach is to use only ASCII characters in passwords excluding the double-quote character ("), which is not allowed by Polarion. If you need to use non-ASCII characters in your password, consult your administrator.

Submit the form.

A logon screen appears and you can log on with your credentials.

### Tip

When you create an account this way, you have basic read access to the system, but you may not have access to all the information or all the actions you need. For example, you may not be able to open some projects, create new projects, or edit Work Items or Documents. The information you can access and the actions you can invoke depend on how your administrators configured default permissions for user-created new accounts. You may need to coordinate with an administrator to review and modify your permissions.

### Automated new accounts

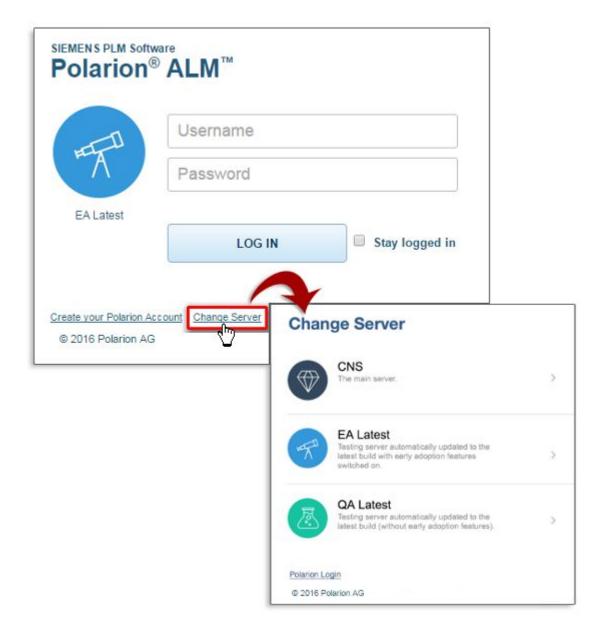
Polarion administrators can optionally configure the system to auto-create user accounts whenever a user logs on who does not already have an account on the Polarion system. This is most likely to be

the case in organizations using some type of central user-authentication system such as LDAP Active Directory. In such cases and with proper system configuration, you can log on to the Polarion portal using your company user name and password, and the system automatically creates a Polarion user account for you. The permissions you have initially depend on the system configuration. If you are unable to access some project or perform some actions such as creating or editing Work Items or Documents, you need to coordinate with an administrator to review and modify your permissions.

### Log on

Polarion has a web-based user interface. There is no local client application — you only need a supported web browser. Your administrator or other designated person should provide you with information about browsers and the access URL of your company's Polarion system.

If you are not already logged on, then when you access any URL in the system, the logon dialog box appears. In most cases, you only need to enter your user name and password. If your system is configured with multiple Polarion servers, the logon dialog box indicates which server you are currently accessing, and then displays a link **Change Server** that enables you to select the server you want to access and log onto.



The first time you log on, you see the home page of the server and the word **Repository** appears in Navigation indicating the scope in which you are now working. On subsequent logons, Polarion opens the project, the Navigation topic, and the page you were viewing when you logged off.

# Change password, email, and personal info

If necessary, you can change your password, email address, and other details of your Polarion user account.

### Open your user profile page

1. In Navigation, click (cog icon) to switch to the tool view.

2. Click **My Account**. The user account page opens.

### Tip

If your company uses a centralized user authentication system such as LDAP Active Directory and has configured Polarion to work with it, you may not be able to modify some or all of the personal information fields on your profile page. The following procedures assume that user authentication is configured to allow editing of the data.

### Change your password

- 1. On the page toolbar, click **Edit** to reveal the hidden password fields.
- 2. In the **New Password** field, type the new password you want to use.
- 3. Enter the new password again in the **Re-enter Password** field.
- 4. Click **Save** to save the changes.

The new password takes effect the next time you log on.

### Caution

The characters allowed in passwords depend on how your system is configured to authenticate users. The safest approach is to use only ASCII characters in passwords excluding the double-quote character ("), which is not allowed by Polarion. If you need to use non-ASCII characters in your password, consult your administrator before changing it.

### Change your email address

- 1. Hover your pointer over the **Email** field and click it. You can now edit the field.
- 2. Replace the existing email address with the new email address you want to use.

### Tip

Email notifications from the Polarion system are sent to this address. If you do not want to receive notifications you can select the **Disable Notifications** check box.

3. Click **Save** to save the changes.

### Change your name or initials

- 1. Hover your pointer over the **Name** field and click it. You can now edit the field.
- 2. Replace the existing name with the new name you want to use.

### Tip

The name you enter here appears in Navigation when you are logged on and in the list of all users. It appears to other users in various places, for example, the Assignee field of Work Items assigned to you.

- 3. If you want to change your initials, click the **Initials** field and change the value.
- 4. Click **Save** to save the changes.

### Change your avatar image

You can change the avatar image that displays next to your name in Navigation when you are logged on, in Activity Stream activities, and other system objects your work generates. The procedure is explained in the topic Your personal Home page.

### Tip

Be sure the page is not in edit mode. If it is, the link enabling you to change the image is hidden.

### Disable email notifications

By default, you receive email notifications according to the notification settings configured by your project administrator. Notifications are sent out in response to events that occur in the system, such as creation of a new Work Item, or change in a Work Item's status. Who receives notification is determined by the configured Notification Targets for each event. A new Work Item event might send notification to targets such as the author, the assignee(s), and any watcher(s) of the item.

While such notifications are generally considered useful, if you do not want to receive them, you can disable email notifications in your user account page.

### Tip

You might opt to disable notifications temporarily, for example, during vacation or holiday time.

- 1. Log on to the portal, and in Navigation click (cog icon) to switch to the tool view.
- 2. Click **My Account**. The user account page opens.
- On your user account page, select the **Disable Notifications** option, and then click **Save**.

# **Understand your user permissions**

Polarion has a sophisticated, multitier security architecture with extensive control over user permissions, which regulates who has what level of access to what. Permissions can be defined for user roles, and/or for specific types of content in the system. Polarion's permissions sit atop the access control mechanisms of the underlying repository.

If you encounter any of the following or similar problems, the most likely reason is that you are not currently granted the requisite permissions:

- You cannot create a Work Item in the Tracker or a Document.
- You cannot use Work Items, Requirements, or Test Cases, for example, after initial import from Microsoft Word or Excel.
- You cannot overwrite a referenced Work Item of some type in a Document.
- You cannot move a Work Item of some type to or from a Document.
- You cannot create linked or linking Work Items of some type.
- You cannot create artifacts in a project, such as Documents, Plans, or Test Runs.
- You cannot create or open a project.

To create new projects your user account must be assigned a user role which has been granted that permission. If your system is configured to support multiple repositories, you must have the necessary permissions for the repository that hosts the new project.

User permissions are defined in the User Management topic of Administration, and configured by administrators. If you cannot create, modify, or view any content that you believe you should have access to, either contact your administrator directly or follow whatever contact chain your team or organization has established.

# **Navigation and content**

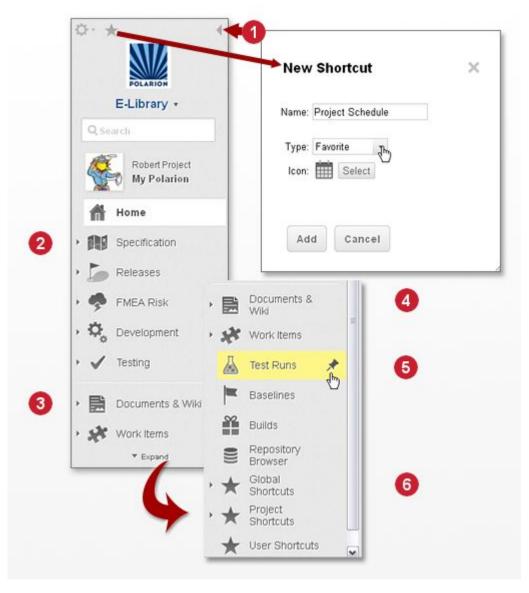
When you first log on, if the product you are using is not yet licensed for production, the log on screen offers the option of running with a built-in evaluation license, or activating the installation online or offline. After a license is in place, you enter the portal and can begin navigating around to different areas.

The Navigation panel occupies the left side of the Polarion portal page. By default it displays the Topics view which shows the topics available to you under the product license you are using. Topics provide access to various feature and content areas and to Shortcuts, which quickly retrieve sets of Work Items together with their context. Some Navigation topics expand and have subtopics that either narrow the scope of the information shown or provide access to additional functionality. There are a number of ways you can Personalize Navigation.

You can easily hide the Navigation panel using the icon in the upper right corner of the panel. When the panel is hidden, you can restore it by clicking the left border of the page as shown in the figure below.

The Navigation topics are specific to the information scope in which you are working — repository, project group, or project. The topics shown may also vary according to the interface view you are using. Polarion administrators can define and customize Interface views. If you have permissions for Administration, only administration and configuration topics are shown when you access it.

You can create **Favorite Shortcuts** in the Navigation panel that take you quickly into the key things you need to access in the project such as Documents, Pages, Work Items, and dashboards. The following figure illustrates sections of the Navigation pane and the New Shortcut dialog box.



1 Hide Panel	4 Expanded Topics
2 <sub>Favorites</sub>	6 Pinned Topic
3 Topics	6 Shortcuts

The Navigation panel has two views — Topics and Tools. The Topics view is shown by default.

The Tools view of Navigation provides access to some things that are usually needed only occasionally.

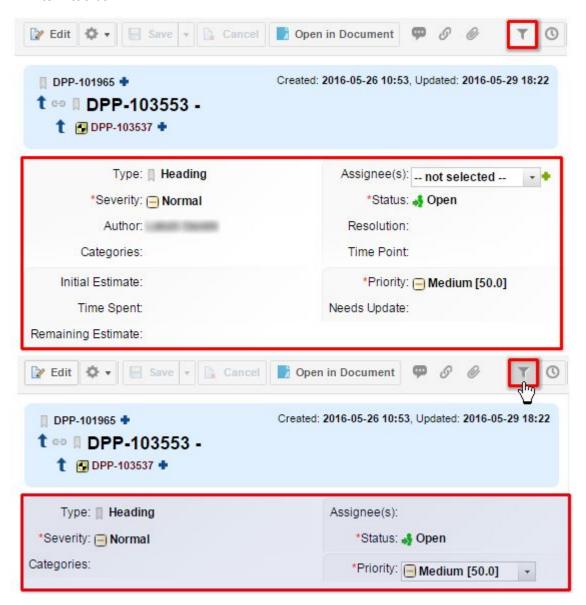
The following figure shows how to switch the views:



The Tools view provides access to:

- Logout: Log off the Polarion portal.
- My Account Access your user profile where you can change your password and edit other details.
- **Manage Shortcuts**: Manage all your current **shortcuts**, including Favorites, and optionally reset your Favorites and other user-specific shortcuts to the original default configuration.
- **Administration**: Appears only if your user account is assigned one of the administrator roles for the current project and/or the repository. Enables access to the all administration features for the scope.
- Help: Launch the online help system in a new browser instance.
- Views: Select the interface view and the data rendering mode you want to work with.
   Interface views support different work contexts, Developer or Manager, for example, getting less needed information out of the way.

You can change the level of detail displayed on a Work Item using the T button on the Work Item toolbar.



Which fields are displayed or hidden can be defined by an administrator in the Form Filters section of Form Configuration.

In this menu you can select a preferred data rendering mode — Normal or Compact. The Compact mode reduces the amount of white space around the data displayed in the user interface, particularly in the vertical dimension. The more compact rendering can be helpful on smaller screens.

The lower portion of the panel provides information about the Polarion system, such as license type and build number. Also provided are links that enable you to send feedback to the development team, contact technical support, and visit the extensions portal.

The rest of the portal page is the area where you work with the various Polarion features. Pages provide access to the content and features of the Navigation topics like Documents, Pages, Dashboards, and Builds. What displays in this pane depends on a number of factors, including

current information scope, currently selected Shortcut or current Work Item query, and whether your are working in Administration.

# **Navigation topics**

The Navigation panel presents various Topics, which are descriptive links leading to content and features. Not all topics listed are available with every product license. Where availability is limited by license, this is noted.

### **Note**

Even if the product license makes a topic available, it may be hidden in the interface view you are using. If you miss a topic you think your license should provide, try switching to another interface view by clicking the provide icon in **Navigation**, and then clicking **View**.

### Home topic and Home pages

The **Mome** topic of every project, project group, and Repository displays a special **Home** page. Polarion provides some default content in these pages, which varies depending on the scope you are currently viewing — Repository, project group, or project. Users can, and normally do, modify Home pages to provide information that is useful for team members and stakeholders working in the respective scope. You can learn how to modify page content in the topic Working With Pages.

### Dashboard

### **ALM**

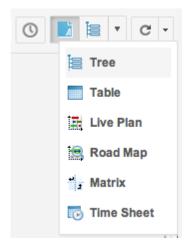
Scopes: Repository, Project Group, Project

The Dashboards topic provides status information through charts, metrics, and reports. Depending on the scope you are working in, the status is compiled for the current project, all the projects in a project group, or all the projects in the entire repository. Dashboard information is rolled up from repository data that is automatically tracked in real time as teams work on projects, create or complete Work Items, and adjust project plans.

Managers can decide how often the status information should be updated, and administrators can create scheduled jobs to update dashboards at less busy times when more system resources are available. Users with the requisite permissions can explicitly invoke a dashboard update to see the up-to-the minute state of a project or projects.

### Work Items

The Work Items topic provides access to the integrated tracker. The topic is available for Repository, project group, and project scopes. This topic provides access to all the Work Items in the selected scope. You can select different views of these Work Items using the view selector control on the toolbar of the Work Items page. Depending on the license you are using, not all these views may be shown on the menu.



# A Test Runs

# ALM REQ QA PRO REV

The A Test Runs topic provides access to Test Runs. Test Runs encapsulate a set of test cases and record an instance of execution of each test case along with its status. You can create Test Runs for both manual and automated tests. To browse Test Runs, you must have read permission for Test Runs. To create new Test Runs or perform manual type Test Runs, you must use a license for a product having test management support, and you must have the requisite permissions for the operation.

This topic also provides access to Test Run templates, and to Round-trip for Microsoft Excel. Test Run templates store information about the Test Cases to be run, the tested build, testing environment, and other information. Excel Round-trip enables exporting a Test Run's Test Cases to Excel for sharing with external testers and reimporting of testing results logged by external testers in Excel into the Test Run.

# **Documents and Pages**

The Documents and Pages topic provides access to the Documents and Pages belonging to the current repository or project. In the Repository scope, it's a good place to create content that all portal users might need to access. In the project scope, the information created in this topic should be relevant for project stakeholders.

When you expand the Documents and Pages topic in Navigation, you see a listing of the spaces, Documents, Pages, and Classic Wiki pages in the current scope, unless there are too many to display comfortably. In that case, you see a prompt to use the search field that appears at the top of the **Navigation** panel. Enter a word or phrase you want to locate, and select the content type. Filter results appear incrementally as you type.

### **Documents**

# ALM REQ QA PRO

Documents provide an easy, familiar way to author, collaborate on, and share content online. Documents can contain Work Items which can be managed throughout a process as artifacts using all available tools, while remaining in an easy to use online Document.

### **Pages**

Pages differ from Documents in that they do not contain Work Items such as requirements and test cases. However, Pages can show information *about* Work Items.

### Plans

# ALM REQ QA PRO REV

The Plans topic provides release management tools and enables access to the release plans in the current project or the entire repository. Plans enable teams to define Iterations and Releases. For both of these, the team can set the start and end dates, specify which Work Items are to be implemented, and monitor progress in near real time via built-in charts.

Plans provide visibility on status and progress for all stakeholders and can reveal potential risks and blocks, so managers can make adjustments to the work scope, delivery date, resources, and other parameters.

### Baselines

# ALM REQ QA

The Baselines topic enables you to see and compare existing project baselines, and to create new baselines. Baselines are basically snapshots of the status of a project some point in time. By creating baselines periodically and comparing them, it is possible to gauge the progress of a project over time. The Time Machine feature of Baselines enables you to load a project baseline and browse it just as if it were the current state.

### Builds

### ALM

Scope: Project

### **Note**

Availability in Polarion Requirements:

This feature is also available in Polarion Requirements, but it is hidden in the default configuration. An administrator can modify the Navigation configuration to show it.

The Builds topic enables you to access the current and recent builds of any project. You can find information about build jobs, create a new build, and access the build results.

# Quality

## **ALM**

Scopes: Repository, Project Group, Project

The Quality topic centralizes the live dashboards that report metrics related to software quality.

The topic has the following subtopics:

- **Process**: Provides a dashboard reporting on levels of compliance with a development process.
- Work Items: Provides a dashboard reporting various metrics and statistics about Work Items in the information scope you are working with.
- **Testing**: Provides a dashboard reporting on levels of code coverage by unit tests, success ratio of tests, and enables drilling down into more detailed reports.
- **CMMI**: Provides a dashboard reporting on levels of compliance with the CMMI maturity model. Report covers the information scope you are working with, so you can judge compliance project by project or for all projects in aggregate.

This topic can be useful if you use Polarion for Quality Assurance.



### Reports

# ALM PRO REQ

#### Note

Availability in Polarion Requirements:

This feature is also available in Polarion Requirements, but it is hidden in the default configuration. An administrator can modify the Navigation configuration to show it.

The Reports topic provides access to several reports related to project source code, including unit tests, test coverage, and Javadoc for Java projects, and also provides read access to source code files. You can explicitly launch report calculation from this topic as well.

## ♣ Monitor

The Monitor topic provides access to information about current and scheduled jobs and, for users with permissions, the possibility to create and launch jobs. Jobs are background processes run on the server that run various operations such as updating the Live Plan or a dashboard.

# **Repository Browser**

The Repository Browser topic provides access to an integrated Subversion client. This client enables you browse the content of the Subversion repository that is installed and used with Polarion ALM by default. You cannot use this client with any external Subversion repository configured in the Repositories topic of Administration or with any other version control system that may be custom implemented as the default Polarion repository.

If your system is configured with multiple Polarion instances, the Repository Browser accesses the default Subversion repository on the server you are currently using. In addition to browsing the repository, the Repository Browser can perform limited basic repository operations such as adding files, managing folders, and committing changed files. It is not designed to replace a full-featured Subversion client.

# Że

### **Shortcuts Topics**

The Navigation panel contains three topics that provide **Shortcuts**. Shortcuts are special saved queries which retrieve both data and some presentation of the data. They provide quick access to frequently-needed information in the Polarion portal regardless of where you are working at the moment. The three top-level shortcuts topics are:

- **Global Shortcuts**: These shortcuts are available to all portal users. The shortcuts found under this topic must be configured by a global administrator.
- Project Shortcuts: These shortcuts are available to all users working in a specific project. Each
  project can have a different set of shortcuts. Project shortcuts can be configured by a project
  administrator in Administration. Project users with appropriate permissions can create project
  shortcuts by saving the results of some query as a shortcut and specifying that the shortcut
  should be saved in the project scope.
- **User Shortcuts**: This topic is empty by default. You can create shortcuts for your personal use by saving the results of a query as a shortcut and specifying that it should be saved as a user shortcut. User shortcuts that you create are not visible to other users.

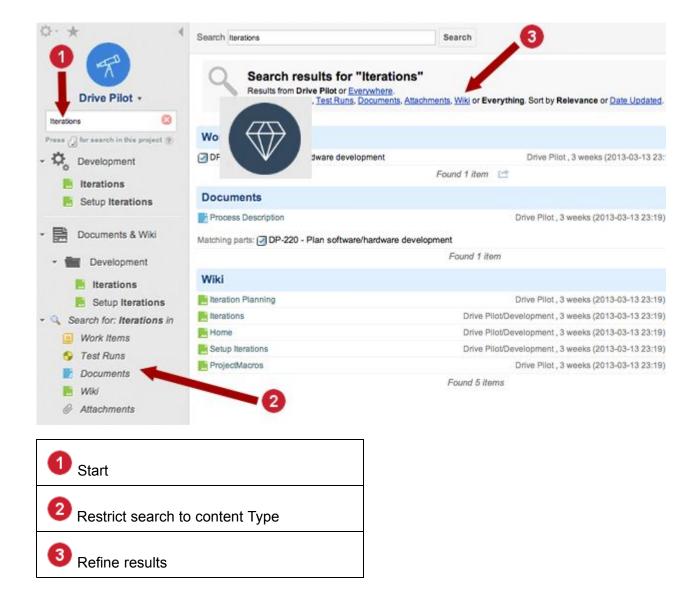
Don't confuse User shortcuts with Favorites. Favorites are accessible only in the project where you create them, and navigate only to content of that project. User shortcuts can be defined to navigate you to anywhere in the portal from anywhere in the portal.

Shortcuts can help you personalize Navigation and quickly search for Work Items you need to refer to often.

### Search the Portal

Your Polarion portal may eventually host a large amount of content and finding what you want may necessitate using the portal search facility. Your starting point for content search is the search field in Navigation. Simply type in a word or phrase you want to search for. You can search Navigation content, current project content, and portal content.

You can restrict and refine your search results.



### **Navigation Content**

When you begin typing a search term, the initial scope is Navigation. For example, if you type the word "Developer" and Navigation contains a page title Developer Guidelines and a Document title Developer Documentation, both these items appear as clickable labels in the Navigation panel. Essentially you are filtering Navigation for the search term.

You can restrict the Navigation search results to a specific content type:

- · Work Items
- Documents
- Pages
- Test Runs
- Attachments

Restricting a search to a content type is useful if your search term returns many items. Choosing one of the content types shown in Navigation when search is active limits the results to the selected type. Following the example above, if you type the word "Iterations" and select **Documents**, only the pages shown above in the **Documents** section of the search results page appear in the results.

### **Project Content**

When you are working in a project, you can enter a search term in the **Search** field and press **Enter** or **Return** to search the project content. The search results appear in a results page, which is divided into sections for different content types, such as Work Items, Documents, Pages, and Test Runs.

The Search Results page has links that filter for specific content types or change the sort order of the results. All individual results are links that take you to the respective content.

#### **Portal Content**

There are several ways you can search the entire portal:

- Open the Repository, enter a search term in Navigation, and press Enter or Return.
- When working in a project, enter a search term in Navigation, hold Shift and press Enter or Return.
- In the Search Results page of a project-scope search, click the **Everywhere** label to search the entire portal for the term used in the project-scope search.

### Tip

If you switch from a project to the repository or vice versa, the scope of the search for the current string changes according to where you are, but your search string remains in effect until you change it.

If you search for Attachments, the search results show the parent object such as a Work Item or Document that contains each found attachment. You can then download found attachments from the search results.

# Create a project

# Project definition and options for creating

A project in Polarion is a collection of artifacts and data residing on the system in a repository folder, which is dedicated to managing some specific development activity throughout its life cycle. There are two basic scenarios for creating a project:

- Create a new project from a Project Template: Available to all users who have permission to create projects.
- Mark an existing repository folder as a Project: Available to project administrators only.

### Create a new project from a template

### **Background information**

Polarion products come with different sets of **Project Templates**. These range from a completely empty project with just basic Work Item types and placeholder content, to projects preconfigured especially for some type of development activity, for example, requirements management, agile software development, test management, or system engineering. These templates contain predefined key artifacts such as specification documents and reports. Work Item types, workflow, and other predefined configurations. The templates can provide a time-saving head start for projects of the type for which they were designed.

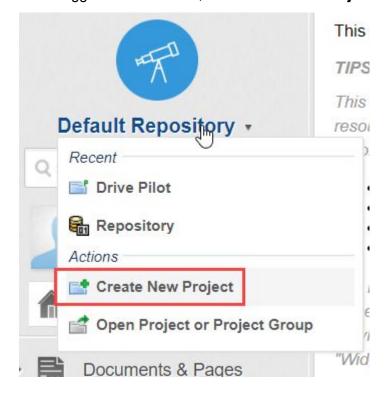
The list of available Project Templates is documented in Create New Project dialog box and Templates Screen. The default templates provided with Polarion products may be suitable for most projects as delivered. However, it is possible to customize any Project Template.

#### **Procedure**

### Tip

The new project process creates a new folder in the repository. The New Project wizard asks you to specify the location of the new folder. You can familiarize yourself with the repository structure using the Repository Browser. The wizard provides a graphical interface to browse and select a parent folder.

1. While logged on to Polarion, click **Create New Project** in the Navigation menu.

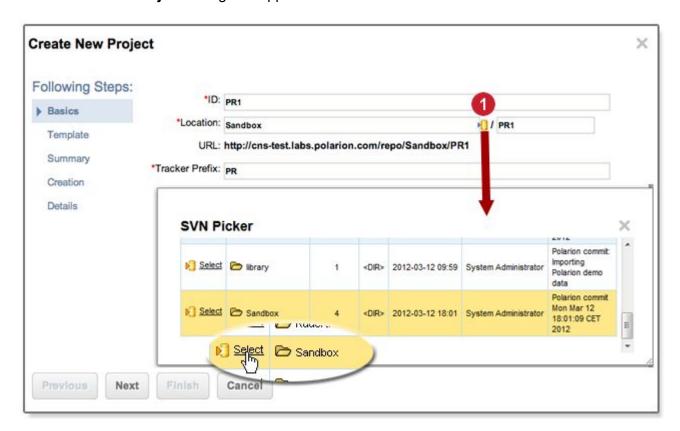


### Tip

The action is also available in the **Open Project or Project Group** dialog box.

If the action does not appear, it means you do not have permission to create new projects.

The Create New Project dialog box appears.



- Click icon to visually select repository location for new Project.
- 2. On the **Basics** page of the **Create New Project** dialog box, fill in the fields as follows:
  - **ID**: Enter a unique identifier for the project. Acceptable characters are a-z, A-Z, 0-9, underscore, dash, and period.
    - As you type the ID string, your entry is suggested as the name for the project's folder in the repository in the **Location** field.
  - **Location**: Specify the path and name of the root folder of the new project, referred to as the *project folder*, to be created in the repository. Note that any value you type for the path must be valid for use in a Subversion repository path. The **URL** label shows the full project path for the new project as a repository URL.

By default, the project folder is created under the repo folder. You can see this when you look at the URL. If you want to create the project under a different folder, you can either type in a path to the parent folder, or click the icon to launch the SVN Picker dialog box in which you can browse the repository folders and select one as the parent location for the new project.

### Note

The Location and ID fields may be prefilled to a default value if system properties are so configured.

Tracker Prefix: Enter a value that is prefixed to the system-generated IDs of all new Work Items in the project. Generally you should limit the value to two to five characters that relate to the project name.

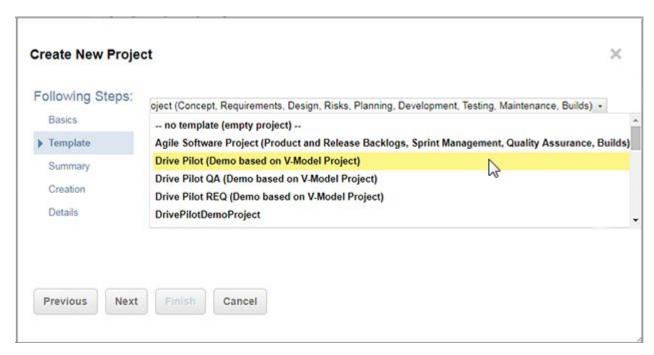
### Caution

Specify a unique value that is not used in any other project. Otherwise, revisions may get linked to the wrong Work Item when developers using an external Subversion client cite Work Item IDs in commit messages to auto-link commits to their tasks in Polarion.

For example, if your project is named **DrivePilot**, you might enter DP as the Work Item prefix. The project's Work Items then have IDs like DP-279 assigned to them by Polarion. If project DrivePilot exists in your system, and you create a new project named DrivePal, you should not specify DP as the Tracker Prefix for DrivePal. If you do, then there could be two Work Items in the system with DP-3 as the ID. If a developer cites **DP-3** in a commit message in an external client, the revision is linked to the first Work Item found by the system having that ID, which may or may not be the right one.

In this case you might specify a value like DPAL, giving DrivePal items IDs like DPAL-42.

3. Click **Next** to move ahead to the **Template** page of the dialog box.



The figure above shows an example of the list of available templates. The actual templates listed vary depending on the product license you are using, and whether or not any custom templates have been created and added in the system configuration.

- 4. Select a template from the list of existing Project Templates. For information about the standard templates that come with Polarion, see Templates page.
- 5. Click **Next**. The **Summary** page appears showing the project parameters you selected. If you want to make any changes, click **Previous**.

When you are satisfied with the summary, click **Next** to create the project in the Repository. The **Creation** page appears showing a processing indicator. The time required to create the project depends on the content of the template and the current load on the server. Processing may take several minutes.

When the process is finished and the **Creation** page displays Project created, click **Next**. The **Details** page appears.

Edit the available project details as desired. For information on the fields, see Details page.

### Tip

After the project is created, its details can only be modified by an administrator.

7. Click **Finish** to complete the procedure.

### **Note**

If a project is deleted, the settings for its location are preserved in the access control file of the internal Subversion repository. If you try to create another project in that same location, Polarion warns you and offers the possibility to overwrite the old settings. You should follow your administrator's policy on such overwrites.

# **Access projects**

### Open an existing project

Users have read access to projects by default. However, administrators can restrict access to any project and to some types of content within projects, such as Documents or Plans. Your ability to open a project and read or change project content and artifacts depends on the user roles assigned to your user account, and the permissions granted those roles by your Polarion administrators.

Follow these steps to open a project:

- 1. Log on to your Polarion system.
- 2. Click the name of the current project or repository shown in the Navigation panel to display a menu containing options for opening a project.
- 3. The **Recent** section of the menus shows the projects you have opened recently. If the project you want to open appears in the section, click it to open the project. You can then skip the remaining steps below.
- 4. If the desired project does not appear in the Recent section, click Open Project or Project **Group** in the **Actions** section of the menu.
- 5. In the Open Project or Project Group dialog box, locate the project you want to access and click the name to open it. If the project is part of a project group (green folder icon), you need to expand the project group node to see the project.

### Tip

Here are several tips about opening projects:

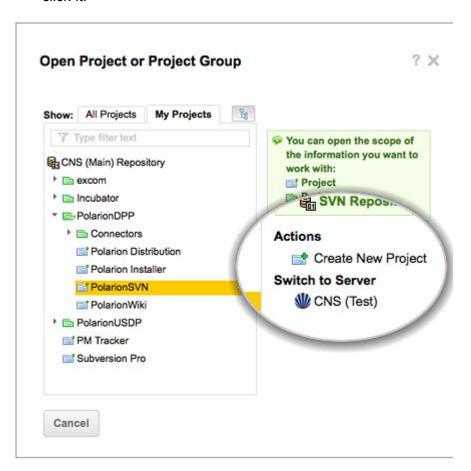
- If there are many projects in the system, you can select the My Projects tab to limit the list to only those project for which you are assigned some role.
- You can navigate directly to any URL in your Polarion system by entering it in your browser's address bar. You may find it useful to create bookmarks or Favorites in your browser for frequently visited URLs such as projects.
- You can create Shortcuts in your Polarion system that quickly navigate you to any project or project content.

 If you want to see dashboard and report data for a group of projects, select the desired Project Group (green folder) in the steps above rather than an individual Project (blue folder).

### Access projects on another Polarion server

Some Polarion systems are configured with multiple Polarion servers. On such systems, you might be logged on to one server and then need to access a project hosted on a different server, requiring you to switch servers.

- 1. Click the name of the current project shown in the Navigation panel, or on **Repository** if you are currently in that scope. A menu appears.
- 2. In the in the **Actions** section of the menu, click **Open Project or Project Group**.
- 3. In the dialog box, locate the server you want to switch to in the **Switch to Server** section and click it.



#### **Note**

If the Switch to Server section does not appear in the dialog box, the most likely reasons

- Your system has only one Polarion server.
- Other servers are not currently accessible to your network location, or are offline.
- You do not have access permission for other servers.

# Access a project's documents

Documents belonging to a project are found under the **Documents and Pages** topic in **Navigation**. The topic is organized into *spaces*, which are analogous to folders. The documents belonging to the project are located in the various spaces. Every project has a space named **Default**, which may not be deleted. Other spaces may be created by the Project Template used when creating the project, or by users. Every space has an Index page that lists all the LiveDoc Documents and Pages in the space. A space's Index page cannot be explicitly deleted — it is only deleted if the space it belongs to is deleted.

#### **Note**

Legacy live documents: This term was used in several earlier Polarion versions through version 2010. It applied to Microsoft Word and Microsoft Excel documents with special customization enabling Polarion to identify content elements as Work Items. The technology was deprecated and replaced with the current LiveDoc technology in version 2011, and Polarion customers were advised to migrate to the new LiveDoc technology.

If you created such legacy live documents and have not deleted them, they still exist in the repository, but they are no longer indexed and do not appear in query results. If you still need the legacy live documents in that format and you want them indexed, please consult Polarion technical support.

- 1. Log on if you have not already done so, and open a project.
- 2. In the Navigation pane, click Documents and Pages.
- 3. Expand the node for any space to see and browse its Documents and Pages in Navigation. The items appear in alphabetical order regardless of type. If there are too many items to browse easily in Navigation, click Index to open the space Index page, which displays a table of the project's Documents and Pages.

#### Tip

If there are many Documents and Pages, you may find it quicker to use the Search feature to locate the content you want.

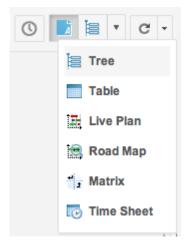
## **Access Project Work Items**

Work Item is a general term applied to granular artifacts that need to be worked on or processed by people through a workflow and lifecycle. Work Items most often represent things like requirements, tasks, change requests, test cases, and defects/issues. However, it is possible to define Work Item types to represent anything that you need to track and manage through a workflow-controlled process.

For quick access to a project's Work Items, click the **Work Items** topic in **Navigation** pane when the project open.

#### Work Item views

The Work Items page enables you to query for Work Items, edit them, and perform other operations on them. You can select any of several views that display sets of Work Items in different ways:



The names of the views describe the format or functionality of the information presented. The views available in the menu when you select Work Items in Navigation depends on the product license you are currently using. Not all licenses show all Work Items views.

- Table: Displays a table of Work Items retrieved by the current query.
- Tree: Displays a table of Work Items retrieved by the current query, plus any Work Items linked to those items, in a tree structure.
- Live Plan: Presents a GANTT style chart containing only the Work Items retrieved by the current query. You can select a Work Item in the chart and edit it in the detail area in the lower part of the page.
- Road Map: Presents a table of Work Items selected by the current query. You can further filter the selection according to Time Point.
- Matrix: Enables you to trace links between Work Items. You create two queries, one for Rows
  and one for Columns of the matrix. With the query results you can specify the relationship you
  want to see in the results, for example, dependency, duplication, or relativity.
- Multi Edit: Places all Work Items retrieved by the current query into edit mode, enabling editing of the main data fields.

Time Sheet: Enables initial time reporting or redistribution of reported time for the set of Work Items retrieved by the current query, usually for a specified period of time.

## Using queries and shortcuts

You can use queries and shortcuts to filter the list of Work Items retrieved in different pages displaying Work Items.

- In Navigation, you can filter the \* Work Items table for a particular type of Work Item by clicking on the one of the subtopics under the **Work Items** topic.
- You can use certain Shortcuts in Navigation to filter the list of Work Items in the Table and other views. For example, the My Work Items shortcut retrieves only items assigned to you.
- You can further refine any set of retrieved items using a query. The parameters of the current query are always shown visually in the Query Builder located in the toolbar of the Work Items page. You can modify the current query to further filter the Work Items shown.

## Tip

You can save the results of any query as a **Shortcut**. Click in the **Navigation** pane.



The saved Shortcut stores the current Work Items view, all query parts, table columns, and table sorting.

# **Chapter 3: User Guide**

## Home

## **About Home pages**

Every project in the Polarion portal has a Home page. The default content is provided by the project template. Home page content can be modified by users with the necessary permissions. The repository also has a Home page, also modifiable by users who have permissions allowing them to change content in that scope.

The **Home** topic of Navigation provides access to the Home page of either the current project, if you have a project open, or the repository if you are working in that scope after opening the repository.

Every Polarion user has a personal home page and can customize the page to show the most useful information. The page is not visible to other users. Users can also personalize Navigation to provide quick access to the project content they need most often.

## Tip

You can always access the repository Home page via the Polarion icon in Navigation.

Home pages are persistent LiveReport type Pages, and cannot be deleted like user-created Pages. If you have permission to modify home pages in the scope (project or repository), you can edit a Home page.

For more information, see Work with Pages.

# Work with Home pages

## **Customize Home page content**

The repository Home page, and every project home page, initially contain some default content. The repository Home page, and every project home page, initially contain some default content.

- Default content for the repository Home page is predefined.
- Default content of a Project home page is provided by the Project template.

The default content provides information about what the Page is and how to begin editing the content. Both types of home pages are customizable.

In the global, Repository scope, the repository Home page can be customized to provide essential information for all users of the portal. This is typically company-wide information. For example, you might add information that new employees or contractors need. The repository has its own space, and as with projects, you can create more spaces and Pages in this scope and link to them in the repository Home page.

On a project Home page, you and your team can provide information that is important to the people accessing the project. You don't need to include everything on the Project Home page; you can create other Pages, and then provide links to them in the project Home page.

### **Edit Home page content**

To be able to edit the content of any Home page, you must have write permissions for the scope in which you are working (project or Repository). If you find you cannot edit a Home page, contact the project or portal administrator.

- 1. Click **Expand Tools** at the top of the page to expose the page toolbar
- 2. On the page toolbar, click **Edit**. The page opens in the Page Designer.
- 3. Edit the page as you need.

#### **Note**

From version 2016-SR1, default pages such as Repository and space Home pages are created as LiveReport type Pages. If these pages were created using a previous version and still have only the default content, the format is switched when the system is updated. If these pages have been customized, they remain as Classic Wiki pages and can only be switched manually to the LiveReport format. For information on the conversion procedure, see Converting from Classic Wiki Format.

# Your personal Home page

## **Background information**

# ALM REQ QA PRO REV

When your user account is created, your personal Home page is also created. You can access it via the **My Polarion** link in the **Navigation** panel.

#### Tip

Don't confuse your personal Home page with your User Account page.

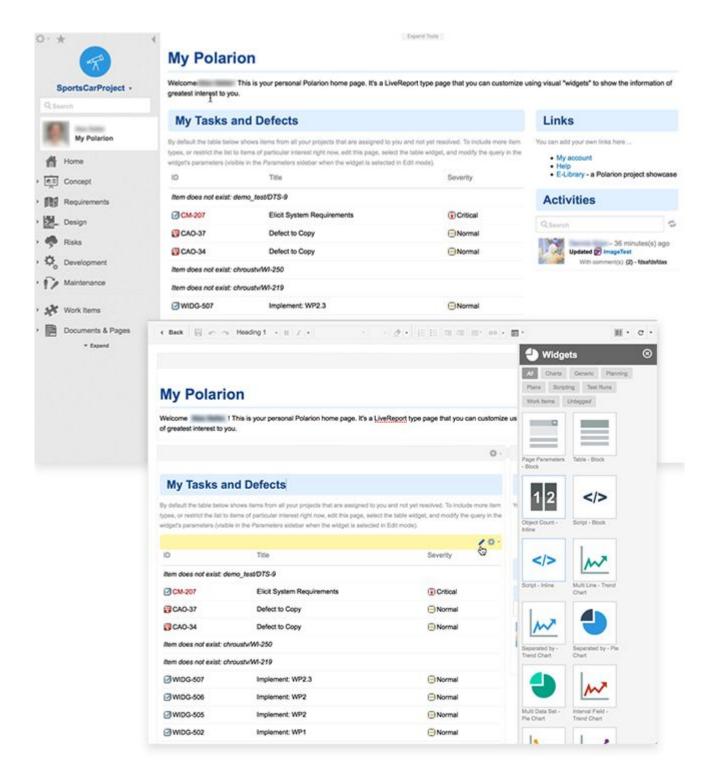
Your personal Home page, named **My Polarion**, is a persistent LiveReport type Page that cannot be deleted like user-created Pages. Your My Polarion page is not public and only you or an administrator can view and edit it.

#### Modify your personal Home page

The default content of the personal Home page is merely a suggestion. You can change both the content and the page layout to show whatever information you find useful for your work now, and you can update the page any time. When you access your personal Home page for the first time, a basic template page is displayed showing:

An introductory paragraph.

- A region displaying your currently-planned Tasks and Defects, retrieved by query in a Table Block Widget embedded in the Page.
- A region displaying Documents awaiting your signature (if any), retrieved by a custom script in a Script Block Widget, and providing links to the Signatures view of the Documents.
- A region displaying Work Items awaiting your approval (if any), retrieved by query in another Table Block Widget embedded in the Page.
- A list of Work Items assigned to you, with outgoing links marked as suspect and implemented in a Table Block Widget.
- A column with:
  - o Links to some key pages which can be useful for you, including a link to your user account page where you can change password or email address.
  - o The Activity Stream of the current project in which you are currently working.



Your personal Home page is a LiveReport type Page built with visually editable Widgets that you use to build your home page. You have the full features of the LiveReport type Pages available to you. For information, see Working with Pages.

### Convert Home page from the Classic Wiki format

From version 2016-SR1, new users' personal home pages (and other default pages such as Repository and space Home pages) are created as LiveReport type Pages. This technology replaces the older Classic Wiki technology for information and report pages. LiveReport type Pages are easier for nontechnical users; thanks to construction using visually configurable Widgets rather than mark-up language and code.

If you have an existing My Polarion (or other default) page created with a Polarion version prior to 2016-SR1, you can easily convert your existing page to use the newer and easier Pages technology.

### **Identify the Page's format**

- 1. Log on to your Polarion portal, and in Navigation click **My Polarion** (or select any space's **Home** node).
- 2. At the top of the page, click **Expand Tools**, and then click **Edit**.

If you see the **Widgets** sidebar on the right, your page is already using the newer Pages technology and there is no need to convert. If you see the **Wiki Syntax Help** sidebar, your page is using the deprecated Classic Wiki technology. Classic Wiki will be supported for some time to come, but conversion to the new Pages technology is recommended. Please note that Classic Wiki content will not be migrated, and you will need to customize the default content of the LiveReport page to suit your needs. The source of the original Classic Wiki report will still be available to you for copy/paste.

## **Convert your Page**

- 1. Access your My Polarion page and click **Expand Tools**.
- 2. Click 🔯, and in the menu choose **Switch To LiveReport Page**.
- 3. In the next dialog box, click **Switch** to convert the page.
- 4. Customize the new LiveReport page, using the source of the former Classic Wiki report as a guide for queries of various Widgets, and so on.

#### Note

After you save any customization to the page, you can reset it to the default content any time by clicking Reset.

## Specify an avatar image

In your user account page, you can specify an avatar image to display in Polarion. The image is displayed with your activities in the Activity Stream, and you see it in the Navigation panel when you are logged on. There are two options for avatar images:

- Upload an image via your user account page.
- Use an image already uploaded to your account on www.gravatar.com.

A default avatar image is displayed in Polarion until you specify a different image. If you remove the image you specified, Polarion reverts to this default image.

Your avatar image should be square, and ideally 50 x 50 pixels. Larger square images are resized to 50 x 50 pixels (maximum image file size is 50 kilobytes). Rectangular images are automatically cropped or expanded to fit into a 50 x 50 space. The result may not be what you want, so cropping the image yourself is recommended. Supported image formats are: .gif, .png, and .jpg/.jpeg.

### Upload an avatar image to your user account

An uploaded image replaces the Polarion default avatar image, and takes precedence over any gravatar.com image that may be set on that site for the email address specified in your Polarion user account.

- 1. After log on click the **Tool** in the upper-left corner of **Navigation**, and then click **My Account**.
- 2. Click Upload avatar.
- 3. In the **Upload Avatar** dialog box, click **Browse**, select the desired image file on your local system, and click the **Upload** button.

#### Use a gravatar.com avatar image

Avatar images on gravatar.com are linked to email addresses. To use an avatar image from gravatar.com in Polarion, make sure the email address used in your Polarion user account is set up in your gravatar.com account and some image is linked to that address. When you first set up a new email address on gravatar.com, it may take as much as 30 minutes before it appears in Polarion or other places that pull gravatar images.

#### **Note**

Your Polarion administrator may optionally disable the pulling of gravatar.com avatar images for your Polarion installation. If you are sure your Polarion email address is set up in your gravatar.com account and you don't see the avatar in Polarion, this is probably the reason. In this case, you need to upload an avatar image to your Polarion user account page (see the previous section).

#### Access your user account page

You can access your user account page from the My Account link in the Tool view of Navigation. You can also access this page from the **My Account** link in your personal home page.

On this page you can:

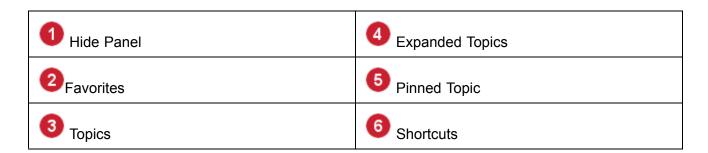
- Change your password. Click **Edit** to reveal the password fields.
- Edit your full name and initials, and provide descriptive information about yourself. This information appears to other users.
- View or change your email address.
- Disable notifications.

- Access your User Calendar. (See Configure your user calendar.)
- Access the Work Items you are currently watching (via the Watches button).
- Reset the user interface. The Reset UI Defaults button removes all your user interface
  customizations and restores confirmation dialogs if you have previously chosen not to show
  them. The reset affects only you.
- View your current **Global** and **Project** user roles, and modify them if you have the necessary permissions.
- View and edit your time-splitting assignments, if you split your working hours among different projects. This makes the calculation of the project plans more accurate.
- View and edit your Votes.

# **Personalize Navigation**

## Personalize navigation overview

In addition to customizing your personal My Polarion home page, you can personalize Navigation to make it quick and easy to access the project content you most need to work with right now.



User Shortcuts

The **Navigation** panel contains two sections of navigation links. The first section is for your **Favorites**. Favorites are personal links to project content that are only accessible when you are working in the project. (This is different from User Shortcuts, which are accessible from anywhere you may be working in the portal.) Favorites enable you to create quick, easily-accessed Navigation links to any project content you need to work with most frequently. If your needs change, you can remove Favorites and add new ones to Navigation. For example, if you are working on a requirements

specification Document, you can create a Favorite that links you to it. If there are some Wiki pages you need to refer to while working on the specification, you can create Favorites for them as well. Then later, if you switch to working on a test case specification, you can add a Favorite for that, optionally removing other Favorites if you no longer need them frequently.

Initially, the Favorites section may be populated with default shortcuts preconfigured in the project template on which your project is based. If you don't need these default shortcuts, you can easily remove them. See Modify Favorites.

The second section contains the portal topics that are currently accessible to you. By default this section is collapsed and shows only those topics currently "pinned". Pinned topics show when the topics list is collapsed. The **Documents and Pages** topic and the **Work Items** topic are pinned by default.

You can further customize your Navigation panel by pinning some topics and unpinning others. Hover your pointer above a topic and click the Pin icon. Keep in mind that the topics that appear in the expanded list depend on several factors including the product license you are using and the interface view configuration, which controls what topics appear in the list when an interface view is active.

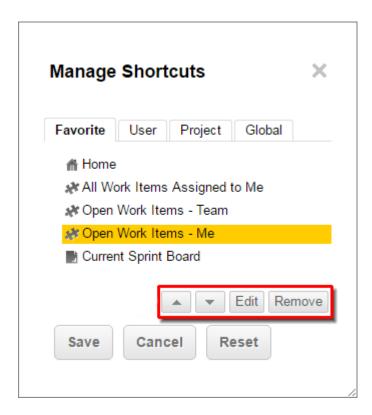
#### Create a new Favorite

- 1. In the project where you want to create the new Favorite, navigate to the content you want to access with it. If it is a specific Work Item, you must open the item itself. If you just select it in a Work Item view (Table, for example) the shortcut will lead to the view, not to the item.
- 2. In **Navigation**, click the star **x**icon. The New Shortcut dialog box opens.
- 3. (Optional) Change the default Favorite name in the **Name** field.
- 4. Leave the default item **Favorite** selected in the **Type** field.
- 5. (Optional) Select a different icon from the icon library, and then click **Add** to create the new Favorite.

#### **Modify Favorites**

You can modify the content of the Favorites section. You can:

- **Remove** existing Favorites. You cannot remove the **Home** shortcut that leads to the project Home page.
- Change the order of existing Favorites using the ▲▼ buttons.
- Click Edit to rename or change a Favorite's icon.
- Reset your shortcuts to the original defaults.
- 1. Click on the upper-left corner of the Navigation panel.
- 2. Click Manage Shortcuts. The Manage Shortcuts dialog box opens.



- 3. On the Favorites tab of the Manager Shortcuts dialog box, select a Favorite from the list of current Favorite shortcuts. Do not select the fit Home shortcut.
- 4. You can modify the Favourite shorctut in the following ways:
  - Click **Remove** to remove a selected Favorite.
  - Click the Av buttons to move a selected shortcut up or down the list.
  - Click Cancel to cancel changes.
  - Click Reset to revert to the default shortcuts.
  - Click **Save** to save changes.

#### Tip

You can reset your user interface to the original state (the state when you logged on for the first time) on your user account page. This restores display of confirmation dialogs if you turned it off.

## **Dashboard**

## Work with the Dashboard

# **ALM**

### Scope(s): Repository, Project group, and Project.

The Dashboard topic of Navigation is useful for managers and other stakeholders who need to keep an eye on the status of development, from single projects, to divisions or departments, on up through the enterprise. The topic provides dashboards in several information scopes that roll up actual progress data maintained in the repository as people begin and complete tasks. The data is presented in charts, graphs, and other meaningful ways that help people understand the true status of development.

The information in the various dashboards and dashboard components can be as current as you need. Updating the dashboards is generally done with a scheduled job on the server configured to run when the extra system resources needed for the recalculations do not have any significant impact on users of the Polarion portal. For example, the update job could be scheduled to run at night, and weekly, nightly, or whatever interval provides enough currency for your needs. It is possible to update most dashboard components explicitly to get up-to-the-minute information in real-time.

# **Repository and Project Group Dashboards**

## ALM

In the **Repository** or **Project Group** scope, the **Dashboard** topic provides several charts with key pieces of information about the status of all projects being managed with Polarion (Repository scope), or in the currently viewed project group. The default information is as follows:

- Work Items Trend: A chart that shows the trend of unresolved Work Items over time. If you see sharp deviations in the curve, this might signal that something is happening that needs attention. For example, if the trend is sharply up, there may be many new defects being reported. If sharply down, some project(s) may be ahead of schedule and perhaps more features or testing can be planned.
- **Test Coverage**: For Maven projects, this chart shows the ratio of lines of code covered by tests versus the total lines of code in all **Projects** or the currently viewed **Project Group**.
- Process Score: This table rolls up key information from the reports in the Quality topic, and
  provides links to details in that topic. In the Repository scope, or a top-level project group, links to
  projects and subgroups appear enabling drill-down to additional information. For an individual
  project group, links to the projects in the group appear, again enabling drill-down to that level of
  detail.
- **Top 5 Projects**: A table shows the Maven-2 projects (up to five) having the highest level of code reuse, and some basic statistics on the levels of reuse.
- **Top 5 Packages**: A table shows the most reused Java packages (up to five), and a count of the number of times each package is reused.

#### Access the Repository dashboard

- 1. Open the **Repository** (See Access Projects).
- 2. In the **Navigation** pane, select **Dashboard**.

### Access the Project Group dashboard

- 1. Open the desired **Project Group** (See Access Projects).
- 2. In the **Navigation** pane, select **Dashboard**.

## **Project dashboard**

# ALM

In the **Project** scope, the Dashboard topic provides information about the currently open project. The following default components appear:

- Several statistics:
  - **Build Frequency**
  - **Build Success Ratio**
  - Test Success Ratio
  - Test Coverage 0
  - Traceability Work Items to Commits
  - Traceability Commits to Work Items

## **Note**

The data in the following is limited to the current **Project**.

Work Item Trend: A chart that shows the trend of unresolved Work Items over time. If you see sharp deviations in the curve, this might signal that something is happening that needs attention. For example, if the trend is sharply up, there may be many new defects being reported. If sharply down, some project(s) may be ahead of schedule and perhaps more features or testing can be planned.

- **Test Coverage**: same as described for Repository and project group dashboards, except the information is limited to the current project.
- **Test Coverage**: For Maven projects, this chart shows the ratio of lines of code covered by tests versus the total lines of code in the current **Project**.
- Live Plan: A GANTT style chart of the project plan based on project team members, the Work Items assigned to them (or unassigned), and dates (including Time Points) to which Work Items have been planned.
- Quality Score Card: This table rolls up key information from the reports in the Quality topic, and provides links to details in that topic. The information scope is the current project.

Facts and Figures: This section rolls up some useful metrics from various sources which can
reveal even more about what's happening with this project including the number of open Work
Items according to severity, recent changes, and most active developer.

## Access the Project dashboard

- 1. Open the desired **Project**. (See Access Projects.)
- 2. In the **Navigation** pane, select **Dashboard**.

## Edit and update the Dashboard topic

## **Edit the Dashboard topic**

The Repository and Project Group Dashboards and Project Dashboard sections describe the default composition of the Dashboard topic in a default installation. You should bear in mind, however, that the topic is implemented as a Wiki page, and is modifiable by any user with appropriate permissions in the information scope. The actual content you see in the topic on your system could be somewhat or even significantly different from the default. The various components could be rearranged, or some could be removed altogether.

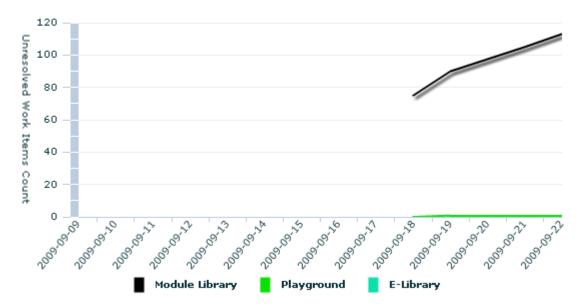
## **Update the Dashboards**

In a default installation, the system configuration contains scheduled jobs that recalculate the information displayed in the dashboards. An administrator can review these in the Scheduler topic of the Administration interface and optionally alter the default cron expressions that control the launching of the jobs.

Most components of the dashboards contain an **Update** link that invokes an immediate recalculation of the information displayed in the component to provide up-to-the-minute information. For example, to see the latest information in the Work Items Trend chart, you can click its **Update** link:

## Work Items Trend

This chart shows the trend of unresolved work items over time. Vertical axis represents a count of work items. A value is displayed for every project group and project in the scope you are currently working in (Repository or Project Group).



<u>Update</u> | Updated: 2009-09-22 14:46

## **Warning**

Recalculating dashboard information places some extra load on the server's resources. The most resource-intensive recalculation is the Live Plan. While Polarion is optimized to update as efficiently as possible, if the scope of the information is very large and the server has a lot of traffic, the update could take some time, and performance could be slowed down somewhat for all users.

## **Plans**

#### **About Plans**

The **Plans** feature enables teams to easily plan their work into releases. Releases can be internal or external. For example, they might be iterations or milestones which are internal to the development organization, and releases that are provided to customers and users.

Plans are created only in the project scope. They are often hierarchical. For example, a Plan for a release named Version 1.1 might have several child Plans for iterations such as i21, i22, and i23. Teams can use any desired naming convention for their Plans.

#### Plan basics

The III Plans topic in Navigation is the entry point to view and manage your release plans. Plans allow you to estimate work time, progress against each Plan, and can include multiple projects under a single Plan.

For each Plan, a team member or manager specifies the existing Work Items the team should complete between the Plan's start and end (due) dates. Estimated Work Items contain a value in the **Initial Estimate** field that indicates how long it takes to complete the item. The Plan aggregates and displays the total time of all initial estimates. The teams and managers then easily compare the estimated total time with the team's known capacity. For example, suppose the aggregated Initial Estimate for all the Work Items added to a Plan is 48 days and 6 hours (48d, 6h), and the period of the plan is 45 days (45d). Before work even begins, the team and managers can clearly see that the plan is overly ambitious and needs to be adjusted.

As work goes forward, charts in the Plan reflect the team's actual progress. Stakeholders clearly see at all times how much work is complete, and how much remains to be completed in terms of person days/hours. Teams and management are never left guessing about the probability of the Plan being completed by the planned Due Date. If progress is not what they originally expected, they can make intelligent decisions about adjusting the Plan, for example, removing some items from the Plan, or pushing the End Date forward.

By default, a Plan includes Work Items from one project. However, Plans can span multiple projects. For example, there might be separate projects for different components, all of which have the same release cycle. Therefore, Plans for iterations and/or releases need to include Work Items from all relevant projects. These projects can be specified in the Plan Properties.

### **Browse Plans**

# ALM REQ QA PRO REV

You can browse existing Plans and their properties. The **Plans** topic in Navigation shows the parent-child hierarchy of Plans in the current project. You can navigate directly to any Plan by clicking on any top-level Plan, or expanding it and an clicking on the node of any child Plan. You can also use the **Search** field in Navigation to search for a specific Plan.

The Plans page shows a tree table of existing Plans in the current project. The **Name** column shows the hierarchy of the Plans.

You can filter the table for a Plan name using the Visual Query Builder in the page toolbar. When the filter finds a Plan, any parent Plans are also shown. You can also filter by the Plan properties listed in the **FAVORITES** column, and/or the Plan fields listed in the **ALL** column of the Visual Query Builder. The instance of the Visual Query Builder on this page contains a limited set of query elements and element value options relevant for Plans.

When a Plan is selected in the upper pane of the table, the Plan details appear in the lower pane.

In the Plan detail you can:

- View the Plan's properties.
- Edit the Plan properties (if you have the necessary permissions).
- Run the actions on the Actions drop-down menu.

Each user can customize the Plan page tree table columns via the Customize Table item on the drop-down menu of the page toolbar (at the far right side) or by right-clicking the table header. The default visible columns are:

- Name
- Status
- **Progress**
- Due Date

#### Tip

Calculated fields in Burn-up and Burn-down charts can only be calculated from direct children. Multi-level calculation is not supported. Also, the relation of calculated fields in these charts cannot be based on other calculated fields.

When customizing the Work Items table, it is possible to use the **Planned In** field as one of the sort criteria.

#### Review a Plan

A Plan is a special type of page built with functionality that reads and processes data from the Work Items that are added to the Plan. This information is formatted in various ways in the Plan page to reflect the actual status and progress of the planned Work Items in aggregate, and the Plan itself. The information presented differs according to whether the Plan is for a Release or an Iteration. An Iteration Plan is more detailed, while a Release plan is broader in scope, rolling up information from all child Plans in its hierarchy.

You can review a Plan by selecting its node in Navigation under Plans, or in the tree table of Plans displayed when you click the **Plans** node in Navigation.

#### Review a Release Plan

To review a Release Plan, examine the Release Plan's four main sections:

- Plan Status: Shows at a glance the status of the Plan (Open, In Progress, Done), the Start and Due dates, and the amount of work done, still to do, and what the ideal progress should be at the time of the most recent data update. (Usually when the page was loaded in the browser).
- Items Overview: Shows a table reflecting a count of planned items currently in each of the project's Work Item statuses, for example, how many items are Open, Reopened, or Done.
- Child Plans: Shows a table of children of the current plan, if any, with a bar chart reflecting the current progress of each child Plan. **New Plan** enables you to create a new Plan in the current project.
- **Unresolved Planned Items**: Shows a table of all currently unresolved Work Items that are part of the current Plan. The table columns present useful statistics about each unresolved item.

The page section located in the right-hand column of the page provides a button set allowing any user with the required permissions to change the Status of the current Plan. Another button, **Open in Table**, is for invoking a special planning mode in which Work Items are located and selected for inclusion in the current Plan.

#### **Review an Iteration Plan**

You can review an Iteration Plan by viewing the major elements on the page of Plan iteration:

- Progress bar chart: Compare the actual progress of the team on the current plan (green) against the calculated ideal progress.
- Plan Board: View a tabular listing of all Work Items of all types that are planned for processing during the running of the current plan.
- Burn-down chart
- Burn-up chart

You can also view other elements including links to the following:

- Work Item Prioritization feature (to set priority for just the items in the current Plan).
- New browser tab in which you can browse just the items in the current Plan.

There is also a button set for setting the status of the current plan (Open, In Progress, Done).

See also Planning Work Items, Prioritizing Planned Work Items.

## Create a new Plan in a project

## **ALM**

You can only create Plans in a project; the global Repository scope is not supported.

- 1. Open the project in which you want to create the new Plan.
- 2. In Navigation, click III Plans.
- 3. On the toolbar of the **Plans** page, click **Create**.
- 4. Select the Plan type in the dialog box that appears. The default types are **Release** and **Iteration**.

#### **Note**

The types are the names of **Plan Templates**, which provide features and data storage for a particular planning use case. Project administrators can create custom Plan Templates with different names, so the selection may be different from the default.

If you are unable to create a new Plan and you have the correct license and permissions, it may be that the feature is not enabled. This can happen if your system is updated from a Polarion version that did not yet have the Plans feature. You will need to have your Polarion administrator enable the feature in Administration. See Enable Plans.

5. After specifying a Plan type in the next dialog box, specify a name for the new plan, and the parent Plan, if any. (You can choose a Parent plan from a list of open Plans.)

After creating a new plan, you may want to modify some properties settings. Once the properties are set, you the add Work Items to the Plan. Once you add Work Items, you may want to prioritize them.

## Set Plan properties

## ALM

After a Plan is created, any user with permissions to modify the Plan can edit the Plan's properties. Generally, Plan properties should be reviewed and set before Work Items are added to the Plan, and before the team begins work on it. This is especially important if the Plan spans multiple projects.

To access and edit Plan properties:

- 1. Click the **Properties** button on the Plan detail pane toolbar (or Plan page toolbar if you are not viewing the Plan in the table).
- 2. Click the **Edit** button to put the Plan Properties form into edit mode. Note that this may not always be necessary, as many of the property fields can be edited in-place. Such fields provide visual feedback when you hover over them.

#### **Basic properties**

Key properties to focus on include:

- **Start Date**: The date planned for the team to start working on the Plan.
  - The Burn-down and Burn-up charts do not render any information until the **Start Date** property is set. In addition, you must set other properties, add Work Items to the Plan, and start the Plan progress.
- **Due Date**: The data planned for the team to finish working on the Plan.
  - The Burn-down and Burn-up charts do not render any information until the **Due Date** property is set. In addition, you must set other properties, add Work Items to the Plan, and start the Plan progress.
- **Status**: The current status of the Plan.

Default statuses are **Open**, **In Progress**, and **Done**. Status values can be customized in the Project Administration.

 Capacity: The capacity of the team during the period beginning on Start Date and ending on Due Date.

The **Capacity** value can be hours or Scrum story points. The value is an important property to specify, and specify accurately. Polarion calculates the progress statistics of the Plan by comparing the Capacity value against the actual progress on processing the Plan's Work Items. (The Initial Estimate field of Work Items reflects the capacity needed to process each item.)

Started On: The date the team actually began working on the Plan.

It is important to set this field in order for the Plan to accurately report actual progress compared to planned progress. The value is set automatically when Plan status changes to **In Progress**.

Finished On: The date the team actually finished working on the Plan.

It is important to set this field in order for the Plan to accurately report how the plan actually progressed compared to how it was planned to progress. The value is set automatically when Plan status changes to **Done**.

• Parent: The ID of the Plan that is the immediate parent of the current Plan, if any. This property can be left empty if the current plan is a top-level Plan in a hierarchy of Plans, that is, it is not a child of some other Plan. The parent is usually specified when the Plan is created. However, if the parent was created later, or the structure of Plans has evolved, you may need to set it here.

If you are not sure about the ID value, open the parent Plan and invoke the **Open in Table** action. The ID is shown under the Plan title in the Planning sidebar.

### **Set Configuration properties**

You can set some properties in the **Configuration** of the Plan properties that control Plan content and calculations. You can set the following configuration properties:

Project Span: By default, a Plan is for the single project that its contained in and only Work
Items within that project can be added to it. Project Span adds Work Items to the Plan for
multiple projects.

Specify the projects to include in this field. Include the Plan's own project if it also contains Work Items to be added to the Plan. When a planner adds Work Items to the Plan, it is possible to select Work Items from all projects specified here. Planners are able to access the Plan and add Work Items from any of the projects specified, as well as the Plan's own project.

• **Allowed Types**: Restricts the type(s) of Work Items that can be added to a Plan. The list is dynamically populated with all types defined in the projects specified in Project Span (or the current project if Project Span is empty). By default, any of these types can be added to the Plan. You can restrict this to one or more types.

For example, if the Plan is for a bug-fixing iteration, this property could be set to the Issue type, which prevents the Plan from being populated with any other Work Item type besides Issues. If some Task items are processed in the same iteration, that type can also be specified along with Issue. Planners can then populate the Plan with only these types. They could not, for example, add Change Request items to it.

- **Prioritization Field**: Specifies which field should be used to store priority values for Work Items in the Plan when the items are prioritized.
  - By default, the standard Priority field of Work Items is used for prioritization. However, it is possible to define one or more custom fields in the project configuration, which can be used to store priority values for different kinds of prioritization. For example, there could be custom fields like Business Priority and Development Priority, with the latter created for use with Plans. For more information, see Prioritizing Work Items, and Prioritizing on a Custom Field.
- **Calculation Type**: Specifies the calculation the system uses to calculate and report the Plan's progress. The default is Time Based, meaning that values for Remaining Estimate and Time Spent are used to calculate Plan progress. By default, Remaining Estimate is calculated on items that have Resolution set, but still have a Remaining Estimate value greater than zero. If this is not the preferred behavior, an administrator can set the system property com.polarion.plans.calculateResolvedRemainingEstimate to false so that the Remaining Estimate value of resolved Work Items is not calculated. The best practice is to clear Remaining Estimate when the terminal state of workflow is reached. If Time Spent and Remaining Estimate are not filled for items planned to a time-based Plan, it behaves like a custom field-based Plan, and the "burn when done" principle is followed using either Initial Estimate (if filled) or the default estimate value as the Work To Do value of the Plan.

The other option is Custom Field Based. This means that a custom field is used to represent an estimate of work. Work done is calculated as the sum of such a custom field for all the items that are resolved. When this option is selected in the menu, the Estimation Field property becomes visible, and you can select from a list of currently configured Work Item custom fields of type *float* (the required type for this property).

- Default Estimate: Specify a value to use for Work Items' Initial Estimate field. Use as the default for any Work Items in the Plan that have no value set for their Initial Estimate field.
- **Previous Time Spent**: Stores the sum of the **Time Spend** field for a Plan that contains Work Items which has some time already spent. That is, the items contain a value in their **Time Spend** field. This property is updated automatically by the system and you should not normally need to edit it. You can find detail about the calculation in Previous Time Spent Property.

The Plan properties page also provides a field for a Plan description, and a section where any custom fields for the Plan, configured in Administration, can be edited.

### **Understand Plan actions**

The **Actions** menu in the Plan detail pane provides several functions you may need:

- Open in Table: Opens the selected Plan in the Table view, in a new browser tab. It also opens the Planning sidebar. You use the Planning sidebar to add the Work Items the team plans to process during the Plan time period to the project Plan.
- Prioritize Plan:

Opens those items in the Table view along with the Prioritization sidebar. This enables the Work Items to be prioritized, after work items are added to the Plan. For more information on prioritizing Work Items, see Prioritizing Work Items.

- Customize Plan Report: (ALM) Opens the underlying report Page of the Plan, enabling you to modify the report layout, modify existing Widgets' parameters, and add other Widgets to provide the functionality and information needed by the team.
  - If the Plan's report Page is shared from a Plan Template, you are offered the option to customize the report Page of the current Plan, or the report Page of the Plan Template. (You must have the necessary permissions in order to modify the template. The option is disabled if you don't.) After choosing an option, the report Page opens in the Page Editor. Modifying the Plan Template report overwrites the existing report, and all new Plans created from the Plan Template will have the changed report Page.
  - If the report Page is not shared from a Plan Template, no choice is presented, and the report Page of the current Plan opens in the Page Editor ready for editing. (See also Working With Pages.)
- **Delete**: (**ALM**) Invoke this action to delete the currently selected Plan. If the Plan is the parent of other Plans, the child plans are not deleted, and the name of the deleted parent Plan remains in their **Parent** property. No Work Items are deleted from the repository, only the Plan object.

#### **Note**

The Actions menu also has actions to export the Plan page to PDF, or to print it.

## Start and finish a Plan

# **ALM**

When work on a Plan actually begins, be sure to set the Plan status to **In Progress**, or use the graphical **In Progress** button on the Plan page. This sets the **Started On** property in the Plan properties to the current date.

When all work on a Plan is completed, set the Plan status to **Done**, or use the graphical **Done** on the Plan page. This sets the **Finished On** property in the Plan properties to the current date.

#### **Note**

While the ideal scenario is for the Started On date to be the same date as Start Date, and Finished On the same as Due Date, this may not always reflect how the Plan actually went. Work on the Plan might actually finish before or after the Due Date, for example. Reporting these dates accurately helps managers and team leaders to understand, over time, their realistic capacity and to better estimate what they can expect to deliver.

# Review a Plan's history

As with other artifacts, Polarion automatically maintains a change history for **Plans**. This history is accessible from the **History** button located on the toolbar of each Plan. Clicking this icon changes the page to show a listing of all revisions that currently exist in the repository. From there, you can view a specific revision, or compare any two revisions. The Plans topic is included when a project Baseline is created. When browsing a Baseline, you can browse Plans in the state they were at the time the Baseline was created.

#### View a Plan revision

You can view the current Plan in the state it was at some point in the past. While viewing the history. click the View a Revision button. On the next screen, specify the revision you want to view. You can use either of two picker icons to pick a revision date, or a Baseline. Then click the **Show** button to view the specified revision of the Plan. Note the following:

- If you know the revision number, you can type it directly in the field.
- If you use the Date picker (calendar) and pick a date in the future, the last existing revision is shown.
- If you use the Baseline picker, the list shows only Baselines in which the Plan existed. If the list is empty, the Plan did not exist when any Baselines were created for the project.

#### Tip

When you view a completed (finished) Plan, the page normally displays a link that leads to the Baseline that contains the historical revision of the Plan, in which the date in the Finished On field was set. If the Finished On date is changed manually in Plan properties, the link leads to the Plan revision when that date was updated. (If the history link is missing, someone has customized the Plan page and excluded it.)

### Compare Plan revisions

When viewing a Plan's history, you can compare any two revisions. Only changes to Work Items in the plan are compared and not changes to Plan properties. This means the comparison shows Work Items added to and removed from the Plan, and Work Item updates. Note that changes to Plan Properties are shown only on the History page, not in revision comparisons.

# **Create Plan reports**

You can create report Pages in your project to deliver up-to-date information about Plans to the team and stakeholders. Plans in projects based on one of the default project templates always include a preconfigured report Page for the Plan. You can customize this default report Page, and/or create additional report Pages for any Plan.

The Pages feature includes several Widgets for retrieving and rendering information about a Plan, and providing Plan-specific user actions. You can find them on the **Plans** tab of the **Widgets** sidebar in the Page Editor.

- **Plan Label**: Renders a label of a specific Plan, showing its due date and type via a colored icon. For each Plan, all its parents are shown to the highest level, as well as direct children. Finished Plans are shown in strike-through font. It is also possible to create new child Plans via the + button.
- Work Items Board: Renders a Work Items Board (sometimes called Scrum Board) for a specific Plan. It shows all the planned Work Items and their children in individual stages of development (grouped if the statuses are similar). By default it shows all child items that are linked with any role with a parent-child-role attribute. Users can select for which statuses the items are displayed compacted, to minimize the space used.

- **Plan Progress**: Renders a progress bar that visually shows how much work from the specified Plan is already done, how much remains, and what should be the ideal progress today to finish the entire Plan on time.
- Plan Burn-Down: Renders a burn-down chart for a specific Plan, with customizable time scale and labels
- Plan Burn-Up: Renders a burn-up chart for a specific Plan, with customizable time scale and labels
- **Plan Activity Stream**: Renders an Activity Stream showing activity limited to the Work Items planned in a specific Plan.
- Plan Status Button: Renders a graphical button set that enables a user to set the status of a specific Plan (Open, In-progress, Done). For a terminal status, the Finished On date of the Plan is shown.
- Open Plan in Table: Renders a graphical button that enables a user to open a specific Plan in the Table view of Work Items. The table is limited to the Work Items currently planned in the specified Plan.

#### Note

You can optionally create Plan-related reports using Classic Wiki pages. Syntax details and usage examples for Plan-related macros are provided in the **Plans** section of the embedded Wiki Syntax Help, accessible from the Wiki Mark-up Editor when editing Classic Wiki pages. Advanced users can consult the API documentation (Javadoc) for details on how to display various Plan statistics on Plan pages or in charts.

Alternatively, if you have an existing Plan or Plan Template with the Plan report implemented as a Classic Wiki page, the action **Switch to LiveReport Page** appears in the Actions menu on the Plan (or Plan Template) page. The source of the existing Classic Wiki page is appended as the last region in the new LiveReport Page. You can refer to it when customizing the new report Page, and delete the region when finished building the new Page. For more information, see **Build Reports**.

# Plan and prioritize Work Items

#### Plan Work Items

# **ALM**

After creating a new Plan and setting its properties, you need to decide which Work Items the team will process and complete during the period of the Plan, and add those Work Items to the Plan. We will refer to this activity as *planning Work Items*.

The main steps in the process are:

1. Access the Plan via the III Plans topic of Navigation.

- 2. Invoke the **Open In Table** action to open the Plan in the table view, together with the Planning sidebar panel.
- 3. Run one or more gueries to locate the Work Items to be added to the Plan.
- 4. Use control and widgets in the Work Items Table and the Planning Sidebar to plan Work Items by adding them to the Plan.

#### Prepare to add Work Items

When you have opened a Plan and are ready to plan Work Items into it, you need to open the plan in the Work Items Table view. There, you can run queries to locate the Work Items for the Plan, select them, and add them to the Plan. You can also modify the Plan, removing previously planned items.

- 1. You can open a current plan in the table by one of the following:
  - On the Plan page, click **Open in Table**.
  - On the Plan toolbar, click and from the list, select **Open in Table**.

A new browser tab opens, loading the Table view, creating and running a visual guery element to retrieve the Work Items for the Plan, and opening the Planning sidebar panel. You are now ready to begin planning Work Items into the Plan.

#### Find Work Items for the Plan

If no Work Items have been planned yet, the Table is empty. This is because of the visual query that is specified to retrieve the items in this Plan. You need to remove the guery element and build a new query to retrieve all or some of the items you want to add to the Plan. (For information on Work Item queries, see Search Work Items.) You may find it easier to build several simple queries rather than a single complex one. Each query should show some Work Items in the table, and at least some of those should be items you want in the Plan. (If none of them are, you need to revise your query.)

You can use the visual guery element Planned In to retrieve the Work Items that have been added (or NOT added) to one specific plan and aggregate the view with items from more than one plan. For more information on using the graphical Query Builder, see Construct Queries Graphically.

## Add items to the Plan

Once the table contains items you want to be processed during the Plan period, you can easily add them to the plan:

- 1. Select one or more items in the table by checking the box on the respective row(s).
- 2. You can add the item to the plan in several ways:
  - On the Planning sidebar, click **Add**. A count of items selected for adding appears
  - Drag a single or multiple selected items from the table to the planning sidebar.
  - Click the icon on the table row of an item you want to add to the Plan.

## Tip

- When items are added to the Plan, the plan Statistics are updated immediately in the sidebar panel. If you added some items by mistake, or you change your mind after seeing the statistics, you can remove one or more items from the Plan by selecting them in the table and clicking the **Remove** button.
- You can save the changes to the Plan any time during the planning process. Be sure to save changes when you are finished adding items to the plan.
- You can review the Work Items in any Plan by selecting the Plan in the Planning sidebar and clicking the (Show Plan Work Items in table) icon.
- You can return from the Table to the Plan by clicking on the Plan name in the Planning sidebar.
- You can navigate a tree of open Plans, and jump into any listed Plan from the Open Plans section of the Planning sidebar.
- When viewing an individual Work Item in the Work Item detail form, the **Planned in** field lists all Plans that include the item. The field is editable. You can remove a Plan from the field, which removes the item from the removed Plan. You can add another Plan to the field, which adds the current item to the selected Plan.
- You can optionally add a column Planned In to the Work Items Table. The column shows all Plans to which a Work Items is added. Clicking on any listed Plan opens the Planning Sidebar and selects the Plan. Because it is multivalued, the table cannot be sorted on this column.

You can also add Work Items to a Plan from the Document Editor. You need to manually open the Planning sidebar and select a Plan. Then click the icon in the Document margin beside any Work Item to add it to the Plan. Note that the type of Work Item must be a type that is allowed in the Plan by its properties. For example, if the Document contains Requirement items, but the Plan properties include only Change Request and Issue type items, you are not be able to add Requirement items to the Plan. You first need to modify the Plan properties to include the Requirement type.

#### **Prioritize planned Work Items**

# **ALM**

Setting priority values for Work Items in your Plan is optional. Assuming that you want to do it, you can use the same basic procedure as for prioritizing Work Items in general. Review Prioritize Work Items. Several issues that are specific to prioritizing the Work Items of a Plan follow:

#### Store priority data

By default, Work Items are prioritized using the standard **Priority** field of Work Items. You should decide whether this meets your needs for prioritizing Work Items in a Plan. For example, you may find that the standard field is fine for setting priority for a Release Plan, but your developers need to set

their own priority for an Iteration Plan. In this case, you might need to have your project administrator create a custom field for relevant Work Item types, with a name such as "Iteration Priority". (The field must be of type float.) This custom field should then be specified in the Prioritization Field property of the Iteration Plan(s).

## Open a Plan's items for prioritization

It is best if you have already added all the Work Items to the Plan that should be resolved during the Plan period. You can reprioritize the items any time, however. Let's assume you have the Work Items already in the Plan and you are ready to prioritize them.

On the Plan toolbar, click Actions Prioritize Plan. The Plan opens in the Work Items Table view, with the Prioritization sidebar open. All the Work Items currently added to the Plan are listed. If the plan spans multiple projects, the list includes items from the projects specified in the Plan Properties. The table of items is sorted top to bottom on the current value of the field specified as the Prioritization Field (see Store Priority Data). You will be changing this value for some or all of the Work Items during prioritization.

With only the Plan's Work Items in the table, you can change their priority by drag-drop repositioning in the table, or any of the other ways described in Prioritize Work Items. You can optionally modify the visual query as needed to limit the items in the table to one type, for example, Issues.

## **Work Items**

#### Work Items overview

The Work Items topic in Navigation provides access to the integrated bug (issue) tracker, one of the most widely and frequently used Polarion tools. The main components of the Work Items page follow:

- Availability: All products
- Scope(s): Repository, Project Group, Project

A Work Item is the Polarion term for an artifact of the development process that is waiting to be implemented, is in progress, or has been implemented.

You can review Work Items in any information scope (all items in the system, a group of projects, or one project). You access Work Items by first opening the scope you want to work with (for example open a project), and then selecting the **Work Items** topic in the **Navigation** pane. The **Work** Items topic contains the integrated **Tracker** feature. You can track any kind of artifact by defining a Work Item type for it.

# Work Item types

Typical Work Item types include: (Icons may vary.)

- Requirement
- Task

- B Defect

Work Item types are generally defined in the underlying Project Template from which a project is created. In any project created from a Project Template, the default Work Item types can be customized by a project or system Administrator, modifying existing types and/or new defining new types. For more information, see Configure Enumerations.

Each Work Item has a number of default data fields which are used to describe and categorize the item, assign it to someone, and incorporate it into project planning and tracking, set its status, and so forth. Custom fields can be defined for any Work Item type, enabling tracking of and querying on any kind of information needed.

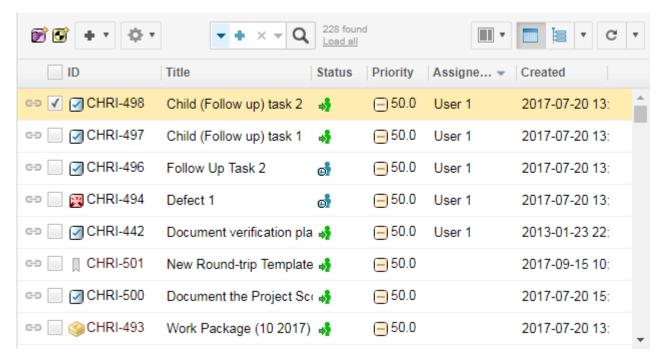
## Table view

By default, this view presents a list of Work Items as a table. The contents of the table varies according to the current query, shown in the Query Builder (see Search Work Items). By default, the table is sorted by Work Item ID in ascending order. You can sort the table on any column by clicking the column header. A Polarion administrator can customize the names and other properties of the columns, and add or remove columns. (See Configure the Work Items Table.)

When you select a Work Item in the **Table** view, the panes are tiled horizontally. The top pane shows a table of Work Items retrieved by the current query, and the bottom pane displays a multisection form providing access to the various data fields the item(s) currently selected in the table. Some of the key Work Item fields are editable in place, so it is not necessary to load the entire work Item in Edit mode in order to make some common edits (setting a new Status, for example).

By default, the IDs of Work Items in this view (and wherever they appear throughout the entire system) are rendered in different font colors according to the value of the Severity field. Items with high severity appear in Red font. Medium and low severity items appear in shades of gray.

The panes can be tiled horizontally or vertically via menu options located on the **Refresh** button's drop-down menu.



In products where it is supported, this view incorporates the Bulk Edit feature. You can select multiple items by checking the box next to the ID, and edits will apply to all the checked items. Export operations are not affected by the selection of items for Bulk Edit.

The Table view's toolbar displays some information about the results returned by the current query:

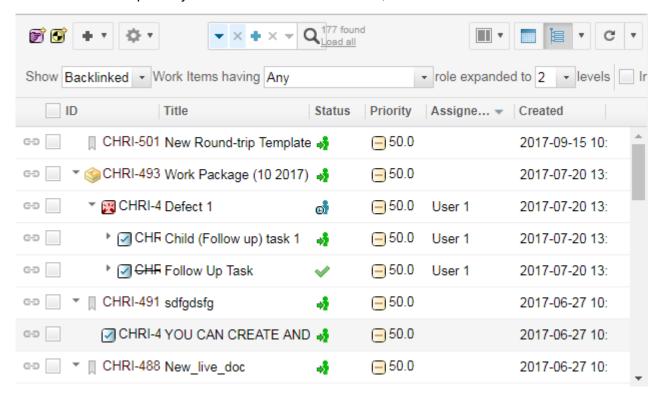
- Total number of items found by the current guery.
- Number of root items found and number of children (only when table is viewed using the Tree presentation option).
- Number of items currently loaded in the table.
  - Note that an administrator can set a limit in the system configuration on the number of items that can be loaded. The default maximum is 3000 items. The limit does not apply when the Tree presentation option is used to view the items.
- Load all link. This appears only if the number of items loaded in the table is less than the total number of items found and less than the maximum number of items allowed in the system configuration.

When the number of items to be loaded exceeds the configured limit, the link label reads Load first NNNN where NNNN is the configured limit on the number of Work Items that can be loaded in the table. (For example: "Load first 3000").

## Tree view

The Tree view is similar to the Table view, except the set of retrieved items is presented as a tree structure. Top level nodes are the items retrieved by the current query. If these items have other items linked to them, their nodes can be expanded to show the linked Work Items. If a child item has one or more items linked to it within the scope of the Depth query parameter, its node is expandable and the linked items appear as child nodes.

In the **Tree** view, the Visual Query Builder has several additional fields that enable you to specify the role(s) of linked items, the maximum number of link levels to search, the direction of linking, whether or not to include repository commits in the search results, and whether or not to filter all levels.



#### Caution

This view can require significant system resources to process, especially if the result set of the query is large and/or there are many linked Work Items. System administrators may impose a limit on the total number of items the query retrieves.

#### **Note**

When opening the table view of a LiveDoc with outline numbering enabled, **Outline**Number appears as the first column unless already defined in the table configuration.

To hide the **Outline Number** column, customize the table in **Table** view.

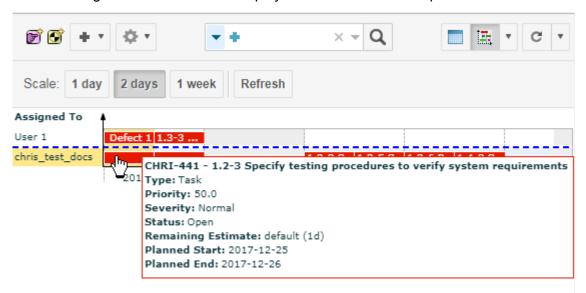
## Live Plan view

# ALM PRO

The **Live Plan** view presents the output of the Live Plan feature. It is a GANTT style chart showing what people have what tasks in some specific time period. Unlike other tools, you never update the project plan. Rather, you create constraints for the project, and set dates, time estimates, and personnel assignment for Work Items. The plan automatically updates as developers start progress

on, and close Work Items. The aggregate result displays in the Live Plan. (There is a repository-level aggregation of the Live Plan in the Repository Dashboard which sums up everything for all projects.)

The vertical axis contains people who have items assigned to them (there is also a common line, labeled with the name of the project, which contains unassigned Work Items). The horizontal axis contains time intervals. Nonworking time (as configured in the global Working Calendar and Working Calendars of users who have a project role) is visualized in the Live Plan. Work Items are represented as color coded blocks. Hover over any Work Item's block for a summary of information about the item. Clicking a Work Item's block displays its detail in the detail pane below the Live Plan chart.



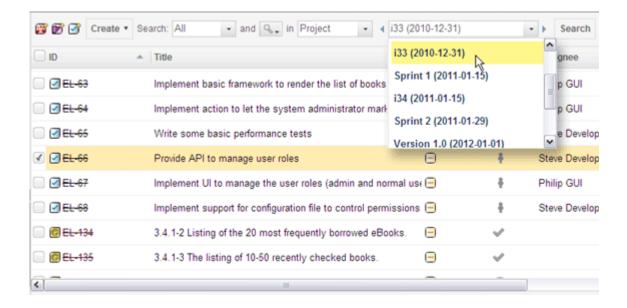
# Road Map view

(Scopes: Repository, Project Group, Project)

The Road Map view shows all Work Items in the current scope that are assigned a specific Time Point. You can select a Time Point in the appropriate combo box to display relevant Work Items.

As in the Table view, the Query Builder appears here, and you can click any Work Item to have its details display in the detail area where you can also edit the data.

In products where it is present, this view also incorporates the Bulk Edit feature. You can select multiple items by checking the box next to the ID, and edits will apply to all the checked Work Items.

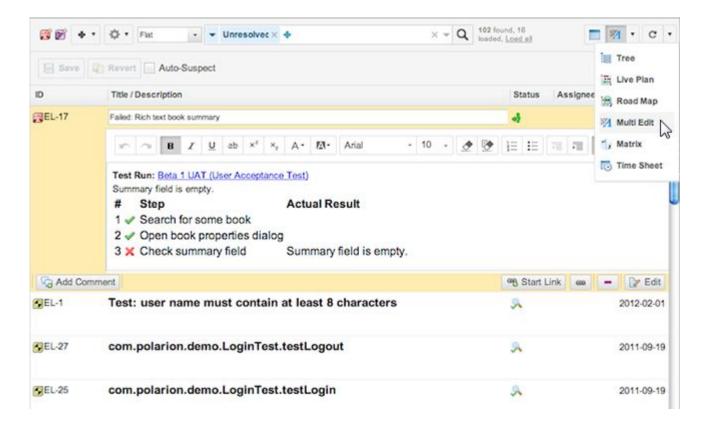


## Multi Edit view

# ALM REQ QA PRO

The Multi Edit view enables you to edit commonly edited fields of multiple Work Items without triggering Edit mode for each one. What Work Items appear in the Multi Edit view depends on the current query in the Query Builder. You can add a comment or edit any of the data fields for items in the page. When finished, you click Save to update all the edited items in one operation. Changes made to an item apply only to that item.

The Multi Edit view has two display modes - flat, and hierarchical. You can switch between the presentation modes in the drop-down list to the right of the button. Note that the Tree presentation option is only available when the displayed data is filtered for a single work item type by selecting one of the types in the Navigation pane under the Work Items node. In Tree mode, the Query Builder has several additional fields that enable you to specify the role(s) of linked items, the maximum number of link levels to search, the direction of linking, whether or not to include repository commits in the search results, and whether or not to filter all levels.

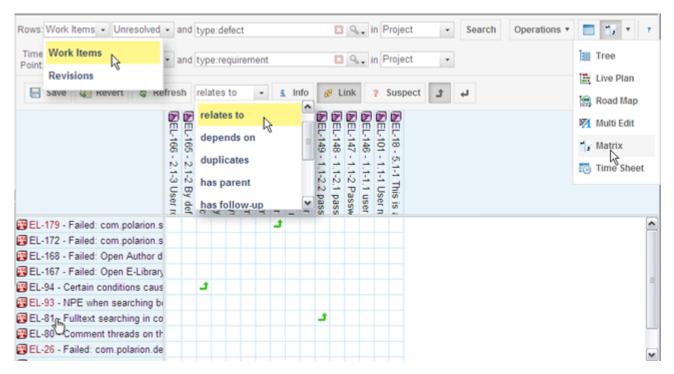


# Matrix view

# ALM REQ QA

The **Matrix** view is an interface for Polarion's traceability and impact analysis features. To establish traceability, you must create links between Work Items, and/or between Work Items and repository revisions. This can be done by editing individual Work Items as they are processed, or you can create links for any number of Work Items using the **Matrix** view.

The view presents a grid of Work Items. The set of Work Items (or revisions) that appears in each axis is determined by a query in the **Search bar**. In the context of the Matrix view, the set of Work Items contains two sets of controls. One control is an upper set for the horizontal axis; the other control is a lower set for the vertical axis. The intersection of the axes represents an actual or potential link between the two items. If no link exists, the intersection is empty. If a link does exist, an arrow symbol appears indicating the direction of the link (for example., from source, to target). You can click on the link symbol to pop up a box with details about the linked items. You can navigate directly to any of the linked items by clicking on the item name in the Link Details popup.

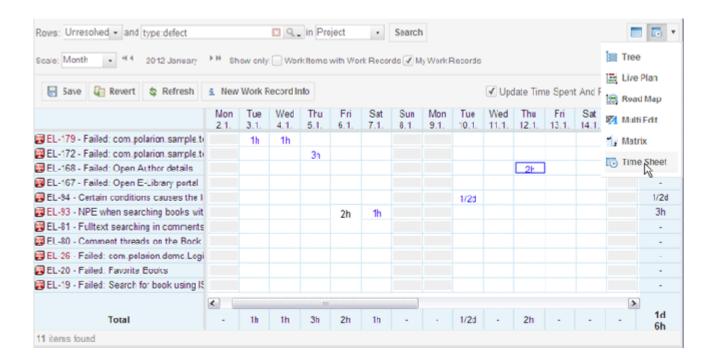


There are two properties in the polarion.properties file that define the maximum number of cells in a Traceability matrix or a Variants comparison matrix.

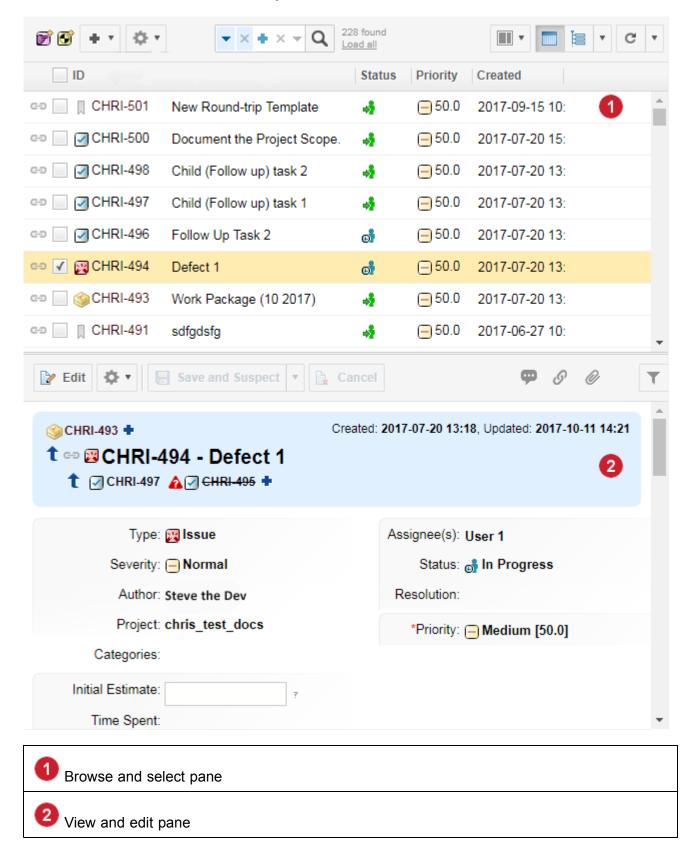
# Time Sheet view

# ALM PRO

With the **Time Sheet** view, team members can review and report the time they spend on Work Items during a period of time which they specify in the relevant field. They can also use the view to edit time already reported for individual items in Work Records. For more information, see **Use** the Time Sheet View.



# Work Item browser and editor panes



#### Work Item browse and select pane

The Work Item browse and select pane enables you to select a Work Item for reading, editing or some other applicable operation. Depending on the current view, you may be able to select multiple Work Items. The layout of this area (such as Table, Tree, and Matrix) varies according to the current view selected. (In products with the Bulk Edit feature, you can also select multiple items for bulk editing.)

#### Work Item viewer and editor pane

The Work Item viewer and editor pane displays the Work Item form providing access to detailed information about the Work Item selected in the current Work Item view. You can scroll, resize, or maximize the pane to browse all available information.

You can also reduce the number of fields shown using the T button on the Work Item toolbar.

You can edit the selected Work Item. Depending on your permissions and the form configuration, some fields may be editable in place without activating Edit mode with the Edit button.

The toolbar provides buttons for a number of common operations you can perform on the selected item(s), for example, editing. Other buttons provide access to common actions, and quick jumps within the form to commonly needed sections such as Comments, Links, and Attachments.

The level of detail displayed on a Work Item is switched using the T button on the Work Item toolbar.

Which fields are displayed or hidden are defined by an administrator in the Form Filters section of Form Configuration.

#### Tip

Why can't I edit? Your ability to edit Work Items depends on the role and permissions assigned you by the system/project administrator. If you do not have write permissions, the buttons and links that trigger edit mode do not appear, or are disabled in the interface.

#### **View Work Item History**

You can review the history of a selected Work Item by clicking on \( \bigcup \) **History** on the Viewer/Editor pane toolbar. You can drill down into any revision of the Work Item.

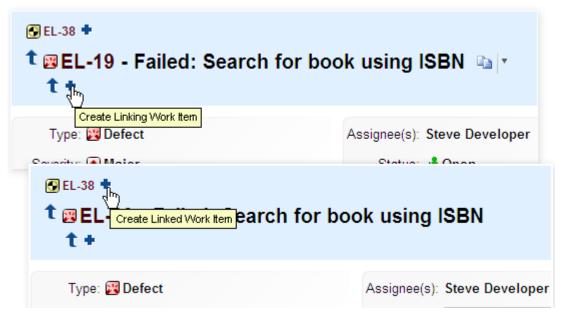
#### Link to another Work Item

You can quickly create a link to another Work Item by clicking icons that appear above and below the title in the Viewer/Editor. You can set different relationships with the current Work Item depending on which icon you click:

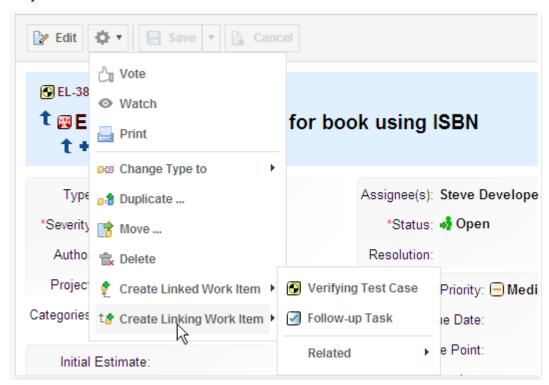
- **Linking**: The Work Item you select, will have a parent link relationship to the current item.
- **Linked**: The Work Item you select will have a child link relationship to the current item.

Any Work Items already linked to the current one display above or below the title, according to their relationship to the current item.

You can quickly create links to another Work Item using the Quick-create Linking and Linked Work Items.



Or you can use the Actions menu.



You can use icons to create links to other Work Items with different relationships.

# **Import Work Items**

#### **Import Work Items overview**

Work Items defined externally can be imported into Polarion and subsequently managed throughout the application lifecycle. This capability is particularly useful when you have existing document-based

assets containing artifacts (requirements, test cases, etc.) that you need to leverage quickly and easily when adopting Polarion. You can import from the following sources to create new Work Items in Polarion:

- Microsoft Office Word
- Microsoft Office Excel
- Microsoft Project

# **Import from Word**

For information, please consult the following topics:

- Importing Microsoft Word Documents.
- Sharing Documents using Word Round-trip and Re-importing Shared Documents.

## Import from Excel

Microsoft Excel workbooks containing artifacts such as requirements, test cases and defects can be imported to Polarion to create tracked and managed Work Items of the respective type. During import you have the option to create Work Items in a Document or as items in the project's tracker. After import into a Document, you have the option to use Word Round-trip to collaborate with external stakeholders who use Microsoft Word. After import to the tracker, you have the option to use Excel Round-trip to collaborate with external stakeholders who use Excel.

## **Preparing to Import**

Before beginning the import from Excel process, there are several things you should do or check:

- Check the configuration of the project into which you will import. Verify that the Work Item type corresponds to the type of artifact you are importing. If necessary, customize the configuration to define a corresponding Work Item type. For example, if the Excel document contains test cases, the target project's configuration should define a Work Item type corresponding to test cases.
  - Work Item types are defined in **Administration**: Work Items: Types. If you don't have access to Administration, you can see the currently configured types by opening the project and expanding the Work Items node in Navigation.
- Familiarize yourself with the data fields of the Work Item type you are importing, including any custom fields that may be configured for project Work Items and for the type of Work Item you import. Look at an existing item of the type, or if none exist yet, create a new one in the Table view of the Work Items topic. Review the fields, but don't save the new item.
- Review the Excel worksheet containing the artifacts you are importing. Rows should correspond to individual Work Items. For example, if the sheet defines requirements, each requirement should occupy one row. (The worksheet can contain a header row.) Columns should correspond to Polarion Work Item fields. Import of multivalued data is supported for some Polarion multivalued fields.

See Import from Excel:Multivalued Data and Fields for more information.

Decide which Excel columns should map to which Polarion Work Item fields. If some columns have no corresponding Work Item fields and you want to store their data in Polarion, it is necessary for a Polarion or project administrator to add custom fields to the project configuration before importing. You are able to specify field mapping during the import process. For information on which Work Item fields are supported for Excel import, see Import from Excel: Supported Work Item Fields.

Identify the cell at which the import process should begin. You need to specify it in the import wizard. Normally this is the leftmost cell on the first row after the header row, if a header row exists. However, it can be any cell. For example, if you have previously imported artifacts from a worksheet, and new ones have been added which you now want to import, you will need to specify the first cell on the first row containing the added items.

- Check that images to be imported from the worksheet are in .gif, .png, .bmp, or .jpg format if you
  want them to appear inline with other content. Images in other formats (.tiff for example) may
  still import, but will be attachments (rather than appearing inline) and they will not appear in
  the import preview.
- Create Work Items with links to Work Items from another Polarion Project before importing. See Linking Work Items Across Projects for details.

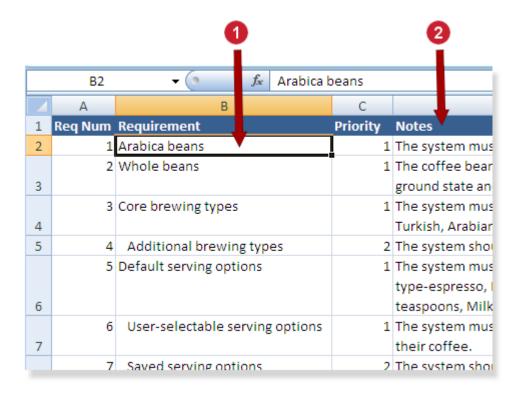
#### Import a Document

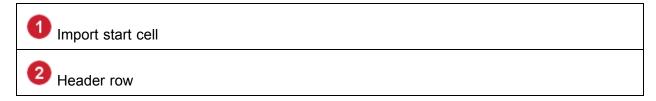
If your team prefers a document-centered approach to artifact management, you can import artifacts from Excel and store them in a LiveDoc Document. This is the preferred approach if you subsequently need to collaborate on artifact content with external stakeholders who use Microsoft Word. Your team will be able to use the Word Round-trip feature.

Import Work Items from Excel into a Polarion Document:

- 1. Open the project into which you want to import Work Items and in **Navigation**, select **Work Items**.
- 2. In the Table view, click and choose Import to launch the Import Work Items dialog box.
- 3. In the **Format** field of the dialog box, select **xlsx: Microsoft Excel Workbook**. In the **Configuration** field, optionally select an import configuration. (Import configurations contain rules that govern how artifacts are imported and mapped.)
- 4. In the **File** field, specify the locally stored Excel worksheet file you want to import and click **Start**. The first page of the Import Excel Workbook wizard appears.
- 5. In the **Import rows from sheet** field, select the worksheet that contains the artifacts you want to import in this operation. The list shows all worksheets in the specified workbook. You can only import one worksheet at a time. For example, if a workbook contains requirements in one sheet and test cases in a different sheet, you must do a separate import operation for each type.
- 6. In the **Begin import at cell** field, specify the cell at which the importer should begin importing artifact data. The importer examines the worksheet and suggests a cell. The cell you specify in this field should not be part of a header row. If the first column is some kind of artifact identifier, you can begin importing from the second column, as Polarion assigns its own IDs to Work Items created during the import process (see figure below).

Review the **Table contains header** field. The importer examines the worksheet and checks this field if the sheet appears to contain a header row. (See figure below for an example of a header row.)





- 7. Click the as new Document link to indicate that Work Items should be imported into a new Document. Two editing fields appear.
- 8. In the drop-down list control, select the space where Polarion should create the new Document, and then enter a name for the new Document in the field provided. Note that the Document name must be unique within the specified space. If your project is configured to support Document Types, specify which type the new Document will be.
- 9. Click **Next** when you are ready to continue. The field mapping page of the import wizard appears.
- 10. Map the worksheet columns to appropriate Work Item data fields. Use the Preview button to check the import result, modify mapping, create more import rules as needed, and optionally saving the rules you define for reuse with subsequent Excel imports.
- 11. Use the **Preview** button to preview how the Work Items looks in the Document after importing.

12. Click **Import** to import the Excel artifacts and create Polarion Work Items in the specified space and Document.

## Import to the Tracker

If your team prefers a tool-based approach to artifact management, you can import artifacts into Polarion storing them in the integrated tracker rather than in a Document.

Import Work Items from Excel into the tracker:

- 1. Open the target project and in Navigation, select **Work Items**. The Table view appears.
- 2. In the **Table** view, on the toolbar, click and choose **Import** to open the Import Work Items dialog box.
- 3. In the **Format** field of the dialog box, select **xlsx: Microsoft Excel Workbook**.

  In the **Configuration** field, optionally select an import configuration. (Import configurations contain rules that govern how artifacts are imported and mapped.)
- 4. In the **File** field, specify the locally stored Excel worksheet file you want to import and click **Start**. The first page of the Import Excel Workbook wizard appears.
- 5. In the **Import rows from sheet** field, select the worksheet that contains the artifacts you want to import in this operation. The list shows all worksheets in the specified workbook. You can only import one worksheet at a time. For example, if a workbook contains requirements in one sheet and test cases in a different sheet, you must do a separate import operation for each type.
- 6. In the **Begin import at cell** field, specify the cell at which the importer should begin importing artifact data. The importer examines the worksheet and suggests a cell. The cell you specify in this field should not be part of a header row. If the first column is some kind of artifact identifier, you can begin importing from the second column, as Polarion assigns its own IDs to Work Items created during the import process.
  - Review the **Table contains header** field. The importer examines the worksheet and checks this field if the sheet appears to contain a header row.
- 7. Click **Next** when you are ready to continue. The field mapping page of the import wizard appears.
- 8. Map the worksheet columns to appropriate Work Item data fields. Use **Preview** to check the import result, and modify mapping, create more import rules as needed, and optionally save the rules you define for reuse with subsequent Excel imports.
- 9. Click the **Import** button to import the Excel artifacts and create Polarion Work Items in the project tracker.

#### Field mapping and import rules

#### Caution

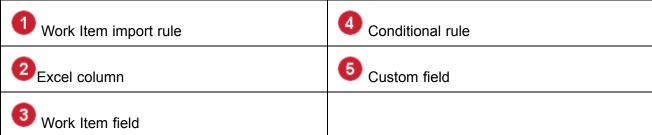
Before importing, link Work Items across projects to create Work Items with links to Work Items from another Polarion project.

After you specify the scope of the data to import, the second page of the import wizard presents an interface enabling you to:

- Map columns in the worksheet to Work Item fields.
- Specify one or more conditional rules that control how the data is imported.
- Reuse a previously saved import configuration that is, a combination of field mapping and conditional rules.
- Preview the result of the import with the current mapping and import rules.
- Save the current field mapping and rules as a reusable import configuration.
- Import the worksheet data and create Work Items from worksheet artifacts.

The page initially presents a new import rule with field mapping:





Start by specifying what type of artifact you are importing from the Excel worksheet. Select from the list of currently configured Work Item types. Next, specify the condition(s) for importing data as the selected type. You can add more conditions by clicking the • icon.

Next, specify how Excel columns should be mapped to Work Item fields. Proceed from left to right. Column headers are worksheet column names. Cells contain drop-down lists from which you can select a field name. It is not required to specify a field for every column. For example, if the worksheet contains a column with some sort of ID for an artifact, you can skip mapping it because Polarion assigns IDs to all Work Items created by the importer.

When you select some fields, they dynamically display fields for mapping date values in the worksheet to Work Item field values. For example, suppose your Excel sheet has a column **Priority** with numeric values, 1, 2, or 3 which actually corresponds to your Polarion project's **Severity** field,

configured with values **Must have**, **Should have**, **Nice to have**. When you select **Severity** in the **Priority** column, the importer detects the data values in the worksheet and presents a drop-down list for each value in which you can specify the corresponding **Severity** field value.

In many cases, a single import rule gives the desired import result. (Remember that you can use the **Preview** button at any time to preview the result of the import without actually importing. Preview shows only the first 100 matching items found in the sheet.) If a single rule is not robust enough for your needs, you can add more rules. To begin, click the • icon labeled **Add Work Item Rule**. Another set of condition rule plus field mapping is added to the page. Use it to add more parameters to the import configuration.

After **Preview** shows the import result you want, but before you click Import to trigger the **import**, consider whether you or other team members might need to import another Excel worksheet using the same rules and mappings you currently have specified. If so, click the gear icon in the header, choose **Save**, and specify the name and scope of the import configuration.

## Create and reuse import configurations

An "Import Configuration" is a saved import rule or set of import rules. Import Configurations can save significant time and effort when there is a need to import multiple similar Excel worksheets. Import Configurations can be saved in the global scope to make them available across projects, or in the project scope to confine their availability to a single relevant project.

Create a new Import Configuration:

- 1. Begin importing an Excel workbook.
- 2. Define the import rule(s) for an imported worksheet, using preview to verify the import result.
- 3. When you have the result you want in the preview, but before you click **Import**, click the drop-down icon (next to the workbook file name on page 2 of the wizard), and choose **Save** on the menu. This launches the **Save Import Configuration** dialog box with the **Create new configuration** option selected by default.
- 4. In the **Scope** field, specify the scope of the new Import Configuration (Global or Project).
- 5. In the **Name** field, specify a meaningful name for the new Import Configuration. This name will appear to users in the specified scope in the **Configuration** field when they import Excel workbooks.
- 6. Click **Save** to create the Import Configuration.

Reuse an existing Import Configuration:

- 1. Begin importing an Excel workbook.
- 2. In the **Import Work Items** dialog box, in the **Configuration** field, select the Import Configuration to use for the current workbook import.
- 3. Select the Excel file to import and click **Start**.

#### **Note**

You can modify the import rules loaded from the saved Import Configuration. You can optionally save the modified import rule as a new Import Configuration, or you can overwrite the existing Import Configuration with your changes (select the **Update existing** configuration option in the Save Import Configuration dialog).

See Remove Unused Excel Import Configurations for more information.

## **Import from Microsoft Project**

Past versions of Polarion have supported Work Item import from certain versions of Microsoft Project. This is considered a legacy feature, and later versions of Microsoft Project are not supported. For more information see Importing from Microsoft Project.

#### **Browse Work Items**

#### **Browse Work Items overview**

Every Work Item is part of a Project, and is planned and tracked with the Project. If a Work Item has dependencies with another project, it is possible to show this using the Work Item Linking feature.

You can browse the Work Items in the current project in different views in the Work Item page: Table, Tree, Live Plan, and Matrix. The views render different presentations of Work Items for different purposes such as task management, project planning, traceability linking and so forth.

To browse Work Items, select the Project in the **Open Project or Project Group** dialog box, select the **Work Items** topic in the Navigation panel and then select the desired view of Work Items from the view selector on the page toolbar.

When you select a Work Item, its detail data appears in the lower half of the Content Pane, and various titled sections in that pane display information about the Work Item such as priority, status, who it's assigned to, when it is (or was) started and when it will be (or was) completed.

#### **Sort Work Items**

When browsing, you can sort the Work Items displayed in the Table view of the Content Pane by clicking on the column header of the data column you want to sort. For example, click the header of the **Status** column to sort the items in the table according to the current status of the Work Items. By default, Work Items are sorted on ID.

#### **Customize the Work Items table**

You can customize the table of Work Items that appears in the Table, Tree, Live Plan, and Road Map views to show different Work Item fields as table columns. You can customize the table content for the current session, or create one or more saved column configurations which you can switch as needed. You can also change the default column layout for all users in the scope where you are working (project, project group, or repository), assuming you have the necessary permissions.

Administrators can adjust the maximum number of Work Items displayed in the tracker by adding the following property to the polarion properties file:

com.polarion.ui.table.hardLimit=X

(Where **X** is the maximum number of Work Items to display.)

The default setting in Polarion, (without using the above property), is 10,000.

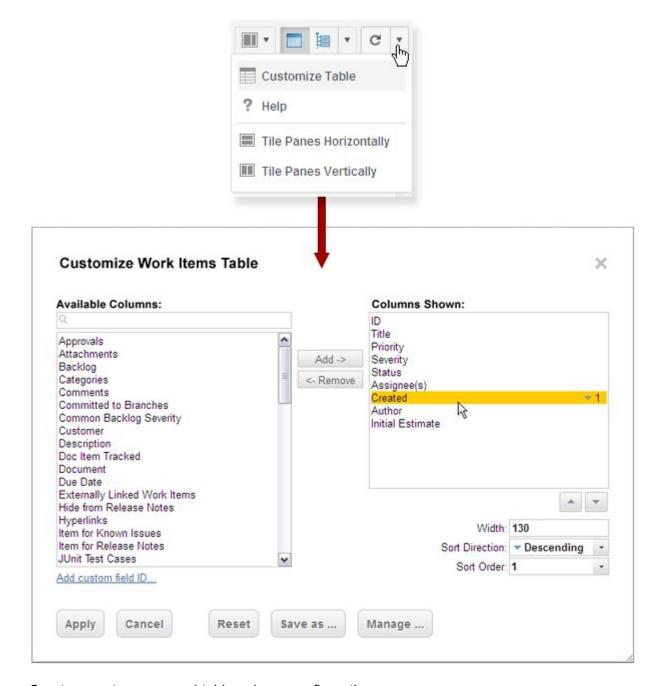
#### Customize the current session

To quickly show or hide table columns for the current session, right-click on any column name. A menu appears listing all currently-shown columns. Columns currently visible in the table show a check mark. Click on a column name in this menu to switch its visibility for the current session.

You can show more columns in the table for the current session by clicking **More** in the menu and adding columns from the **Available Columns** list to the **Columns Shown** list of the **Customize Work Items Table** dialog box.

# Create persistent table configurations

You can customize the Work Items table to show any set of columns in any desired order, specifying the sorting level and direction of any column. You can save any custom table configuration you create and reuse it any time. You can optionally save multiple named column configurations for different personal needs. If you have the necessary write permissions, you can save a custom table column configuration as the default for all users of the current project, project group, or repository, depending on which scope you are working in when you create the custom configuration.



Create a custom personal table column configuration:

- 1. Open the project, project group, or repository you want to customize.
- 2. In Navigation, select **Work Items**, and on the page toolbar select the **Table** view if it is not already the current view.
  - If you want to see the Table view of Work Items contained in a Document, open the Document and select **Table** on the **Views** menu.
- In the Table view, on the Page Actions menu, choose Customize Table (see screenshot above).

- 4. In the **Customize Work Items Table** dialog box, add columns from the **Available Columns** list to the **Columns Shown** list. You can also remove columns from the **Columns Shown** list until the list contains only the columns you want to show in the table when this columns configuration is invoked.
- 5. (Optional) Change the appearance order of columns in the **Columns Shown** list by selecting a column and changing its position in the list using the **Up** and **Down** arrow buttons (or keyboard shortcuts shown in tooltips). Top to bottom in the list corresponds to Left to Right in the table.
- 6. (Optional) Set properties for some or all columns. Select a column to show its properties settings.
  - Width: Displays the width of the column in pixels. Enter a whole number value such as 60.
  - **Sort Direction**: Specifies the sort direction of Work Items on one or more columns in the table. You can specify Ascending, Descending, or None. If the column is not included in the sort order, specifying the sort direction in this field adds the column to the sort order. To remove a column from the sort order, specify None.
  - Sort Order: Specifies the position of the sort order. Once a column is included in the sort, this field specifies its position in the sort order. Enter an integer value from 1 to the total number of columns in the Columns Shown list.
    - For example, if you wanted Work Items in the table sorted by Severity first and Priority next, you would specify **1** in this field for the Severity column, and **2** in this field for Priority column.
- 7. Click **Save As** to save the configuration of specified columns and column properties.
- 8. In the Save Custom Table Configuration dialog box, choose one of the following options:
  - **Default for all users in scope**: Saves your changes as the default table column layout and sorting for all users in the current scope (project, project group, or repository). Users are able to customize for the session or invoke their own saved column configurations.
  - Personal custom configuration: Saves your changes to a named personal table column configuration that you can invoke on the Work Items table any time. Specify a name in the Name field.
- 9. Click **Save** to complete the customization operation.

## Tip

If the table-setting.xml configuration file is customized externally, and settings in the **Customize Work Items Table** dialog box are also changed, the settings from table-setting.xml are not reflected in the table. Only the settings in the dialog box are reflected. To also reflect settings from the configuration file, click **Reset** in the dialog box.

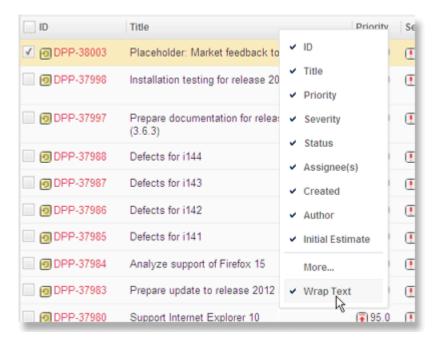
# Wrapping text in table cells

The **Description** field of Work Items can contain tables. By default, the text in table cells does not wrap and the text in some cells may appear truncated when you browse the description. In this case you can switch the Wrap Text feature of Work Item tables.

1. In Navigation, click **Work Items** and select Table view.

If you are viewing a Document on the Document Editor toolbar, click the **Views** menu and choose **Table**.

- 2. In the **Table** view of Work Items, right click on any column header in the upper part of the table of Work Items to display the Columns menu.
- 3. On the Columns menu, choose **Wrap Text**. When text wrapping is active, a check mark appears in the menu items.



#### Switch interface views

# **ALM**

The following must also have at least one of the ALM licenses:

# PRO REQ QA REV

In many development teams, as a team member you may work in different contexts at different times. You may sometimes function as manager, and other times as developer or a tester. In these different work contexts, it can be useful to only view and access what is most needed for that role, and have everything else out of the way temporarily.

Your system or project administrator can work with teams to configure interface Views for different work contexts that adjust the user interface to show only what is needed for each work context.

The **Views** list appears in the Navigation pane's Tool view when you are logged on (click the Tool icon in the upper left corner of the pane). When you first log on, the default interface View is active, and you have access to all topics and Work Item fields allowed by the configuration. In the default configuration the list provides several other interface Views that you can switch to as needed:

- Requirement Engineer
- Developer

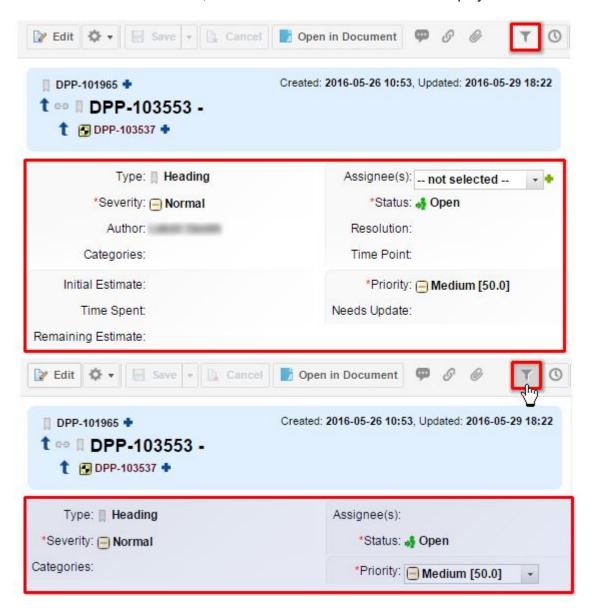
## Manager

If you see different names in the **Views** list, keep in mind that the list may have been reconfigured by your administrator for your Polarion system or the particular project you have open.

Each interface View can show its own set of topics in the Navigation pane. Each interface View can also show a different set of fields on the Work Item detail form, a different layout of the visible fields, and can control which fields are editable. What each interface View presents, and how, is controlled by the View configuration which can be customized by an administrator. For information, see Configure interface Views.

# Switch the Work Item detail level

On the Work Item toolbar, click T to switch the level of detail displayed on a Work Item.



Which fields are displayed or hidden are defined by an administrator in the Form Filters section of Form Configuration.

#### **View linked Work Items**

The Work Item details panel includes a section titled **Linked Work Items**. This section displays a listing of Work Items of all types that are linked to the Work Item you are viewing. If your installation is licensed and configured to use multiple repositories, the listing also displays any linked items that reside in a different repository from the one you are using. You can open any item listed by clicking on its title in the listing. (You may not be able to open a linked item in a different repository if you do not have access permissions for the other repository.) To open all items listed in this section which are hosted in the current repository, click the small icon in the lower right of the section footer (tooltip: Browse all items on this server in the Table tab).

Linked Work Items are also displayed above and/or below the title of the current Work Item in the viewer/editor. You can easily see all the incoming and outgoing links. Clicking on any linked item opens it in your browser.

See also Linking Work Items.

#### Search Work Items

#### Search Work Items overview

Polarion has extremely powerful Work Item query capabilities, thanks to its integration of the Apache Lucene guery engine. Lucene is mainly a command-line tool. Users who know Lucene guery syntax and Polarion's Work Item fields and variables can run command line queries in the Work Item Query Builder. You can user the Query Builder's graphical assists to easily build simple and complex queries to search for and retrieve the Work Items you want in any view of the Work Items topic, as well as in dialog boxes such as the Work Item Picker where you query for Work Items.

#### Tip

You can also use the Query Builder to search for Documents, Pages, Plans and Test Runs.

- Construct queries graphically
- Save and share queries

#### Construct queries graphically

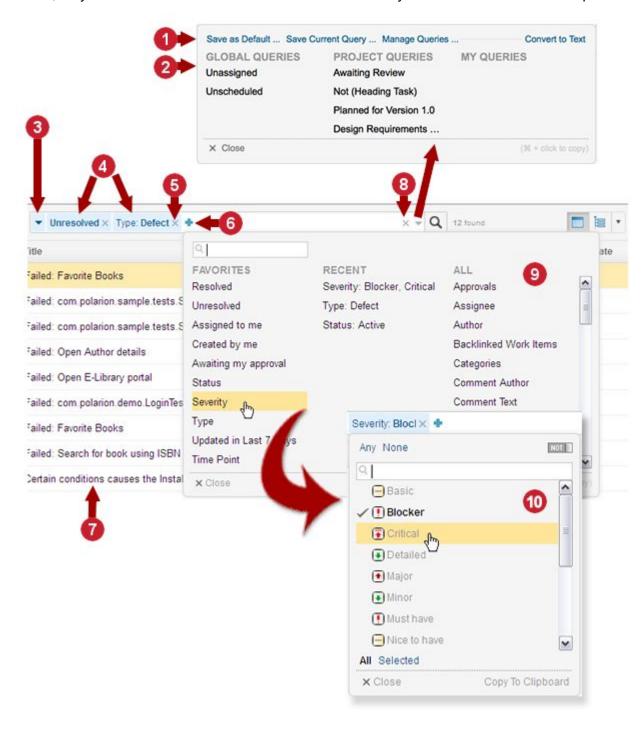
Every view in the Work Items topic contains the Query Builder in the toolbar or header. You construct queries by visually adding query elements to the Visual Query Builder. Query elements visually represent parameters of the underlying command-line syntax.

When you enter the Work Items topic, one default query element is shown in the Visual Query Builder: Unresolved. This element, alone, retrieves all unresolved Work Items. To get the results you want, you can either keep this element and add more to refine the results, or remove it and start adding different elements.

Click the \* inside a query element to remove it.

Click the in the query builder to add a query element.

When you click • to add an element, a pop-up bubble shows you three lists: often-queried fields (Favorites), recently queried fields (recent), and all fields, including custom fields. Fields are listed by name, so you do not need to remember or use field IDs as you must for command-line queries.



1 Actions	6 Add element
2 Saved queries	Work Item Table view
3 Element list	8 Clear all
4 Query elements	9 Select field
6 Remove element	Select field values

## Tips and tricks

- You can clear all visual query elements or command-line text from the Visual Query Builder by clicking the X icon near the right-hand end of the tool.
- Close element pop-up bubbles by clicking anywhere outside them.
- If there are many visual query elements and some are scrolled out of view and/or condensed, you can see the condensed names and/or access and edit them using the list of all current elements. To display the list, click the drop-down arrow on the left side of the query builder. You can also use the list to remove any query element from the Query Builder.
- The Recent section in the query element pop-up displays a list of the fields and field values you selected most recently. The items in this list can help you create often-needed queries and eliminate the need to save them.
- You can convert any graphically constructed query to command-line text by clicking the drop-down arrow at the right-hand end of the query builder and selecting the **Convert To Text** option. This lets you build queries visually and copy the syntax to use it elsewhere, for example, in the Query parameter of a LiveReport Widget.
- You can use the Query Builder in command-line mode by removing all graphical query elements and typing in valid command-line query syntax.
- You can use both visual query elements and command-line text in the query builder. See Combine Text with Visual Queries.
- The URL displayed in your browser contains a parameter query, which contains the full command-line syntax of the current query in the Query Builder. You can copy this parameter and use it in the query parameter of LiveReport Widgets or other places where Lucene query syntax is required.
- The Visual Query Builder contains an element SQL, which can be used alone to filter the Work Items table by an SQL query only, or in conjunction with other visual query elements. For more information, see Use SQL Queries to Retrieve Work Items.

See also Advanced Work Item Querying.

## Save and share queries

Commonly needed queries can be saved and reused. Saved queries can be global in scope (that is accessible to all portal users), limited to one Project (accessible only to users who have a project role assigned), or limited to access by you only.

Save the current query in the Visual Query Builder:

- 1. Click the drop-down arrow at the right-hand end of the Visual Query Builder, and click **Save Current Query**.
- 2. In the **Save Query** dialog box, enter a descriptive name, and select whether the saved query should be accessible globally by all portal users (**Global**), project locally by users having a project role (**Project**), or only by you (**User**).

## Run a saved query

You can rerun any saved query to which you have access.

- 1. Click the drop-down arrow at the right-hand end of the Visual Query Builder.
- 2. In the pop-up bubble listing available saved queries, click on the name of the query you want to run.

#### Tip

You can copy the command-line syntax of a saved query to your clipboard by pressing your **Ctrl** or **Cmd** key, and then clicking the saved query name.

#### Create a personal default query

By default, Polarion runs a system default query specified in the system configuration whenever you access the Work Items table. You can override that default and set any query as your personal default query, to be run whenever you access the Work Items table. Your personal default query is also used wherever the system default query is normally run.

You can create different personal default queries for different scopes. For example, you can create one default query that runs when you are working in the scope of a project, and a different one when you are working in the scope of a project group or the Repository. If you switch to a project or project group for which you have not defined a personal default query, the system default query is applied.

- 1. Use the Visual Query Builder tool to construct the guery you want.
- 2. Click the drop-down control on the right-hand side of the query builder, and in the pop-up panel, click **Save as Default** (see the screenshot in the Construct queries graphically section).
- 3. Confirm the action in the dialog box that appears.

You can change your personal default query any time by repeating these steps with a different query.

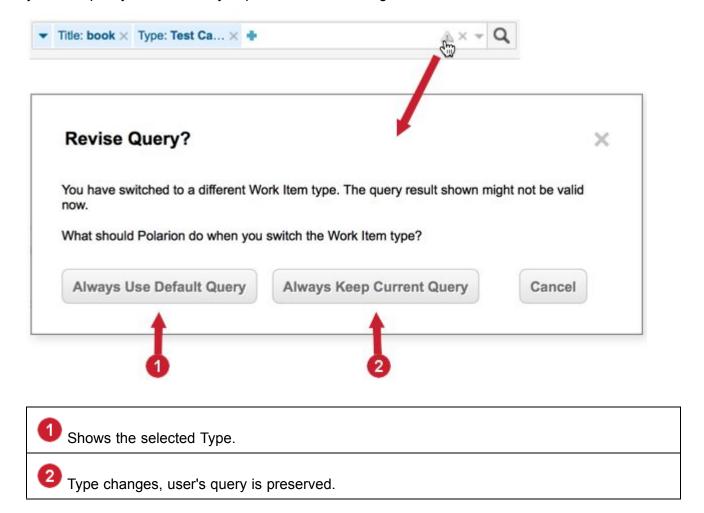
## Preference when switching contexts

Suppose you expand the **Work Items** topic in **Navigation**, and then select one of the types, for example, Requirement. Then, suppose you specify a query like **Status: Draft**, which filters the table

to show only Requirements items with Draft status. If you now select a different type in Navigation, for example, Test Case, by default your query Status: Draft is preserved and the table is filtered to show Test Case items with Draft status. This may or may not be what you want when you switch the context.

The same is true in a larger context switch. For example, you might be looking at the table showing Requirements with Draft status, and then switch to a different project. Or you might be looking at the Table view of a Document, and switch to a different Document. In all cases, your current query is preserved and the table in the new context is still filtered to show Requirements with Draft status. In the new context, the query results might not be what you need.

Whenever you switch to a context in which the current query's result may no longer be valid, a warning icon appears in the Visual Query builder. Clicking the icon displays a dialog box in which you can specify the behavior you prefer when switching contexts.



The dialog title and text vary depending upon the context switch (switching between Work Item types as opposed to switching to a different Document, for example), but the options are the same.

Always Use Default Query: When you switch to a different context, the default system query for the context is always executed. For example, when switching to a different Work Item type, the Work Items table always shows only Work Items of the selected type in the table. Any user-defined query present in the Visual Query Builder before the switch is deleted.

 Always Keep Current Query: The system preserves any user-defined query present in the Visual Query Builder before the switch. Be sure you are aware of the effect of the preserved query. You may want to modify or remove it after the context switch.

#### Create and edit Work Items

#### Create and edit Work Items overview

You can only create new Work Items if the relevant permissions are granted to your user profile:

- You must be granted the CREATE permission for Work Items in the project.
- If you are denied the MODIFY permission for any Work Item required field(s), you cannot create a
  new Work Item unless the project configuration specifies a default value for all such required
  fields. The restricted field(s) are assigned the default value in new items, and you cannot modify
  the field value in the new items. For details about this restriction, see Work Item Field Permission
  Restrictions and New Work Items.

Work Items can be created in the system in two ways:

- Manually in the tracker.
- Mark content in a Live Document.

If you find you cannot create a new Work Item, contact the administrator of your project or system and review your permissions.

See Duplicate to copy an existing Work Item from the Work Item tracker or a document.

#### **Create new Work Items manually in the tracker**

For information about creating Work Items from content in Live Documents, see Working with Documents: Work Items in Documents.

- 1. Open the project where you want to create a new Work Item.
- 2. In the **Navigation** pane, click the **Work Items** topic.
- 3. In the Work Items page, select the Table view. A table of the currently existing Work Items in the project appears in the upper half of the page.
- Click ♣, and select the type of Work Item you want to create from the drop-down menu.
   (Alternatively, you might use one of the Quick Work Item icons.) A new Work Item form appears in the Work Item viewer/editor, ready to edit.
- 5. Fill in the relevant fields for the new Work Item. Fill in all required fields (marked with an asterisk). Keep in mind that some fields may be hidden if the clicked. The **Create** button is enabled.
- 6. Add watcher and attachments as needed.
- 7. Click **Create** to create the new Work Item. The new item does not automatically appear in the Work Items table. Assuming the current search query retrieves it, you need to click **Search** in the table toolbar to refresh the listing and show the new Work Item.

#### Use the Quick Work Item icons

One or more "Quick Work Item" icons appear to the left of the Create (•) icon. These represent the first Work Item types defined in your system configuration. These are in the same order that appears under the **Work Items** topic. Click one of the icons to create a new Work Item of the respective type.

#### Tip

It is also possible to create a new Work Item using the Actions menu button (Gear icon) in the Work Item Viewer/Editor pane. Its menu contains special items for creating a new Work Item and linking it to some other existing one at the same time. For more information, see Actions.

#### Create new Work Items in Excel

New Work Items can be created in Microsoft Excel in an Excel document that has been exported from Polarion. See Using Excel Round-trip.

#### Create a Work Item in the browser address bar

You can also create a new Work Item by adding workitem?form mode=create to your URL address. The Work Item type and any string field can also be added to the address and their values will be autofilled when the new Work Item appears. All other fields must be entered manually.

#### **Example:**

```
polarionservername.com/polarion/#/project/mr demos/workitem?form mode=
create&form field type=defect&form field occurredInVersion=123
```

#### **Edit Work Items**

In order to edit Work Items, you must be granted permission to modify Work Items by the project administrator in the project's permissions configuration. Even if you have this permission, you still may not be able to edit some Work Item fields if the permissions configuration denies you the permission to modify some specific field(s). Such fields appear in the Work Item editor, but are read-only. For more information, see Configure User Permissions.

You can edit existing Work Items either singly or several at a time using (Bulk Edit). The latter is useful for such tasks as assigning multiple Work Items to the same person, or setting the same status, priority, or start/end date for several items.

To edit a single Work Item, select it in Table, Road Map, or Live Plan view, wait for the detail to load, and then click **Edit** in the Work Item Viewer/Editor toolbar. The **Edit** button only appears if you have edit permissions.

#### Work Item detail level

A Work Item can contain a number of data fields. You can minimize the level of details displayed using the \_\_\_\_ button on the Work Item toolbar.

When the \_\_\_\_ button is clicked, the detail level is minimized in the Work Item Viewer/Editor.

(Which fields are displayed or hidden can be defined by an administrator in the Form Filters section of Form Configuration.)

Click the button again to deactivate it, and view all the available details.

Polarion remembers your preference and displays all other Work Items with the same detail level.

#### Filter lists and menus

The Work Item Editor contains various fields with drop-down menus or lists from which users can select a value for the field. In some cases, the list of items can be quite long, even to the point of slowing page response while the list of items is populated. Long lists can also be difficult to browse. You can alleviate both these cases by filtering the list or combo box.

List filtering occurs as you type. After clicking the drop-down control, simply start typing the first few characters of the value you are looking for. The first character typed focuses the first item that begins with that character. The second typed character begins filtering the list incrementally as you type, eliminating from the list the items that do not match at all.

#### Tip

List filtering works not only in the Work Item Editor, but for all drop-down list/combo boxes everywhere in Polarion including dialogs. The default number of items loaded to lists is configurable by an administrator.

#### **Use Bulk Edit for multiple Work Items**

The **Bulk Edit** feature enables you to select multiple Work Items and edit, move, or delete them as a unit. When you save editing changes, values of modified fields are applied to all selected Work Items.

The set of Work Items available for editing varies according to the current Search query. You can use Shortcuts or saved or custom Search queries to isolate or narrow the subset of Work Items you want to edit all at once.

Bulk Edit is available in the following views in the Work Items topic:

- Table
- Road Map

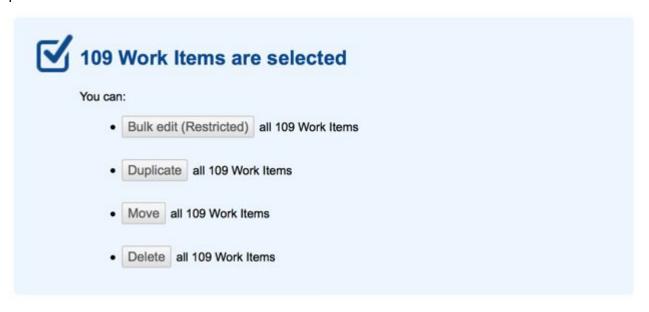
Select and edit multiple Work Items:

- 1. Make sure you are working in the correct Project, and in the **Work Items** topic in that project.
- 2. Select any Work Item view in which Bulk Edit is available, as mentioned above. A listing of Work Items appears in the upper pane. The Work Items that appear listed are retrieved by the current Search query.
- 3. Select the Work Items you want to edit using one of the following methods:
  - Use the Query Builder to run a query the result of which contains all the Work Items you
    want to edit. You can then quickly select all the items in the table with a single click on the
    check box in the Select column header. You can also individually select the check box for
    each Work Item you want to edit.

- Select the individual items from the initial Work Items list. A check box appears to the left of the ID of all the Work Items listed. Select each check box for the Work Item you want to edit.
- 4. In the Work Item Viewer/Editor (lower half of the Content Pane in the Work Items topic) edit and change field values as desired. (Some fields cannot be edited using Bulk Edit and are disabled. They need to be changed individually.)
- 5. Click Save and Suspect or Save to save your changes and apply them to all the selected Work Items.

# Bulk edit a large number of Work Items

The maximum number of Work Items that can be edited using normal Bulk Edit is set in the system configuration. If more than the number set by the limit are selected, **Restricted** Bulk Edit is activated. A panel of options appears and the word **Restricted** appears on the **Bulk Edit** button. You can edit field values in this limited view, but the current values are not loaded in order to optimize server performance for all users.



#### **Bulk Edit: disabled fields**

The following fields cannot be editing while using Bulk Edit:

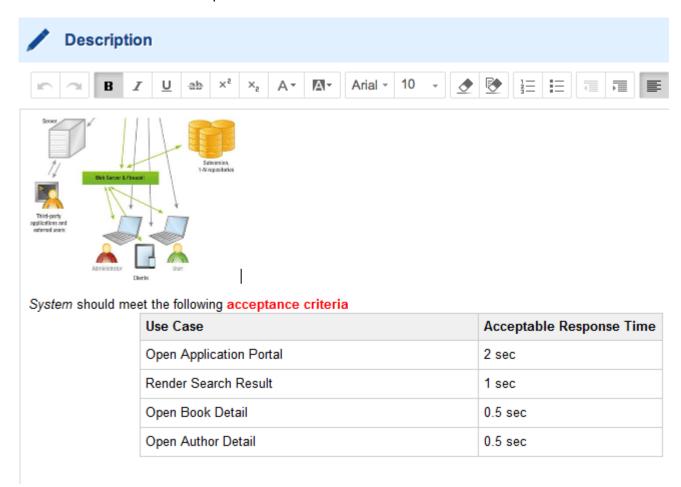
- Rich text fields, for example the **Description** field.
- All fields that contain multiple lines.

#### Warning

The number of Work Items you can modify via Bulk Edit is practically unlimited. Changing a large number of Work Items places an unusual processing load on the server and also results in large numbers of email notifications, with the resulting load on the SMTP server and the network.

# **Edit the Description field**

The **Description** field is a special field type that supports basic text formatting, images, tables, and hyperlinks. When you edit the field, a toolbar displays above the text field and provides access to the formatting and other features. The **Description** field is editable when the Work Item is in Edit mode. If the Work Item is not in Edit mode, you can edit just the **Description** by clicking the **Edit** button. Features of the description look as follows:



#### Format text

The following text formatting features are available in the **Description** field:

- B Bold font
- I Italic font
- Underlined font
- Strike-thru font
- x² Superscript font
- x₂ Subscript font

- A\* Text Color
- A\* Background Color
- Font family (drop-down menu)
- Font size (drop-down menu)
- Remove formatting
- Convert to Plain Text
- **≡** Bullet list (unordered list)
- Outdent text
- Indent text
- Left-aligned text
- Centered text
- Right-aligned text

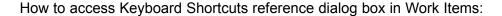
Text formatting features work in that same way as most word processors. For example, you can select some text and apply some formatting to just the selected text. Or you can position the caret and select formatting, and then everything that comes after has the specified formatting until you change it. The toolbar also provides the **Remove Formatting of Selected Text** button, which removes all formatting from selected text.

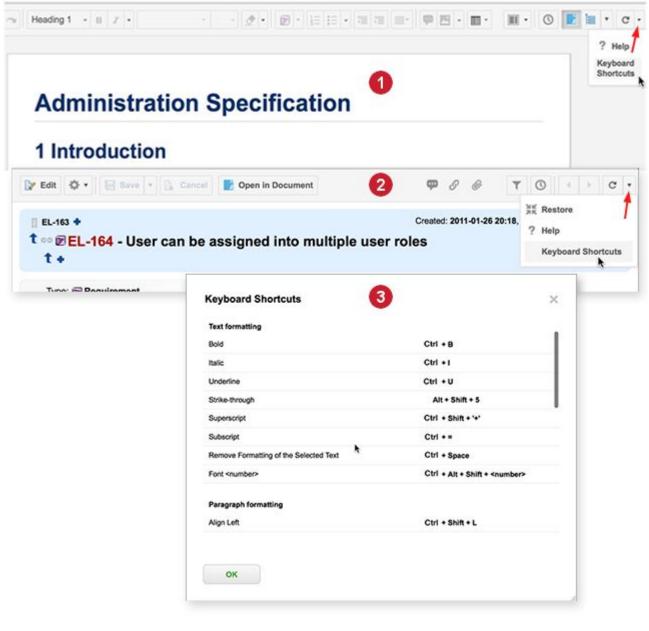
#### **Note**

- It is possible to configure Custom Fields with the same rich-text features as the **Description** field. For information, see Implementing Rich-text Custom Fields
- Available fonts can be configured by an administrator.
- Background color is only available for text. Background color of table cells cannot currently be formatted. Most browsers do not print text background color by default. If such printing is desired, it may be necessary to change the browser's print options or settings.

## **Keyboard shortcuts**

Keyboard shortcuts in the **Description** field (and other rich-text type fields) are similar to those of applications such as Microsoft Word and Google Documents. A comprehensive reference dialog box, context-sensitive for your operating system, is available via the Keyboard Shortcuts item on the Help menu of the respective editors.





1 Document editor.
2 Work Item details section.
3 Shortcut for the OS you are using.

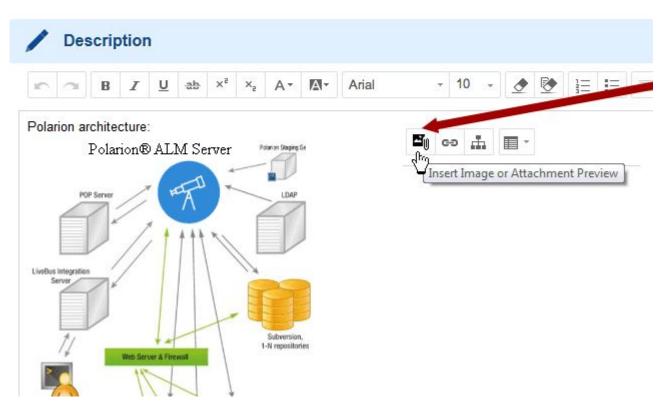
## Insert images or attachment previews

You can display images or attachment previews in the **Description** field of Work Items. (Or any other field of the Rich Text type). When you are focused on this field, the text formatting toolbar appears in the Work Item Editor toolbar. It contains a button that enables you to select an image or attachment file and insert it in the Work Item description.

After inserting an image or attachment preview, you can add a Caption above or below it, Resize it, or launch a full screen Preview of it.



## Categories:



You can insert an image or an attachment preview a Work Item's Description or Rich Text fields.

## Paste images from the clipboard

With some limitations (see Note below), you can paste an image from your clipboard into a rich text field. If there is a mixture of images and text in the clipboard, then only the name of the image file is pasted, and not the image itself. When you paste an image, an attachment is automatically created with file name <code>screenshot-[timestamp].png</code>, where <code>[timestamp]</code> is the date-time when the image was pasted. Supported image formats are: BMP, JPG, PNG, and GIF.

To paste an image from your clipboard, simply locate the insert point where you want the image to appear in the rich text field and use your browser's *Paste* keyboard shortcut or menu command.

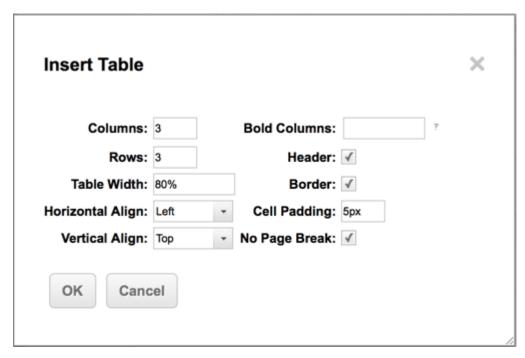
#### **Note**

The quality of the pasted image can vary depending on whether you copy the image from your file system, a web browser, a photo viewer, or image management program, as well as which supported browser you are using.

#### Insert and format tables

When the **Description** field is in Edit mode, the (table) button provides a menu with options for inserting and managing a table in the field. If the caret is not currently placed in a table, only the **Insert Table** command is available. All other commands are available when the caret is inside a table. When a table exists, you can:

- Insert columns and rows.
- Move columns and rows.
- Delete columns, rows, or the entire table.
- Access table properties.
- Add a caption above or below it. For more information, see Captions.
- 1. Make sure the field is in Edit mode and place the caret on a new line.
- 2. Click (table) on the toolbar and choose **Insert Table** from the menu. The **Insert Table** dialog box appears.



This dialog box is also accessible from Table Properties menu.

#### Note

If the Work Item you are editing is one that is stored in a LiveDoc document, the Table icon is disabled and you cannot insert any tables in the **Description** field.

3. In the **Insert Table** dialog box, set the properties for the new table as desired. (You can modify them later using the Table Properties command on the menu.) Most of the properties are self-explanatory.

The Bold Columns property enables you to reformat one or more columns to use bold font. For example, if you want bold text in the first column, enter 1 in this property.

The No Page Break property, when checked, prevents a page break from occurring in the table when the Work Item is exported to PDF or for Word Round-trip.

After inserting a table, click inside any cell to enter text. All of the **Description** field's text formatting options are available for text in table cells.

You can also insert images in cells in the same way as described in insert images.

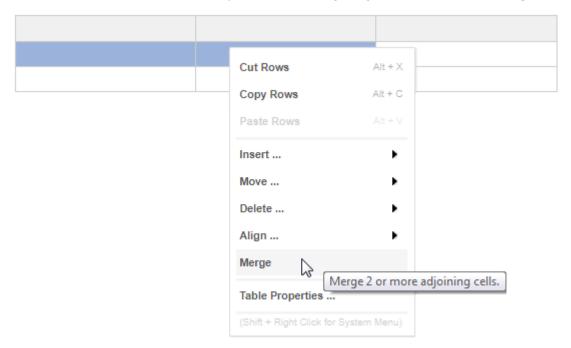
Table columns are self-sizing. You will notice some variation as you enter content in cells. Just enter your content in the cells, and the end result should look okay. If necessary you can increase or decrease the width of the table in the Table Properties.

Once inserted the following actions are supported:

- Copy Rows or Cut Rows.
- Paste Rows to the same or another table. Pasted rows insert above the current row. If pasted rows contain more columns than the target table, columns are added to the target table on the right.

Merge or Unmerge cells.

(Hold **CTRL**, select the cells that you want to merge, right click and select **Merge**.)



To unmerge previously merged cells, click on a merged cell, and then right-click and select **Unmerge**.

#### Merge / Unmerge in Description-based test cases

**Description** field-based Test Cases that use a table in the **Description** field as the source for the test execution extension are affected by the merge command. If you **Merge** the cells of a test execution table, the test execution extension does not work. Avoid merging cells in description based Test Cases.

#### Compare historical changes after using Merge

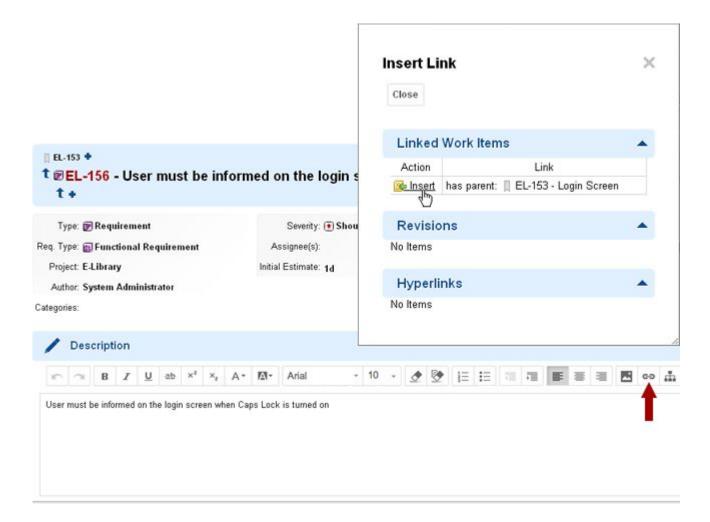
See Compare table changes when cells have been merged for details on comparing historical changes and how merging tables affects it.

# Insert hyperlinks

You can insert hyperlinks in the **Description** field of Work Items using the **Insert Link** button on the Work Item editor toolbar when the Work Item is in Edit mode. Links are local to the Work Item. That is, you can only create links to things that are linked to the Work Item:

- Work Items appearing in the Linked Work Items section.
- Revisions appearing in the Linked Revisions section.

Depending on the configuration of, and current selection in the interface list (in the Tool view of Navigation), these resources may not appear in the Work Item form for some users. Placing a hyperlink in the description provides exposure in the Work Item Description (always visible) of potentially hidden linked resources.



# Insert a diagram

# ALM REQ

Polarion provides the Diagram Editor tool for creating and editing diagrams. You can use it to create and edit diagrams in the Description field of Work Items in the Table view of the Work Items Tracker, or any Rich Text custom field.

For information on using this feature, see Diagram Editor.

#### Insert a mathematical formula

You can add **X** Mathematical Formulas to a LiveDoc or Work Item rich text field.

Block:

$$\sum_{i}^{n} x = 1249.4324^{2z}$$

Inline:

Selecting Inline lets you add additional content  $\sum_{i=1}^{n} x = 1249.4324^{2z}$  on the same line.

The formulas support LaTeX syntax.

#### Create and view merge requests

If a GitLab repository is configured and the field below is added to the Work Item form layout, GitLab merge requests can be done within a Polarion Work Item.

<extension id="GitLabExtension" label="Merge Requests" branchNameField="[id of the branch
name's custom field]"/>

## Create a merge request

- 1. Before creating a merge request within a configured Work Item, you should have already created your branch in GitLab and committed the changes to your local branch using git command line or GUI software.
  - Navigate to the local Git branch that you're working on.
  - b. Add, remove, or edit files and then save the changes.
  - c. (Optional) Type git status to see what files have changed.

```
$ git status
On branch doc_GITL-210_Upload_Documentation_PDFs_to_repository
Your branch is up to date with 'origin/doc_GITL-210_Upload_Documentation_PDFs_to_repository'
Untracked files:
   (use "git add <file>..." to include in what will be committed)

    Polarion_Enterprise_Setup_Guide.pdf
    Polarion_Install_Guide_linux.pdf
    Polarion_Install_Guide_win.pdf
    Polarion_Setup_Single_Sign_On.pdf
    Polarion_Trial_Guide.pdf
```

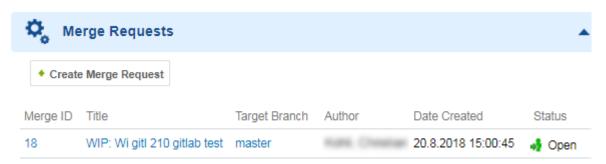
- d. Type git add . and press Enter.
- e. Type git commit and press Enter.
- f. Depending on your configuration, a text editor appears prompting you to add a commit message. Be sure to add the Polarion Work Item number to the commit message.

# Tip

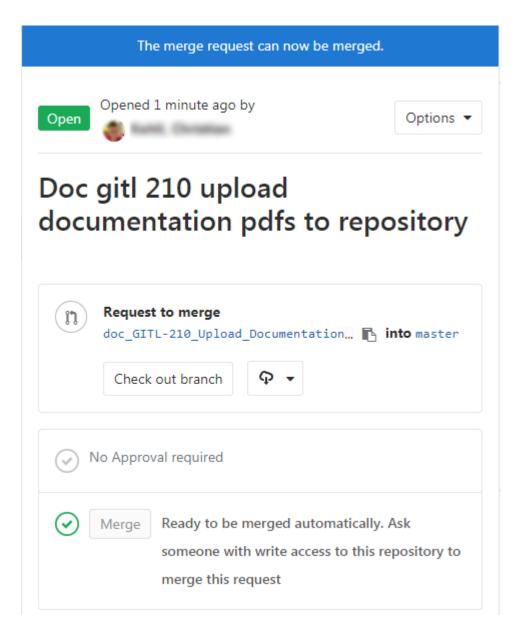
The **Copy ID + Title** command makes for a great commit message.



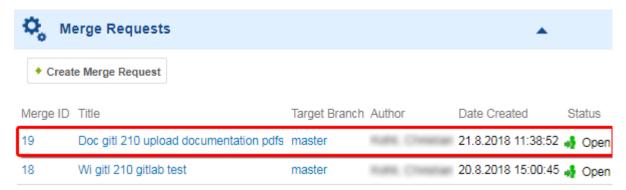
- g. Save and close the text editor.
- 2. Open the Work Item associated with the changes in Polarion.
- Scroll down to the Merge Request section.



- 4. Click \* Create Merge Request.
- 5. A GitLab **New Merge Request** page opens in a new browser window with the **title** and **Source Branch** preselected and the master branch selected as the **Target Branch**.
- 6. (Optional) Scroll down to the bottom and click **Commits** or **Changes** to review the changes.
- Click Submit merge request.
- 8. You are redirected to the merge request's details screen.
  - (Whether your merge request requires an approval or is merged automatically depends on how your administrator configured GitLab.)

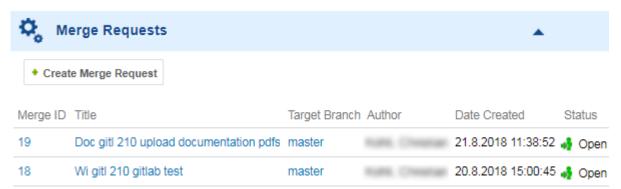


9. If you refresh the Polarion Work Item, the new merge request appears in the **Merge Request** section.



### View merge requests

Any GitLab merge request with a Polarion Work Item's number in its **Title** appears in the **Merge Request** section of the referenced Work Item.



- Click the **Merge ID** or **Title** to launch the merge request in GitLab.
- Click the **Target Branch** to launch the target branch's repository in Gitlab.
- Click on the **Author's** name to learn more about the GitLab user that requested the merge.

# **Using Multi Edit view of Work Items**

# ALM REQ QA PRO

The Multi Edit view of the Work Items topic also enables you to edit multiple Work Items, but the functionality differs from Bulk Edit. With Bulk Edit, you have a single editor pane and changes are applied to multiple Work Items. With Multi Edit view, all Work Items accessed by the current query are editable in the same page. You edit a single Work Item at a time, but all items are editable and changes to multiple items can be saved all at once.

# **Review and Approve Work Items**

### **Review and Approve Work Items Overview**

# **Approval Center**

Polarion supports a formal approval process for any type of Work Item through the Approval Center feature. The preconfigured project templates delivered beginning with version 2013-SR1 have preconfigured support for approvals.

The out-of-box approvals support delivers several features and capabilities for organizations that need a formal approval process with an audit trail.

# Why it's worth using

Workflow Integration: Approvals are integrated into the workflow of Work Items, including all types of requirements. Approvals can be added to the workflow of any Work Item type that does not already have it, including your own custom types. Integrating approvals into the workflow ensures that steps and people are not skipped or bypassed in the approval process.

Workflow also automates some tasks, such as designating users as Approvers and notifying them to review and approve items. For example, when a Work Item transitions from an initial draft status to a ready for approval status, a specific list of users, and/or all users whose user profile includes a certain role can be automatically designated as Approvers. These users' names are added to the Approving Users list in the Approvals field of the Work Item, and email notifications are sent to them.

- History and Audit Trail: An automatic, reportable history of the approval process is automatically maintained, so it is always be possible to find out who approved what and when.
- Easy for Approvers: The Approval Center is designed to facilitate users who are designated to review and approve or disapprove items:
  - o Links in email notifications lead users directly to items they should review and approve or disapprove.
  - The portal user interface makes it fast and simple for any user to access the items awaiting his/her approval.
  - o Approvers can approve or disapprove multiple items at once in a single operation.
  - o Approvers can discuss items with other stakeholders in threaded Approval Comments.
  - o Approvers can choose to work in a document interface (assuming the items are contained in a LiveDoc Document) or in the Tracker tool.
  - Organizations can obtain inexpensive Reviewer licenses for use by people whose only role in projects is to review and approve Work Items. The Reviewer license also enables users to review and approve Work Items on Apple iOS mobile devices using Polarion's ALM2GO app.
- Easy for Managers: The Approval Center also helps project managers, team leaders, and others to track and manage the approval process:
  - o Easily review, respond to, and resolve Approval Comments, as well as create new Approval Comments.
  - Hide or reveal resolved Approval Comments.
  - o Explicitly invite users to review and approve items via email or instant message.
  - o Quickly review the current state of the approval process for any item in real time.
  - o Reopen approvals for revised items that were disapproved, removing the disapproval state and starting the review and approval process again.

### **Getting Started with the Approval Center**

Before starting to use the Approval Center feature, there are some preliminary issues with which you should become familiar:

Roles and permissions: Important for project leaders and administrators. Make sure that all users
responsible for reviewing and approving Work Items have the necessary role assignments with
the necessary permissions.

- Workflow Configuration: Important for administrators and project leaders. Others may find the information useful to better understand underlying concepts, but it is not essential in order to find and approve the Work Items you are responsible for reviewing.
- Approvals Interface: Important for everyone involved in the approval process. You need to familiarize yourself with approvals-related user interface features in order to find, review, and approve/disapprove the items for which you are designated as an Approver.
- Require a signature when approving a Work Item: Polarion can be configured so that a Work Item's workflow status and the item itself require an electronic signature to complete the approval process.

# Roles and permissions

There are several user permissions that provide security for approvals.

Approve Work Items: To be able to approve Work Items, users must be assigned the project role project approver, and this role should be granted the permission named Permission to APPROVE/DISAPPROVE.

Approve as Different User: To allow users to mark items as approved or disapproved by another user (that is a different user other than themselves), an administrator can grant the Work Item permission named Permission to APPROVE/DISAPPROVE as ANOTHER USER.

Resolve Approval Comments: To be able to resolve Approval Comments, users must be granted Permission to RESOLVE COMMENTS.

For information on user permissions, see Configure User Permissions.

# **Review and Configure Workflow**

Your approach here depends on whether you need to run an approval phase for a new project, or in an existing project where approvals are not configured into the workflow.

# New Projects:

For new projects, it is recommended to create them using one of the project templates delivered with Polarion version 2013-SR1 (or later). Workflow in these templates is preconfigured to support a formal approval process. After creating the project, you should review the workflow for each Work Item type and make sure it is okay with your process. In particular, review the Actions section of the Workflow Designer page, looking at the Actions on each row in the table (see links below).

Polarion's project templates are designed to meet the needs of a wide variety or organizations. However, if the default workflow, including workflow statuses and actions, do not meet the needs of your project you either need to create a custom version of one of the stock project templates, or customize the workflows of the specific project. In this case, see these topics:

- Configure Work Item Workflow (See this topic if you find you need to customize any part of your project's workflow. In particular, see the subtopic Workflow for Approvals.)
- Customize Project Templates (See this topic if you find you need to customize your project's workflow, and the same customization is needed for all or many other projects.)

# **Existing Projects**

If you have an existing project created from a project template prior to Polarion version 2013-SR1, and that project still needs to go through a formal process of approval of requirements or other types of Work Items, you may want to customize your project's workflow to include or better support approvals by using the Approval Center feature.

The easiest approach is to create a sandbox project based on one of the new project templates, and in that project review the workflow configuration. You can then duplicate that configuration in your existing project. Be sure to edit the Actions and check for workflow conditions and functions, incorporating them "as-is", or tweaking them in your existing project. See Workflow Functions and Configure Work Item Workflow, especially the subtopic Workflow for Approvals.

# **Electronic Signatures**

In some industries you must meet strict regulatory requirements for traceability and be able to document when requirements were approved and by whom, ensuring that only authorized people can "sign off" on them. Your administrator can configure Polarion to support electronic approval signature that will satisfy many regulatory requirements in many industries. Be sure to check that electronic signatures for requirements approval are allowed by the body whose regulations you must meet. For information, see Require Signatures for Workflow Actions.

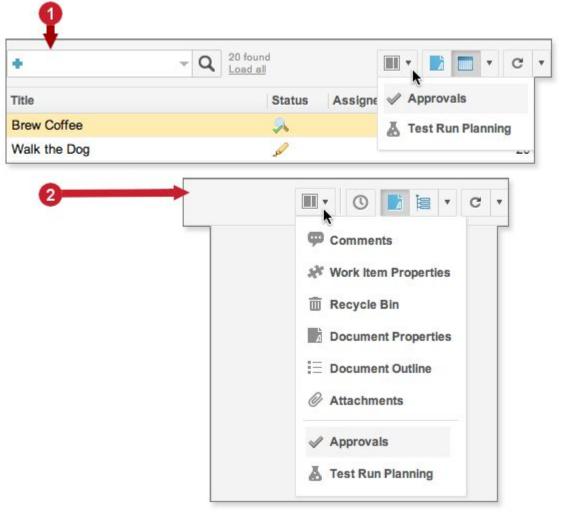
#### **Note**

If a Work Item that has been approved using an electronic signature is duplicated, the new Work Item does not show any indication of the source item's signature, and the new item is *not* approved or electronically signed. New duplicate items must be reviewed, approved, and e-signed.

### **Approval Center interface**

The main user interface component for reviewing and approving Work Items is the Approvals panel. In a Document, it is one of the panels available in the Document Sidebar. You can show and hide it with the Sidebar menu in the Document Editor toolbar. In the Table or Tree view of the Work Item Tracker, you can find it on the panels menu, on the toolbar next to the Query Builder. Refer to the following figure to locate the access control:

How to access the Approvals panel from a LiveDoc or the Tracker:

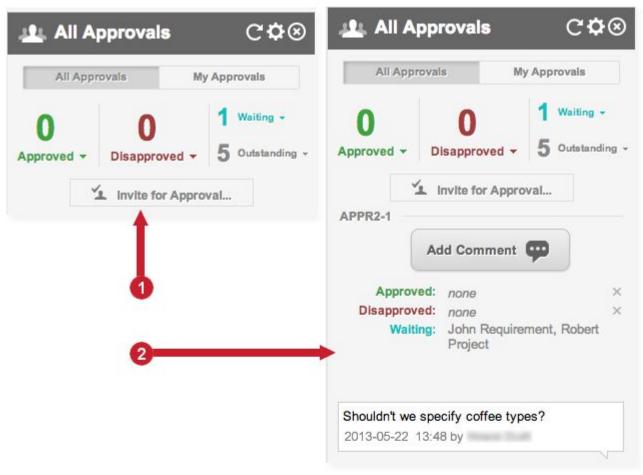


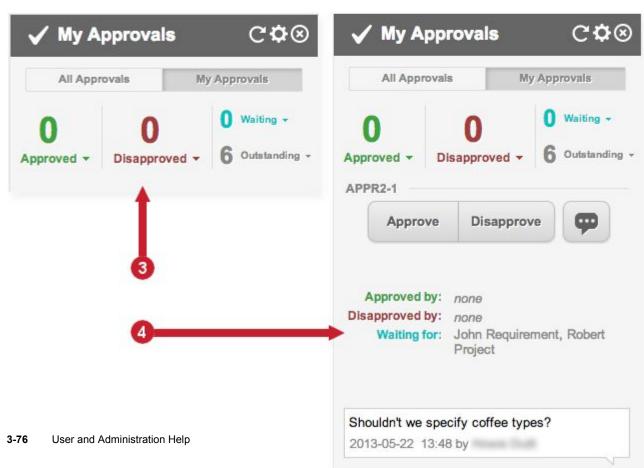


The Approvals panel has two tabbed pages:

- All Approvals: Reports statistics of the current state of all Work Items, regardless of who is designated as Approver(s). Also enables you to invite people to approve Work Items by providing a URL which you can paste into email or instant messages. When a Work Item is selected, this tab also provides a button for adding an Approval Comment, displays the names of users who are assigned to approve, have approved, and have disapproved the current item, and displays all Approval Comments for the Work Item, if any.
- My Approvals: Reports statistics of all Work Items for which you are designated as an Approver. When a Work Item is selected, this tab also provides buttons for approving or disapproving the current item, provides another button for adding an Approval Comment to the item, displays the

names of other users who are assigned as Approvers of the current item and what action each of those users has taken, and displays all Approval Comments for the Work Item, if any.





No selection
Work Item selected
No selection
Work Item selected

# Understanding the approval statistics

Approval statistics are color coded in the user interface (see previous figure).

- **Outstanding** All Approvals: No user is currently designated as Approver. My Approvals: You are not currently designated as Approver.
- **Waiting** All Approvals: Waiting for approval by at least one user and no user has disapproved it. My Approvals: You are designated as an Approver, and the reported number of items are waiting for you to review and approve them.
- **Approved** All Approvals: The item is approved by *all* the currently designated Approvers. My Approvals: You are designated as an Approver, and you have approved the item.
- **Disapproved** All Approvals: The item is disapproved by at least one of the currently designated Approvers. It is reported as Disapproved until such time as the disapproval is resolved, even if one or more Approvers have approved it.

My Approvals: You are designated as an Approver, and you have disapproved the item.

The Approvals panel statistics are for the current context such as Document or project.

Approvers are normally designated automatically by the workflow engine on the basis of some role ('project\_approver', for example), or an explicit list specified in the workflow configuration. The list of designated Approvers appears in the **Approvers** field of a Work Item when viewed in the Table view. It is possible to add Approvers to the list manually.

# **Using Approval Center features**

The topics in this section focus on tasks involved in the default approval process (as configured in Polarion's project templates), and explain how to access and use the relevant Approval Center features. Unless specifically noted otherwise, the information pertains to Work Items contained in a LiveDoc document and to Work Items contained in the Work Item Tracker (Navigation Work Items).

# **Review current Approvals**

1. Open the Approvals panel for the context you want to review (Document or project tracker), and select the All Approvals tab.

2. If reviewing in a Document, do not select any Work Item. If reviewing Document-based items in the Table or Tree view of the Tracker, select any Heading item.

The Approvals panel shows a statistics summarizing the overall status of approvals, reporting counts for items Outstanding, Waiting, Approved, and Disapproved. See Understand the Approval Statistics for more information.

If Work Items are not contained the Work Item Tracker, one Work Item is always selected, so you cannot see the Approval in its state when no item is selected. You can, however, still see the overall approval statistics in the Approvals panel.

# **Invite Approvers**

Approvers designated for a Work Item are listed in the **Approving Users** table of the Approvers field of Work Items. You can view the list in the Table or Tree view of Work Items. Ideally, Approvers should be automatically designated via the workflow. It is also possible to add Approvers by manually editing the Approvals field. Either way, when a user is designated as an Approver the system automatically sends him/her an email notification.

In an ideal world, all Approvers respond to the automated email notification and log on to review the Work Items they must approve. However, in reality, many people set up rules in their email clients that file or even ignore Polarion-generated notification emails, or they simply don't read notification emails. For this reason the Approval Center provides the Invite for Approval feature. When a set of Work Items in a project is ready for approval review, the project leader or other team member can use the feature to copy a URL that links to the items to be reviewed and approved. This URL can be pasted into emails or instant messages created by the team member inviting people to approve the waiting Work Items. These usually get more attention than system-generated communications.

- 1. Open a Document containing the Work Items that are ready for approval review, or open the Work Items topic of a project containing tracker-based Work Items that are ready for approval review.
- 2. Open the Approvals panel and select the All Approvals tab.
- 3. Click **Invite for Approval**.
- 4. In the Invite for Approval dialog box, copy the URL displayed there to your computer's clipboard.
- 5. Create email messages or instant messages for people who must review and approve the Work Items, pasting the URL from your clipboard into the message body along with any other text you write. For example, you might provide explicit instructions for your approval process in addition to the link.

# Find your items to Approve

If your job is to review and approve Work Items, the Approval Center feature provides an easy way to locate the items you need to review to approve or disapprove.

- 1. Log on to your Polarion portal and open the project that has items awaiting your review.
- 2. If the items are contained in a Document, open it by browsing the **Documents and Pages** topic in **Navigation**, or searching for the Document name.
  - If the items are created directly in the Work Item Tracker, click \*\*Work Items in Navigation to open the Table view of project Work Items. You may want to expand Work Items to show the

types that exist in your project, and click a Work Item type to filter the table for that type. For example, if you will be reviewing Requirements, expand Work Items and click on Requirements.

- 3. Open the Approvals panel, and select the My Approvals tab.
- 4. If there are any items awaiting your approval, the tab shows a number greater than zero in the Waiting state. Click the drop-down control for Waiting, and on the menu choose **Open Filtered Document** (if working in a Document), or **Open Filtered Table** (if working in the Tracker's Table view).

After following these steps, a new browser tab opens that displays the Document or the Table filtered to show only the Work Items awaiting your approval. These items list you as an Approving User in their **Approvals** field. The items are in the **Waiting** approval state, meaning you have not yet marked them as approved or disapproved. You can now proceed to review and mark items in this browser tab.

# Tip

The default configuration of your My Polarion page lists items awaiting your approval.

# **Query for Approval items**

In the Table view of Work Items, you can use the Query Builder to query for items with a specific approval state, and/or with specific users designated as Approvers.

You can query for a specific approval state selecting the **Approval State** field as a visual query element, and selecting the value(s) to query for. For example, to query for disapproved items, select Approval State in the fields panel, and then select Disapproved in the values panel.

You can query for items having one or more specific users designated as an Approving User. Create a visual query element and select **Approvals** in the fields panel, and then the name(s) of the approving user(s) in the fields panel. You can query for items where you are designated as an Approver by selecting **Me** in the fields panel.

For more information on using the Query Builder, see Search Work Items.

### Approve or Disapprove Work Items

Mark a Single Work Item:

- 1. Be sure you are working in the filtered Document or Table and have selected the **My Approvals** tab in the **Approvals** panel.
- 2. If working in a Document, select the Work Item you want to approve or disapprove. If working in the Tracker, select the item in the table.
- 3. In the **Approvals** panel, click **Approve** or **Disapprove**, depending on your decision.
- 4. Repeat the previous steps for other single Work Items in the Document or Table.
- 5. If working in a Document, save it to complete your review and log the approvals and disapprovals.

  If working in the Table view, click **Save Approvals** in the **Approvals** panel to complete your review and log the approvals and disapprovals.

# Note

- As soon as you mark any item using Approve or Disapprove, a link named Revert approval state change(s) appears beneath the buttons in the Approvals panel. You can click it to revert the approve and disapprove action on each single item you have marked, or you can select multiple items and use the link to revert the approve and disapprove actions on all of them at once.
- When you mark an item with **Disapprove**, a new Approval Comment is automatically created in the **Approvals** panel. It is recommended that you enter a comment before you save your changes.
- Your user profile must contain a role assignment that is granted the permission to approve Work Items. If you do not have this permission, you see a message when you try to approve or disapprove items. Contact your project leader or portal administrator to resolve permissions issues.

### Mark multiple Work Items:

In some cases, marking items approved/disapproved one at a time is too time-consuming. For example, suppose you review a specification document that has 50 Work Items in a Functional Requirements section, and you decide all of them can be approved. You can mark all or any portion of the requirement items in that section as approved in a single operation.

As with single Work Items, you should be looking at the items you need to review for approval in the filtered Document or Table.

To approve/disapprove multiple Work Items all at once:

- 1. Be sure you are working in the filtered Document or Table and have selected the **My Approvals** tab in the **Approvals** panel.
- 2. If working in a Document, select all the Work Items you want to approve or disapprove. Note that the selection must be contiguous; you cannot skip some item(s) in between.
  - If working in the Tracker, select all the items you want to approve or disapprove in the table using the check boxes on each row, as you would for Bulk Edit. The selected items need not be contiguous in the table. Alternatively, you can modify the query to display just the items you want to mark and select them all using the check box on the table header.
- 3. In the **Approvals** panel, click **Approve** or **Disapprove**, depending on your decision about all the selected items.
- 4. If working in a Document, save it to complete your review and log the approvals and disapprovals.
  - If working in the Table view, click Save Approvals in the Approvals panel to complete your review and log the approvals and disapprovals.

# **Note**

- As soon as you mark any item using Approve or Disapprove, a link named Revert
  approval state change(s) appears beneath the buttons in the Approvals panel. You
  can click it to revert the approve/disapprove actions on all the selected items.
- When you disapprove multiple selected items, a new Approval Comment is
  automatically created in the **Approvals** panel, and you must enter a comment before
  you can save your changes. The Approval Comment is applied to all the disapproved
  items.
- In a Document, if there is, for example, one item to be disapproved in between multiple items to be approved, you need to do multiple select-and-approve actions, selecting and approving all items before, and then all items after the disapproved item.
- Your user profile must contain a role assignment that is granted the permission to approve Work Items. If you do not have this permission, you see a message when you try to approve or disapprove items. Contact your project leader or portal administrator to resolve permissions issues.

# **Work with Approval Comments**

The Approval Center feature tags Work Item comments to identify them as Approval Comments. The following are useful points to know about Approval Comments:

- Approval Comments can be added to a Work Item at any time during the approval review process.
- Any user with the correct permissions can reply to an Approval Comment, and replies may have replies, creating a comment thread.
- A user with the correct permissions can mark Approval Comments as **Resolved**. Resolved comments are hidden, but can be shown to any user who wants to see them.
- An Approval Comment is automatically created when a single item or a group of items is marked as disapproved. While not required, it is generally good practice to fill it in.
- Approval Comments are regular Work Item comments, stored in the Comments field of Work Items, but are labeled as Approval Comments in the UI.

# **Add an Approval Comment**

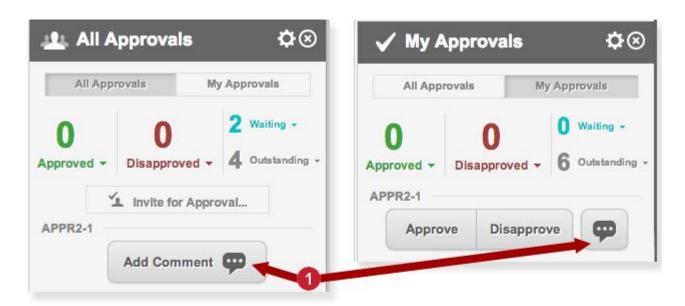
The Approvals panel displays a button for adding an Approval Comment when a Work Item or group of Work Items is selected. If multiple Work Items are selected, the comment is added to all of the selected items.

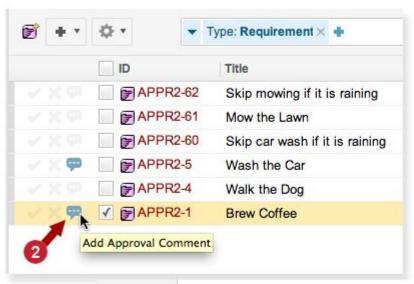
- 1. Select one or more Work Items in a Document or the Work Items Table.
- 2. In the Approvals panel, click **Add Comment**. A new empty comment appears in the panel.
- 3. Fill in the text of the comment.

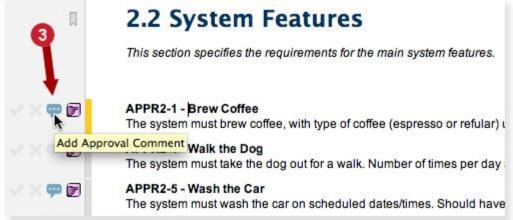
4. Save the Document, or if working in the Table view, save the Approvals.

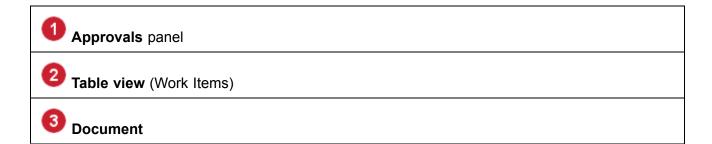
# Tip

You can also click P Add Approval Comment in the Document margin next to each Work Item, and on the row of each Work Item in the Table view.



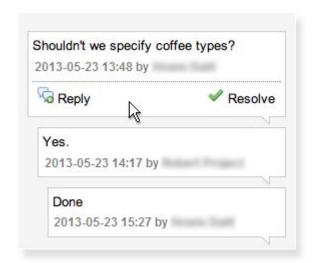






# **Reply to an Approval Comment**

- 1. In the Approvals panel, hover your pointer over the comment. Two icons and labels appear in the comment.
- 2. Click **Reply**. A new reply comment appears in the panel.
- 3. Fill in the text of the reply.
- 4. Save the Document, or if working in the Table view, save the Approvals.



Hover over Approval Comments to reply to them.

### Tip

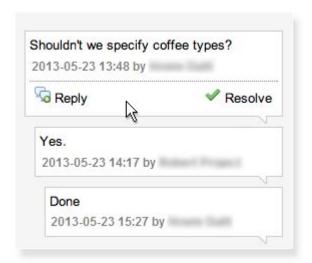
You can also reply to an existing Approval Comment using the Comments section of the Work Item form in the Work Items Table view.

### **Resolve Approval Comments**

When the approval discussion is complete, it can be useful to mark the Approval Comments as Resolved. This is especially true for items that were initially disapproved. These will always have one comment, and often have some back and forth discussion until the item is revised to resolve the issues that resulted in disapproval. Once the issues are addressed, marking the existing comments as Resolved hides them. The item can then go through the review and approval process again

without the distraction of the original Approval Comments. Any user who needs to see the original discussion can unhide the resolved Approval Comments.

- 1. Show the Approvals panel if it is not displayed.
- 2. Select the Work Item containing the Approval Comment.
- 3. Hover your pointer over the Approval Comment you want to resolve. If the comment is in a thread, select the top-level comment in the thread.
- 4. Click **Resolve**. The Approval Comment and all replies are shaded.
- 5. Save the Document, or if working in the Table view, save the Approvals. The Approval Comment and all replies are hidden.



Hover over an Approval Comment to resolve the entire thread.

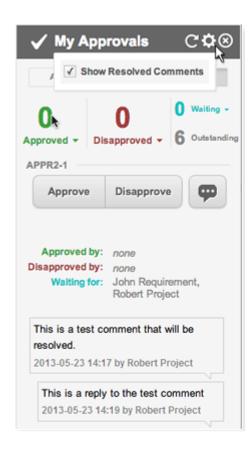
### View resolved comments

There are two ways you can show resolved Approval Comments, which are hidden by default when you access the Work Item they belong to:

 In the Approvals panel header, click Pane Settings and check the Show Resolved Comments box.

Resolved comments appear shaded in the Approvals panel when shown.

 In the Table view of Work Items, scroll to the Comments section of the form and check Show Resolved Comments.



How to reveal hidden resolved Approval Comments. (Resolved Approval Comments appear shaded.)

# **Reopen resolved Approval Comments**

It may be necessary to reopen some discussion thread around approval/disapproval of some Work Item. To reopen a resolved Approval Comment, you must first show resolved comments (see View resolved Comments). Once you are seeing resolved Approval Comments, there are two ways you can reopen one:

- In the Approvals panel, hover over the Approval Comment you want to reopen and click **Reopen**.
- Reply to an Approval Comment in the Approvals panel, or any reply in its comment thread.

# **Reset the Approval state**

Some Work Items are inevitably rejected (disapproved) during the initial phase of the approval process. The items may undergo discussion in Approval Comments and other means, and eventually the items are revised. After revision, they must undergo another review and be approved or rejected again. The situation can arise where you have revised items that were initially disapproved and are ready to be reviewed again, but are still marked as disapproved. For these items you can reset the approval from the Disapproved state to the Waiting state.

- 1. Select the Work Item in its containing Document or the Table view of Work Items.
- Open the **Approvals** panel and select the **All Approvals** tab.

- 3. Under the **Add Comment** button, you see the label **Disapproved by**: followed by the names of Approvers who disapproved the item. Click the **X** icon associated with this label and the names. (You see a tooltip **Reset approvals to the Waiting state**.)
- 4. Save the Document, or if working in the Table view, click **Save Approvals**.

# Tip

You can reset the approval state of several Work Items at once. In Step 1 above, simply select multiple Work Items.

If working in a Document, select contiguous items by dragging over them to highlight them. In the Table or Tree view, check the box on the row of all the items you want to reset at once. (They need not be contiguous in the listing.)

# Sign Work Items

This section details how signatures work when approving a Work Item or changing its workflow status.

See the Sign documents and Approval sections for additional information on how signatures and the approval process works and how to enable and configure them.

# **Work Item Approval Signatures**

When a Work Item is approved, the name of the user that approved it, the date, and the approval verdict appear in the comment section of the Work Item.

# Approve and Sign a Work Item

# Tip

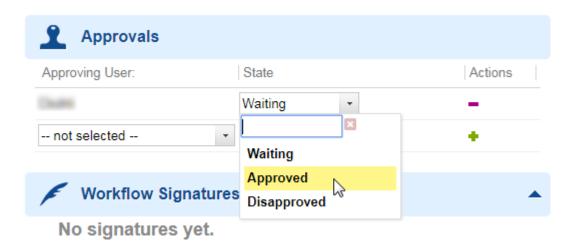
- Users with sufficient rights can Approve a Work item "On behalf of" the assigned approver.
  - (For example, if the assigned approver is on holidays but the item needs to be approved before they get back.)
- Likewise the designated approver can "Disapprove" any approval made on their behalf.
- Polarion supports up to 20 approvals per Work Item.

In the example below, a system administrator is approving the Work Item on behalf of another **Approving User**. When **Approving User** returns, they decide that the Work Item is not actually ready to be approved and reverses the decision.

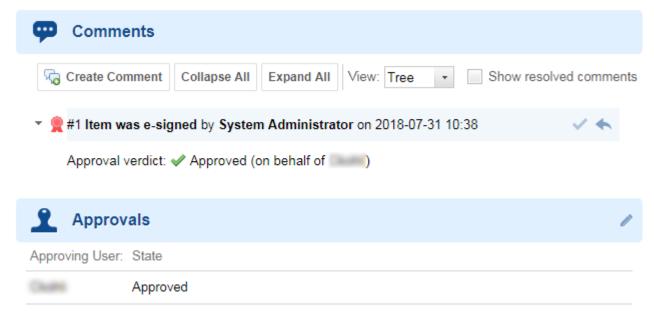
- 1. Open the target Work Item.
- 2. Scroll to the **Approvals** section and click in the approvals heading.
- 3. Select the **State** beside the target approver.

# **Note**

If you have sufficient rights, and want to add another approver, select the user in the drop-down list on the left and click 📩 on the right.



- 4. Change the approval **State** and click **Save**.
- 5. An approval comment containing the date, the user that approved it, and the approval verdict appears in the **Comments** section of the Work Item.

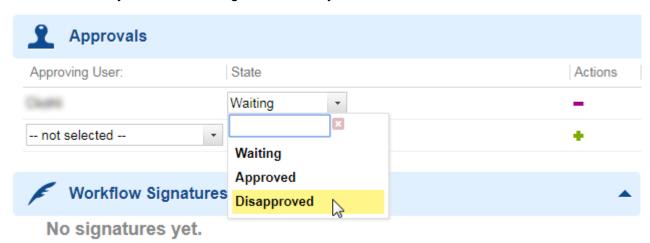


Now let's say that the **Approving User** returns from holidays and decides that the item approved on their behalf is not actually ready to be approved.

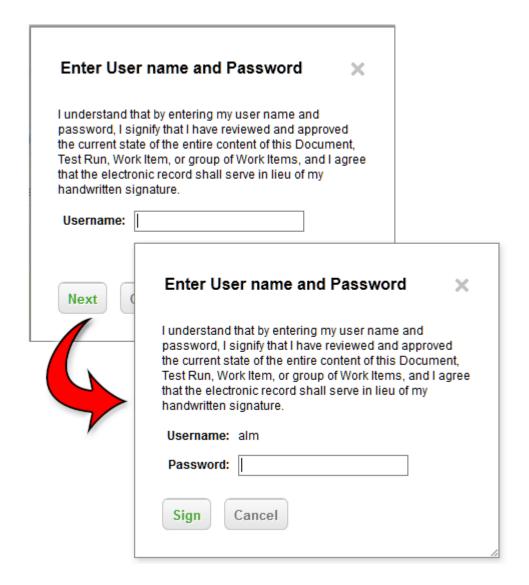
# **Change an Approval State**

1. Scroll to the **Approvals** section and click in the approvals heading.

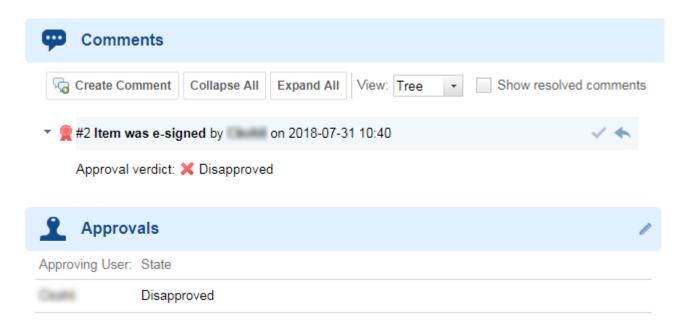
2. Select the **State** you'd like to change it to beside your username.



- 3. Click Save.
- 4. Enter your **Username** in the e-signature confirmation dialog and click **Next**.



- 5. Enter your password and click **Sign**.
- 6. The original approval comment will disappear and be replaced by the updated date, user, and approval state information.



# Work Item approval and workflow change traceability

Obsolete approval states are removed from the **Comments** section of the Work Item, but the entire approval process is still visible in the Work Item's **History**.

Field	Removed	Added
Author: Date: 2018-07-31 10:43 , Revision: 281		
Status	Reviewed	✓ Approved
Workflow Signatures		Status Change Done - Approved Signed: Declined: Invited:
Author: , Date	: 2018-07-31 10:41 , Revision: 280	
Status	<i></i> Draft	Reviewed
Approvals		System Administrator: Waiting Karel User: Waiting REVIEWER User: Waiting
Author: , Date: 2018-07-31 10:41 , Revision: 279		
Comments	#2 Item was e-signed by on 2018-07-31 10:40 Approval verdict: X Disapproved	#3 Item was e-signed by on 2018-07-31 10:41 Approval verdict:  Approved
Approvals	Disapproved	Approved
Author: , Date	: 2018-07-31 10:40 , Revision: 278	
Comments	#1 Item was e-signed by System Administrator on 2018-07-31 10:38 Approval verdict: Approved (on behalf of	#2 Item was e-signed by on 2018-07-31 10:40 Approval verdict: X Disapproved
Approvals	Approved	Disapproved
Author: System Administrator , Date: 2018-07-31 10:38 , Revision: 277		
Comments		#1 Item was e-signed by System Administrator on 2018-07-31 10:38 Approval verdict: Approved (on
		behalf of )
Approvals	: Waiting	Approved
Committed by: , Date: 2018-07-31 10:36 , Revision: 276 (created)		

# Sign a Work Item workflow status change

You can require users to electronically sign a Work Item when changing the workflow status.

The latest workflow change will appear in the **Workflow Signatures** section of the Work Item. Administrator's can configure the section so that if a workflow status is reversed, all signatures that **Reviewed** and **Approved** a **Reopened** Work Item will be cleared from this section. (All signatures and status changes are still tracked in the Work Item's History. See image above.)

### **Note**

 The Workflow Signatures section is configured by default on fresh, 18.2+ Polarion installations, but if you've updated from an older version, you need to add the following to the form layout configuration:

<field id="workflowSignatures"/>

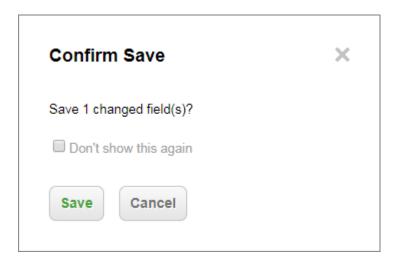
- See Require Signatures for Worklfow actions for details on how to make signatures mandatory when changing a Work Items workflow status.
- 1. Open the target Work Item.
- Select the Status you'd like to change the Work Item to in the drop-down list.



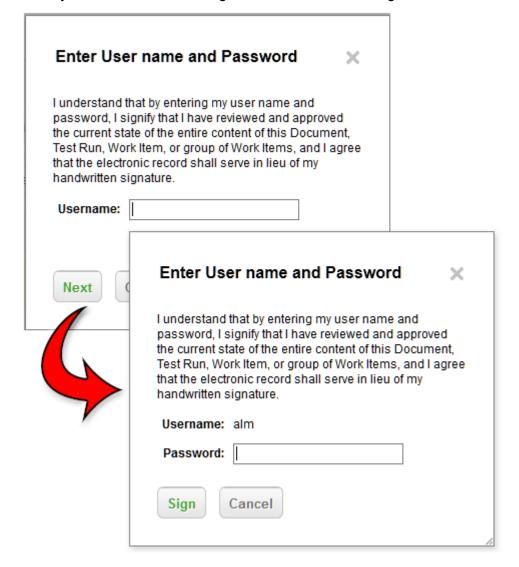
### **Note**

If you haven't approved the Work Item in the **Approvals** section, the **Perform** action **Approve** status will be greyed out.

- 3. Click 🖥 Save.
- 4. Click **Save** again on the **Confirm Save** dialog.



5. Enter your **Username** in the signature confirmation dialog and click **Next**.



6. Enter your password and click **Sign**.

7. The status will change and an entry will appear in the **Workflow Signatures** section of the Work Item.



#### Note

- Polarion supports up to 20 Workflow Signatures.
- The available workflow states will vary depending on your configuration.
- The number of signatures that appear in the **Workflow Signatures** section also depends on your workflow condition and function configuration.

# Assign a Work Item

You assign a Work Item to a person using the **Assignee** field in the Work Item detail. This field presents a drop-down list of Polarion users (filtered according to permissions). You assign the selected Work Item (or Items, if performing a **Bulk Edit**) by selecting the user's name from the list. Note that the assignee must have an account on your Polarion system, and must be assigned the **assignable** role in order to appear in the **Assignee** list. (It can be easy to omit setting up accounts, or assigning roles for e.g. a domain expert who is not on the engineering team but who needs to be assigned a Requirement, for example.) For more information on roles, see: **Configure User Roles**.

# Assign to multiple assignees

It is possible to assign the same Work Item to multiple people. For example, you might have a Work Item type "Meeting" which is scheduled, and for which you want to track time spent by all participants. You can accomplish this by assigning the item to multiple assignees.

The procedure is essentially the same as for assigning a single person. After you assign the first assignee, click • in the **Assignee** field, and select another user name from the drop-down list.

When there are multiple assignees, each additional assignee has a — icon beside the user name. If you need to remove an assignee, click this icon. The Assignee field cannot be left empty. To unassign the item completely, select the **not selected** value in the drop-down list of users in the **Assignee** field.

Any of the assignees who spend time on a Work Item with multiple assignees can create a Work Record to report time spent on it.

# Change a Work Item's status

You can initiate a change of status for a selected Work Item directly in the Work Item editor (Table and other views) without entering Edit mode. The Status field has a drop-down list that displays a choice of actions as configured in the project workflow.

Selecting an action value in the Status field enables the Save and Cancel buttons in the Work Item editor toolbar. Depending on the workflow configuration, you may be able to view and/or edit other fields such as **Resolution**.

The Status drop-down shows only transitions permitted by the workflow configuration, based on the item's current status. For example, the workflow configuration might require an item with status **Open** to transition to a status **Verified** before the **Close** action appears in the Status field's drop-down allowing the item to transition to the **Closed** status.

# Tip

Your project leader or administration should familiarize the team with the workflow configuration so that every understands the different Work Item statuses and the actions available in each status. In general, the workflow configuration maps the team's process and prevents steps from being skipped or bypassed.

(See also Configure Work Item Workflow)

### **Prioritize Work Items**

### **Prioritize Work Items overview**

The **Priority** field sets the priority of a Work Item for project planning purposes. Work Item priority is a float type numeric value. In products that have the LivePlan feature, the item's numeric value is used in conjunction with other factors by the LivePlan engine when calculating and rendering the project plan. In every case, it is possible to query for Work Items based on the **Priority** value, and to show the results of such gueries in Wiki pages.

There is also a set of priority "ranks". These are simply named ranges of priority values. The names of the ranks and the range of numeric priority values is specified in the global and/or project configuration, and are customizable by an administrator. For example, you might have ranks named High and Low with numeric value ranges of 51-100 and 1-50, respectively. (For a listing of the default rank names and range values, see Priority Ranks.) The Priority field displays the current Work Item's priority rank and numeric priority value.

The project workflow can be configured with an action that sets a default rank/value for new Work Items. This is useful when new items are created via import from Microsoft Office documents. For example, Priority of each new item might be set to **Medium**, and that rank might be configured to have a minimum value of 50.0.

# Set the Priority value

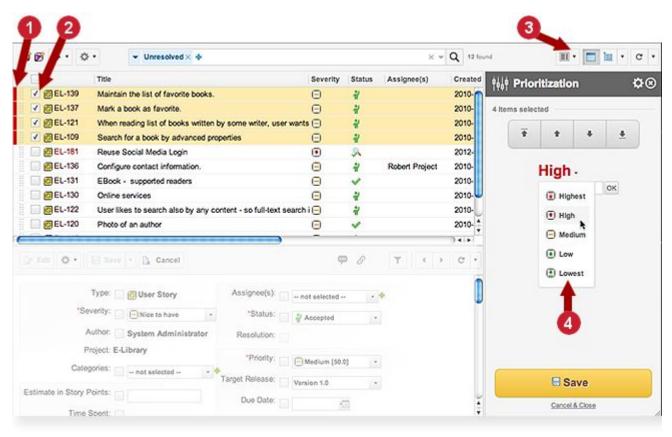
At some point, the project team and/or leader needs to prioritize their work, which means reviewing Work Items and setting their Priority value. Priority can be set for a single item, or for many items at once. For a single item, simply change the value of the **Priority** field and save the change. Note that you can select one of the configured ranks, or set a numeric priority value. The latter is useful

if you want to give some item a higher priority while keeping the same general rank. For example, if an item currently has priority set to "High", which is configured with a minimum numeric value of "70", you might assign the numeric value of "80". The item will then be given higher priority in the project plan by the LivePlan engine (and presumably by people processing their assigned items), but will still appear in results of queries for items with "High" priority.

For setting Priority for multiple items at once (in a Scrum planning meeting, for example), Polarion provides the Prioritization sidebar in the Table and Tree views of the Work Item Tracker.

(Navigation - Www.ltems).

#### **Table in Prioritization mode**



1 Drag item
2 Select item
3 Open sidebar
4 Work Item selected

When you invoke the sidebar, the table enters a special "prioritization mode" in which the items in the table can be dragged and dropped to prioritize them. You can drag a single selected item, or a group of selected items (table rows in prioritization mode display drag "handles"). The table reflects priority with highest priority at the top. By reordering the position of items in the table top-to-bottom, you reorder the relative priority of all items in the table.

### Tip

The set of items in the prioritization table is determined by the current query. Before invoking Prioritization, build and run a query to retrieve just the items you will be prioritizing.

Before reordering the items in the prioritization table it is advisable to run the Rescale All operation from the Pane Settings menu of the Prioritization sidebar (see Prioritization sidebar). This enables you to define the numeric range within which priority values will be assigned to the items in the table based on their position in the displayed set of Work Items.

### **Launch Prioritization Mode**

- 1. Open the Table view in the Work Item Tracker (Navigation → \*\*Work Items).
- 2. Open the Prioritization panel using the icon on the Table view toolbar (see the figure above).

#### **Note**

You cannot open the Prioritization panel from a Document. You must use the Work Item Tracker. If you want to prioritize all the items contained in some Document, you will need to build and run a query that retrieves all the items in the Document.

# **Techniques for setting Priority**

You can use one or more of several techniques to set priority value for items in the prioritization table:

- Assign a Rank
- Set a numeric value
- Reorder priority using buttons
- Reorder items visually
- Rescale selected items
- Rescale all items

The Prioritization Panel can help you understand the interface for each of the above techniques.

### Assign a Rank

You can assign one of the configured priority Ranks to items in the table using the Prioritization panel.

1. Select the items you want to rank by checking the box on the respective rows of the table in prioritization mode.

- 2. In the Prioritization panel, click the Rank selector and choose the desired rank from the drop-down menu.
- 3. Click the **Save** button to apply the rank to the selected items.

If all selected items had the same priority value before you assign rank, the Priority field of all selected items will be set to the minimum value of the chosen rank. For example, if all items had a value of 50 and you assign a rank whose minimum value is 70, all the selected items will be set to 70. If the selected items had different priority values before you assign rank, they will all be set to the same value.

#### Set a numeric value

You can set the same numeric value in the Priority field of multiple Work Items in a single operation. On save, the items will be repositioned in the table if the setting changes their priority relative to other items in the table.

- 1. Select the items you want to set by checking the box on the respective rows of the table in prioritization mode.
- 2. In the Prioritization panel, enter the desired numeric value in the **Priority** field and click **OK**. You can optionally select a different set of items and assign a number value to them in the same way.
- 3. Click the **Save** button to apply all settings to all changed Work Items.

# **Reorder the Priority with buttons**

The button set in the Prioritization table enables you to shift the position of one or more selected Work Items up or down one row in the table, or to the top or bottom position. (Keep in mind that in prioritization mode, top-to-bottom position in the table reflects highest-to-lowest priority.)

The button actions are as follows:

- **Up** Acts as if you have dragged the selected items between the item preceding the topmost selected item and the item preceding that one.
- **Down** Acts as if you have dragged the selected items between the item succeeding the bottommost selected item and the item following that one.
- **Top** and **Bottom** Use the configured Priority Increment so that all the selected items are at the top or bottom (depending on which button is pressed), and there are increment-sized gaps between them.

# Warning

No operation (Up, Down, Top or Bottom) touches any other item unless the target point is between items with the same priority. In that case, the priority values of items above the target point are rescaled.

To change an item's priority using buttons in the Prioritization panel:

- 1. Select the items for the operation in the table.
- 2. Click the button for the desired operation (Top, Up, Down, or Bottom).
- 3. Click the **Save** to apply the change.

# Reorder items visually

In **prioritization** mode, Work Items in the table are listed by priority from top (highest priority) to bottom (lowest priority), except when all items have the same priority value. It is often the case with a set of newly created Work Items or items imported from Microsoft Word or Excel that they have the same priority value.

You can re-prioritize the set of items by visually moving one or more selected items to a different position in the table using the drag-and-drop option.

It is important to understand that when you shift the position of some item(s), the relative priority of other items above or below the new position may be changed, and this will be reflected visually in the table (more items will become selected, and/or the rank colors of some items may change) and in the Range Selector of the Prioritization panel (the label may change).

If you shift the position of a selected group of items, the items in the group will keep the same priority relative to each other if they had different values to begin with. If they all had the same value, they will be rescaled within the scope of all items in the set being prioritized.

- 1. Select the items you want to reorder in the table.
- 2. Hover your pointer over the drag handle of any selected item, drag the selected item(s) to the desired position in the table, and drop the selection.

You can optionally repeat this operation, selecting other items and moving them in the same way.

3. Click **Save** to apply all changes.

#### Rescale selected items

You can optionally rescale the relative priority of several selected Work Items to each other so that the priority values are more evenly distributed. You can optionally limit the rescale operation to one of the configured priority Ranks ("High", for example).

- 1. Select the items you want to rescale in the table.
- 2. In the Prioritization panel, click and choose Rescale Selected. The Rescale Selected dialog appears.
- 3. Depending on the current priority of the items in the selection, the **Top** field or the **Bottom** field may appear. Top lets you specify the maximum priority value for items in the selection. Bottom lets you specify the minimum.
  - Review the field value, if present, and change if desired.
- 4. If you want all items in the selection to keep their current priority Rank, check the **Keep Rank** box. This will prevent, for example, items having a rank "Medium" from being rescaled to fall into a rank "High", or items having a rank "Low" from being rescaled into a rank "Lowest".

- 5. Click **Rescale Selected** to close the dialog.
- 6. In the Prioritization panel, click **Save** to apply the changes to all affected Work Items.

### Rescale all items

You can optionally rescale the priority of all items in the Work Items table (when it is in prioritization mode). The set of items rescaled is determined by the current query. Be sure that your current query retrieves all the items you want to rescale. For example, suppose you want to scale all the Work Items contained in a requirements specification document. You would want to query for all Requirement type items contained by the Document before invoking the Rescale All operation.

The Rescale All operation is designed to result in a reasonably even distribution of priority values among all Work Items in a given set. You can specify the range of numeric priority values to be used. Scaling will result in an even distribution of values within the specified range.

It is highly recommended to run Rescale All on sets of new Work Items before beginning to prioritize them manually using any of the techniques previously described. For example, when a specification document is imported from Microsoft Word, all the Work Items in the resulting Polarion LiveDoc will often have the same priority value. (Import rules that set different priorities based on content in the original document could be defined for the import operation, but this is not always done by the person importing a document.) Having an even distribution of values to begin with can facilitate the re-prioritizing process.

- 1. Be sure the current query has retrieved all the Work Items you want to rescale.
- 2. In the Prioritization panel, click and choose **Rescale All**. The **Rescale All** dialog appears.
- 3. Optionally change the default values in the **Top** and **Bottom** fields to specify the scaling range.
- 4. Click **Rescale All** to start the rescaling operation and close the dialog.
- 5. In the Prioritization panel, click **Save** to apply the changed priority values to all Work Items.

### Prioritize Work Items on a custom field

By default, the prioritization feature uses the standard **Priority** field of Work Items (field ID **priority**), writing a value to that field. However, it is not necessary to prioritize items on that field. You can prioritize items on any existing custom field, subject to the following limitations:

- The custom field must be of the same type as the default Priority field, i.e. float.
- The custom field must be configured for all the types of Work Items listed in the prioritization table.
  - For example, suppose the table contains both Requirements and Test Cases, and that Test Cases have been configured with a custom field "Test Priority", of float type. This field will not be available to select as the prioritization field because the Requirement type items do not have it. The table would have to contain only Test Cases in order to prioritize them on the custom field. This is easily done by build and run a query that retrieves only Test Cases.
- 1. Open the **Prioritization** pane as described earlier.
- 2. Run a query that retrieves only Work Items having the desired custom field.

- 3. Optionally select a subset of the items in the table.
- 4. In the **Prioritization** sidebar, click and select **Change Prioritization Field** on the menu.
- 5. In the Change Prioritization Field dialog, select the field on which to prioritize the items.

### **Note**

The dialog lists only those fields valid for all the items and that the standard Priority field is always present in the list.

See also Configure Custom Fields.

### **Work Item Comments**

#### **Work Item Comments overview**

Comments are an important tool for collaborating on Work Items. Comments can have multiple threads that discuss different topics related to the Work Item. When stakeholders reach consensus, individual comments and/or entire comment threads can be marked Resolved, which hides them from view unless individual users opt to view them. In order to add comments and replies, or resolve comments, you must be assigned a role that grants these permissions.

# Add a Comment to a Work Item

- 1. Select the Work Item in the Work Item browser pane.
- 2. In the Work Item editor, click **Comments** to quickly scroll to the Comments section of the Work Item form.
- 3. In the **Comments** section, click **Create Comment**.
- 4. Enter a title for the comment thread, type your comments in the multi-line text field, and click **Save**.

### **View comments**

The **Comments** section has several controls for adjusting the display of comments:

- Collapse All button: Collapses the comments hiding the comment text so that only the header of each comment is visible.
- **Expand All** button: Expands any collapsed comments so that the text of all comments is visible.
- View list: Controls the way comments are indented. The Flat option indents all comments at the same level. In this mode, comment threads are not readily apparent. The Tree option indents comments according to comment threads, making it easier to follow any particular thread. You can individually collapse the top level of any threads you don't want to read.

# Reply to a specific Comment

Every comment has its own **Reply** control ( ), which creates a new child comment. If a comment thread is collapsed, you see only the **Reply** control for the top-level comment, which may or may not be what you want to reply to. Expand any comment thread to see all comments and replies and to be able to access the **Reply** control for any comment in the thread.

# **Query Comments by author**

A frequent user need it to find comments written by a particular user. This is possible by querying for the user ID of a comment author. The basic syntax of such a query is:

```
comments.author.id:USER ID where USER ID is the ID of the user whose comments you want to find.
```

By itself, such a query will probably return many results, so you would most likely use it as an element in a more detailed query. For example:

```
type:defect AND updated:[20120527 TO 20120528] AND comments.author.id:jSmith
```

The above query looks for Defect type Work Items, updated "today" (i.e. within the 24-hour period defined by the 2 date values), and having comments by user "jSmith".

#### **Resolve Work Item Comments**

When stakeholders reach consensus through comments and comment threads, it can be useful to mark the comments and/or comment threads as Resolved. This hides them, after which individual users can opt to view them. It is also possible to reopen resolved comments if a stakeholder decides that further discussion is needed.

### Note

Work Item comments are distinct from Document comments. The latter are stored with Document data and can only be viewed in the Document Editor. Work Item comments are stored in the Comments field of Work Items. Approval Comments are Work Item comments, stored in the Comments field, but which pertain to, and constitute part of the audit trail for formal approvals.

See also Resolve Approval Comments, Resolve Document Comments and Threads.

### Tip

Replying to any comment in a resolved thread reopens the entire thread.

# To mark comments or threads as Resolved:

• In a Document, select the Work Item having the comment(s), open the **Comments** panel of the Document sidebar (**Show Sidebar** on the Document Editor toolbar), locate the comment to mark, hover your pointer over it, and click **Resolve**.

To mark an entire thread as resolved, hover over the first comment in the thread and mark it as above. (You cannot resolve individual child comments within a comment thread.)

In the Table or Tree view, select the Work Item having the comments. Then in the Comments section, locate the comment you wish to resolve and click \$\sigma\$ icon **Resolve this comment** thread on the comment header.

Mark the first comment in a thread as **resolved** to resolve the entire thread. (You cannot resolve individual child comments within a comment thread.)

### To view resolved comments:

- In a Document, select the Work item having the comment(s), open the Comments panel of the Document sidebar, click the [22] (Pane Settings) icon in the header, and choose **Show Resolved** Comments on the menu.
- In the Tracker, select the Work item having the comment(s), scroll to the Comments section of the detail form, and check the box labeled **Show resolved comments**.

The header of resolved comments is light grey as opposed to the header of unresolved comments which is light blue.

# To reopen previously resolve comments:

- In a Document, select the Work item having the comment(s), and open the Comments panel of the Document sidebar. Show resolved comments as described above. Locate the resolved comment you want to reopen, hover over it, and click **Reopen**.
- In the Tracker, select the Work item having the comment(s), scroll to the Comments section of the detail form, and check the option to show resolved comments. Locate the comment you want to reopen and click **Reopen this comment thread** on the comment header.

### Attach files to a Work Item

You can attach any type of file to a Work Item that will facilitate collaboration and/or implementation of the Work Item. For example, for a defect, you might attach a screen capture image showing the problem you encountered.

- 1. Select the item in some page Work Items view (Table, Road Map, and so on).
- 2. In the Work Item editor, click **Attachments** to jump to the Attachments section of the Work Item detail pane.
- 3. In the **Attachments** section, click **Attachments** section header.
- 4. Optionally enter a meaningful title in the **Title** column... one that will help you and others understand what the nature of the attachment.
- 5. Click Choose File in the File Name field. The file selection dialog for your operating system appears.
- 6. Select the file in the file selection dialog. Its path appears in the **File Name** column the field.
- 7. Click **Save** in the Work Item Editor toolbar to upload the file to the repository and attach it to the Work Item.

# **Note**

You can update an existing attachment with a new file version by clicking **Update Existing File** in the **Actions** column, clicking **Choose File**, and selecting the new file on your local file system.

Your system or project administrator can optionally limit the file size of attachments. Contact your administrator if you are unable to upload an attachment due to such restriction.

# **Voting**

If voting is enabled by the Polarion administrator, you can cast a vote for a Work Item. When many people vote for an item, project managers may decide to increase the priority.

When voting is enabled, the **Vote** item appears in the drop-down menu of the **Actions** in the Work Item Viewer/Editor toolbar. To vote, just click the menu item.

# Remove your vote

If you vote for a Work Item by mistake, or change your mind later, you can remove your vote. After you have voted for a Work Item, the **Vote** item no longer appears in the Actions menu. It is replaced by the **Remove Vote** item. Just click it to remove your vote.

# **View Work Item History**

You can see the complete change history of the currently selected Work Item using **Whistory** in the Work Item Viewer/Editor toolbar.

### **Set and remove Watches**

A Watch is a background process that monitors changes to a Work Item and notifies any users who have been specified as Watchers when a change occurs. The Author (creator) of a Work Item and the assignee are always notified of changes without the need to explicitly set a Watch. The author and assignee of linked Work Items are also notified of changes.

### Add or remove yourself as a Watcher

To add yourself as a Watcher:

- 1. Navigate to the Work Item you want to watch and select it in the Table view.
- 2. In the Work Item editor toolbar, click and select Watch in the drop-down menu.

To remove yourself as a Watcher:

- 1. Navigate to a Work Item you are currently watching and select it in the Work Items table.
- 2. In the Work Item editor toolbar, click and choose Stop Watching.

#### **View your Watches**

You can see a summary of all items to which you are assigned as a watcher. For server performance reasons, this information is only provided on demand.

- 1. Open your user profile (link on the My Polarion page or the tool view of Navigation).
- 2. In your user profile page, click the **Watches** button.

A listing of items to which you are assigned as Watcher is displayed in the Table view of the Work Items topic.

#### Add and remove Watchers

For any Work Item to which you have write access, you can add other users as Watchers of the item.

- 1. Navigate to the Work Item you want someone to watch and select it in the Table view.
- 2. In the Work Item's detail pane, scroll downward so you can see the **Watchers** section.
- 3. Click the on the right.
- 4. Select a user name in the **User** list and click the **Save**.

# **Set Planning constraints**

The **Planning constraint** field enables you to specify when work on the item should start or end. The date you specify is calculated in the Live Plan.

The following constraints can be set in the **Planning Constraints** list:

Start no earlier than: The Live Plan engine plans the item so that it cannot be started before the specified date.

**Start**: When selected, the Live Plan engine will plan the item so that it starts on the specified date.

End: When selected, the Live Plan engine will plan the item to finish on the specified date. The Start date will be calculated taking into consideration various factors including values set in Initial Estimate and Remaining Estimate fields, and planning data for linked items with dependencies.

This field, if used, usually has its value set during the project planning phase. For additional information, see Project Planning.

#### **Print Work Items**

You can print a listing of one or multiple selected Work Items to any printer to which your computer has access. You can select a single Work Item for printing in the Table, Live Plan, or Road Map views. You can also select multiple Work Items for print in the Table view. Alternatively, you can formulate a query to retrieve exactly the item(s) you want to print. You can print multiple items and their structure in the Multi Edit view.

- 1. Select the Work Item(s) you want to print in the appropriate Work Item view, or formulate a query to retrieve just the item(s) you want to print.
- 2. Locate the substant button in the toolbar, and click it to drop down a menu.

- 3. Choose Print in the menu.
- 4. Your system print dialog appears. Choose the target printer in this dialog, set desired print job properties if your print dialog provides this functionality, and launch the print job.

#### Print in Multi Edit view

# ALM REQ QA PRO

When the **Links** column is shown in the table of items, it is not printed at the position it occurs in the table, but rather last so that it can be higher to have more space to show the links.

# **Warning**

When printing the Multi Edit view in outline mode, the browser option print background must be enabled in order for the outline structure to show in the printed output.

# **Clipboard operations**

You can quickly copy the following Work Item data to your computer's Clipboard:

- Work Item ID (Copy ID)
- Work Item ID + Title (Copy ID + Title)
- Work Item URL (Copy Link)

When you hover your pointer over a Work Item title in the Work Item viewer/editor, an icon and a drop-down control appear to the right of the Work Item title. When you click the drop-down control, a menu displaying the above options appears.

#### Browser settings for clipboard access

Network and/or browser security configuration may disallow Clipboard access by applications such as the Polarion portal and prevent you from using this feature. You can change settings in different browsers to make clipboard operations smoother in Polarion.

#### Microsoft Internet Explorer

In the Scripting section, choose Internet Options - Security - Custom level - Allow programmatic clipboard access

- If set to Prompt, Internet Explorer will ask whether to allow clipboard access. This will occur
  only once per session.
- If set to Enable, clipboard access will always be allowed without any prompt.

#### **Mozilla Firefox**

By default, the security settings do not allow web applications and scripts to write data to the clipboard directly - the user must explicitly invoke copying to the clipboard. Therefore, when you invoke Cut or Copy actions when using Polarion in Firefox, Polarion presents a dialog containing your selected data and information on how to complete the operation and write the data to your clipboard.

## Tip

The AllowClipboardHelper extension was recommended for clipboard support in some Polarion versions. It it is no longer needed by Polarion and if you have installed it to Firefox, you can remove it if you do not need it for other applications.

# **Google Chrome**

The Chrome browser currently does not allow direct clipboard access by applications, and there is currently no option to change its configuration to allow it. The user must explicitly invoke copying to the clipboard. Therefore, when you invoke Cut or Copy actions when using Polarion in Chrome, Polarion presents a dialog containing your selected data and information on how to complete the operation and write the data to your clipboard.

Note that the same dialog may be presented for Internet Explorer and Firefox users if permissions settings block direct clipboard access by Polarion.

## **Link Work Items**

#### Linked Work Items and traceability

Linking Work Items is how you take advantage of the Traceability and Impact Analysis features. Even though these may not be available to everyone on your team (depending on licensing), you should plan and follow a process of linking Work Items from the outset, as it will be important for those users who do have access to traceability and impact analysis.

You want to create links from items like requirements to do the following:

- Test cases that verify them.
- Test tasks and/or change requests that implement them.
- Link from tasks and change requests to the repository revision containing the implementation source code.

By doing this, your team will have several key advantages with Polarion. You will be able to:

- Trace from requirements to the Work Items and source code that implement them
- Back-trace from some defective code (in a repository revision) to the tasks/change requests that implemented the defective code, and from those to the requirement that called for the functionality.
- Assess the impact (and by extension, the cost) of a changed requirement by seeing all the artifacts that would be affected by the change.
- Review and analyze various relationships between Work Items such as dependency, relevance, parent-child, duplication, and follow-up.

Some of the ways you can use linking for traceability:

- Link a Resource, like a section of source code, to a Work Item.
- Create links while editing an individual Work Item.

- Create links for multiple Work Items at once using the Bulk Edit feature.
- Create links using the Matrix View.

It is also possible to link Work Items while working in a Document. This is covered along with topics on working with Documents. See Link a Document's Work Items.

#### Resource Traceability through linked resources

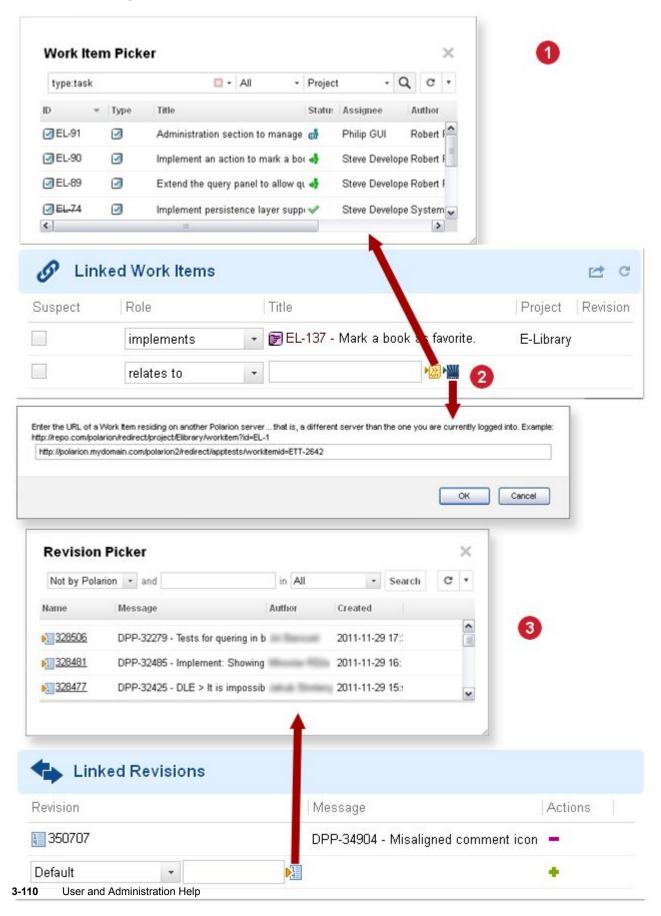
Once administrators have "Configure Resource Traceability" all developers have to do is add a comment in the source code they are creating and a reference will appear in the Work Item(s) they reference.

Search strings can also be defined that only search for Work Items with or without Linked Resources.

#### **Link individual Work Items**

In the Work Item Editor you can click the **Links** button to auto-scroll the page to the link sections of the Work Item Form: **Linked Revisions** and **Linked Work Items**. Use the small **Edit** button in the header of either of these sections to open the link editor in which you can select the repository revision or Work Item you want to link to the current Work Item.

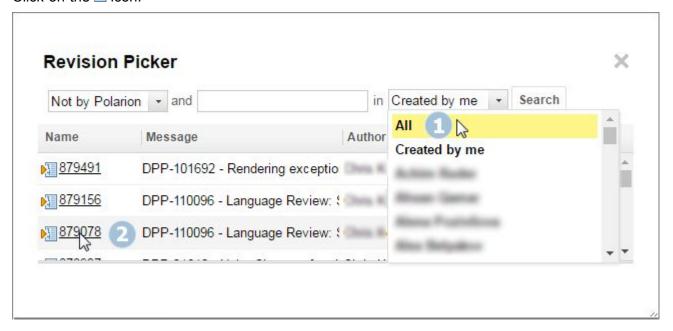
# Work Item linking in Edit mode



1 Link to a Work Item.
2 Link to item on a different server.
3 Link to a revision.

# Link a Work Item to a specific revision

- 1. Select a work item in Tree view or click on it in a document and click the **Open Item** menu option.
- 2. Scroll down to the **Linked Revisions** section and click on the *icon* on the right.
- Click on the licon.



- 4. Narrow or expand the scope of the listed revisions by selecting a user in the drop-down list. (It lists only the current user's revisions by default.)
- Click on the name of the target revision.
- 6. Click → to the right of the newly added revision.
- 7. Click **Save** at the top then **Save** in the confirmation dialog box.

The Work Item is now linked to the selected revision.

## Link Work Items across projects using Quick-link

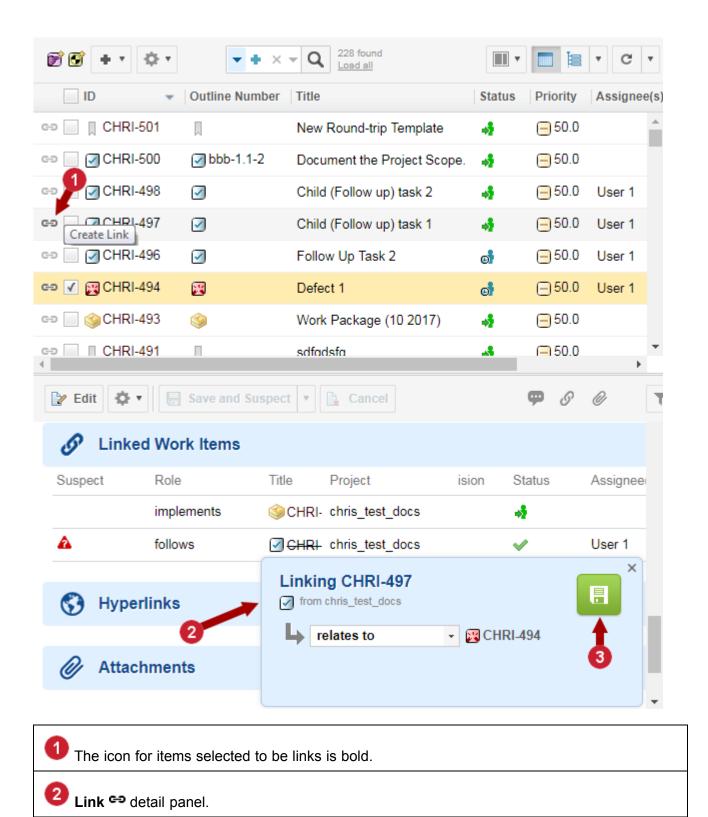
To create Work Items with links to Work Items from another Polarion project, add the **Project ID** from the other Polarion project to the front of each Work Item in their respective Excel cells. (ProjectId/WorkitemId).

#### **Use Quick-link**

You can quickly link a selected item in the table to one or more other items in the table using the Quick-link feature. You can invoke it from the Quick-link icon on the row of the Work Item you want to link. The target item or items must also appear in the table. You might need to modify the search guery to retrieve all the items to be linked.

- 1. Select an item in the Work Items table and click on the item's row. The **Link Detail** dialog box appears.
- 2. In the Link Detail dialog box, select the link role for the link you are creating. This describes the relationship of the source item to the target item(s) (Relates to or Depends on, for example).
- 3. If you want to link to multiple other Work Items, hold Ctrl (Windows) or Command (Mac and others).
- 4. In the table, locate the first Work Item you are linking to, and click the left of it. If you are linking to multiple items, scroll the table as necessary to locate each target item and click on it, continuing to hold down Ctrl or Command during the operation. The Link Detail dialog box updates with each link added.
- 5. You can save the links in two ways:
  - To save a single link just click in the Link Detail dialog box.
  - If linking to multiple items, click in the Link Detail dialog box then

(Links are not saved in the repository or the Work Item history until you save.)



Save and link selected items.

#### Tip

After initiating a link, if you then create a new Work Item, the new item will be linked to the first with the role selected in the link detail panel when you save the new Work Item.

If you make a mistake selecting a target item, you can click X in the link detail panel, confirm the cancel changes, and begin the linking process again.

You can open source Work Items in one browser tab/window and target Work Items in a different tab/window. Start the link with an item in the first, and finish by clicking link icons of one or more items in the second tab/window. The tabs/windows can contain either a Document or the Work Items Table.

The Quick Link icon for initiating a link is also available in the Work Item detail, to the left of the Work Item's title.



# Repository revision links

For information on linking Work Items to repository revisions, please see: Linking a Resolved Item to a Repository Revision.

## Tip

You can use Work Item Actions to create individual linked Work Items.

# Link multiple Work Items at once

The technique using the Bulk Edit feature is the same as for single Work Items. The same sections exist in the editor.

Bulk Edit specific features include:

- Bulk-edit only modifies selected Work Items.
- When you select multiple Work Items, you can create the same link(s) to all the selected Work Items at once.
- Direct links can be created because they are storsed in the selected items.
- Backlinks, using the Opposite Name, are stored in the backlinked Work Items. So you cannot create backlinks using Bulk Edit.

#### **Link Work Items in Documents**



Work Items may be defined and contained in a Document. Your can create links while working in a Document, and you can view and manage links in the Work Item Properties pane of the Document Sidebar. For information, see Link a Document's Work Items.

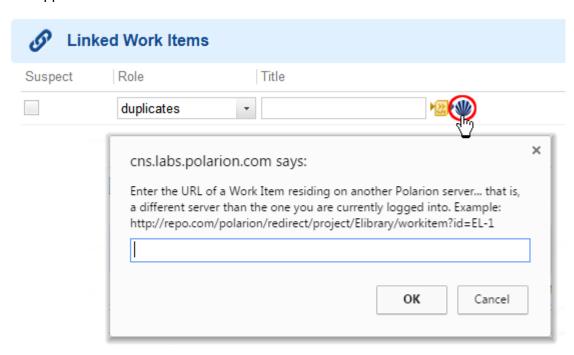
#### **Link remote Work Items**

# **ALM**

If your installation has multiple Polarion servers in a clustered environment, and if any of those servers hosts its own Polarion repository, you can link to any Work Item in the repository of one server to another Work Item hosted on a different Polarion server ("remote" Work Item). To do this, you must know the URL of the item you want to link to on the other server. If you don't already know it, you will need to log in to the other server, search for the target Work Item, select it, note its URL, then log back into the first server and proceed to link your Work Item to the one on the other instance. You can link items across any two remote servers.

Once you know the URL of the remote Work Item, the procedure to link to the remote item is straightforward:

- 1. Navigate to and select the Work Item from which you want to link to a remote Work Item.
- 2. Scroll to the **Linked Work Items** section of the form. The last line in this section is ready for you to specify a new link.
- 3. In the **Title** column, click the **(Select Work Item from Remote Server)** icon. A browser prompt box appears.



4. Enter the URL of the target Work Item on the remote host, and click **OK**.

## Tip

Remote Work Items appear in the same **Linked Work Items** table as locally hosted items, but they display the remote server icon in the **Project** column.

#### Add Linked Data items to Work Items

Polarion supports externally Linked Data that lets users link to objects from other tools, like Siemens PLM Software Teamcenter.

The Linked Data can be used to:

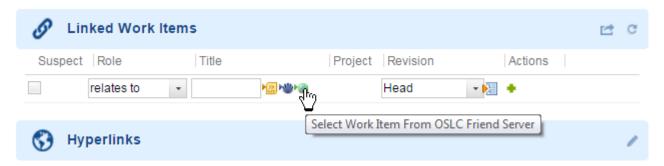
- Create links for or LiveDocs from objects that reside on external, Linked Data enabled tools. (Act as a Linked Data consumer.)
- Link Polarion Work Items to objects that reside on external, Linked Data enabled tools. (Act as Linked Data provider.)

(This is generally done by creating or selecting a remote object in a delegated UI provided by the Linked Data provider.)

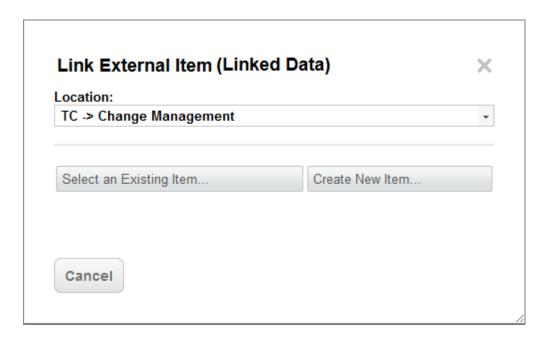
Once Linked Data Friends are configured by an administrator, they appear as an 🜆 icon in the 🗸 Linked Work Items section of a Work Item.

#### Add an external Linked Data item to a Work Item

- Select the target Froject.
- Click on Work Items in the navigation bar.
- Select an existing Work Item or create a new one.
- Scroll down to the **All Linked Work Items** section.
- Click on the blue **Linked Work Items** masthead or **to** edit the section.
- 6. If you have associated a Linked Data artifact container with the current project, you will see a 🍓 icon.



- 7. Click on the \inspec icon.
- The "Link External Item (Linked Data)" dialog appears.

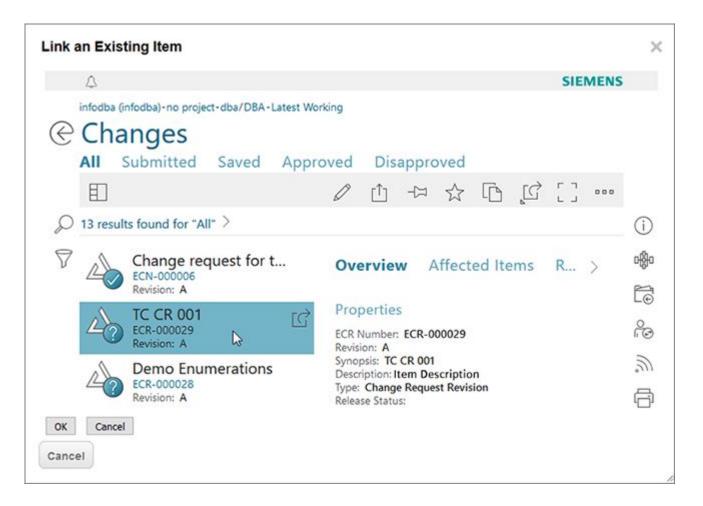


- 9. Select the External Linked Data location from the **Location** drop-down box.
- 10. Click **Select an Existing Item**. (Or **Create New Item** to create a new item within the external application.)

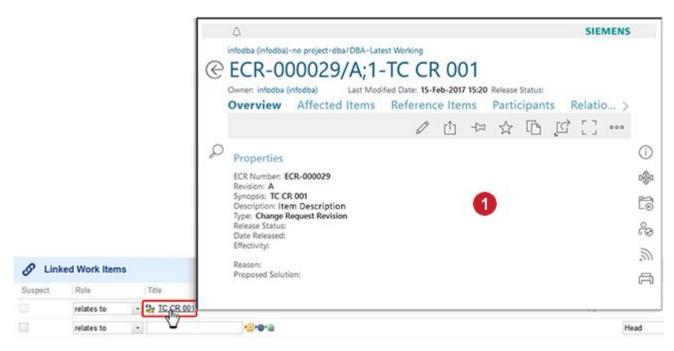
### **Note**

**Create New Item** will launch the external Linked Data application in a pop-up window. (The workflow to create a new Item will vary depending on the application.)

- 11. Example: Existing Item from Teamcenter:
  - a. Select Teamcenter from the **Location** drop-down box and click **Select an Existing Item....** (Teamcenter appears in a pop-up window.)



- Select an item from the list on the left. (An overview of its details appears on the right.)
- c. Click **OK**. The item is linked to the Polarion Work Item.
- d. Hover over the link to get a preview of the linked object.

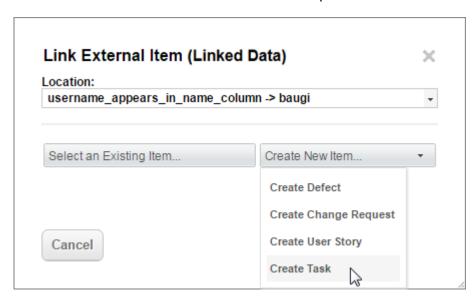


\* Mouseover preview

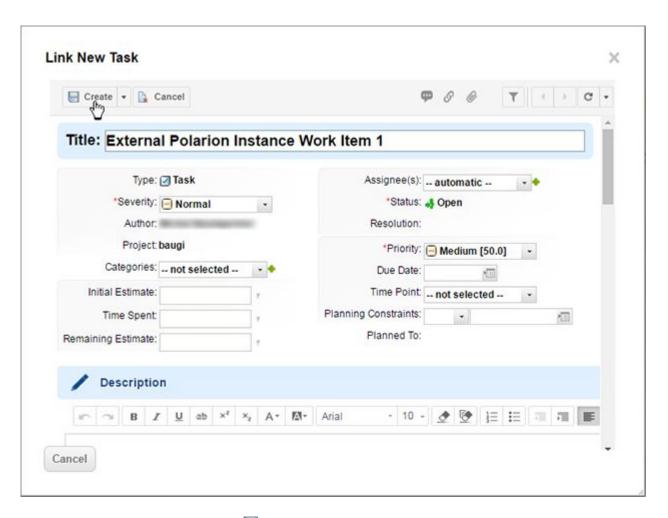
12.

a. A new Work Item from an External Polarion server:

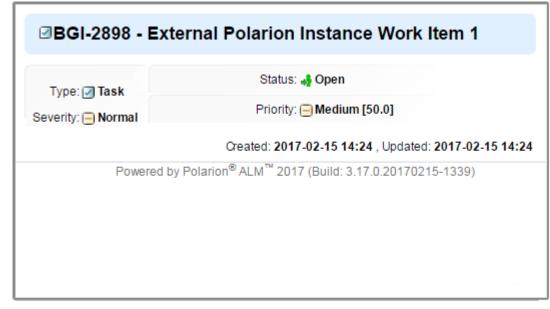
Select a Polarion Server from the Location drop-down box and click Create New Item...

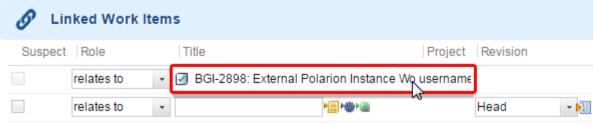


- b. Select an item type from the drop-down menu.
- The Link New Task dialog appears with the external Polarion embedded within the pop-up.



- Enter the required fields and click 🗏 Create. The Work Item is created and linked to the local Work Item.
- Hover over the link to get a preview of the linked Work Item.

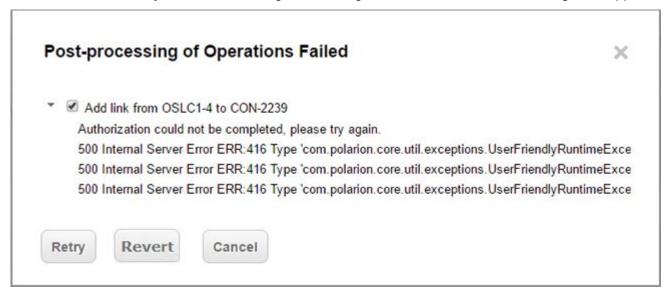




13. Click **Save**. (Links are created on both the local and external applications.)

# If Post-processing operations fail

If there is a connectivity issue while adding or removing **Linked Data** links, the following error appears.



Click Retry to resubmit the link addition or deletion.

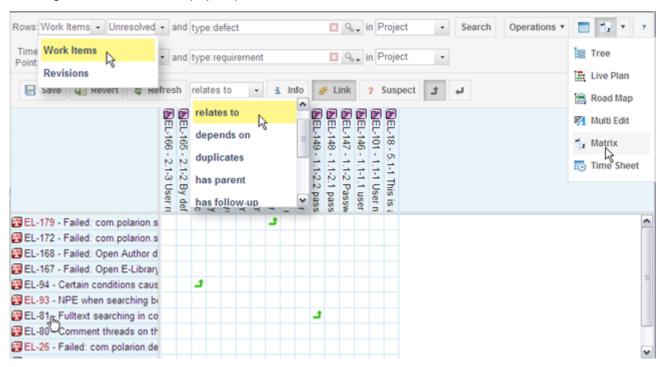
Click Revert cancel the link addition or deletion.

#### Link Work Items in the Matrix view

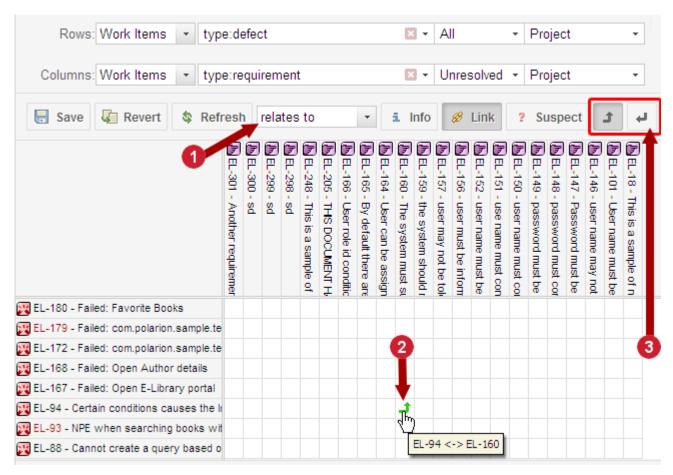
# ALM REQ QA

The Matrix view provides a robust way to both create and browse links. The matrix is populated with two subsets of Work Items - one for matrix rows, and one for columns. Each subset is built with a query in the special dual version of the Query Builder. Each query can access either Work Items or repository revisions, so the matrix can show links between Work Items only, or between Work Items and repository revisions. Work Item queries can also filter for link Role, **Depends on** or **Related to**.

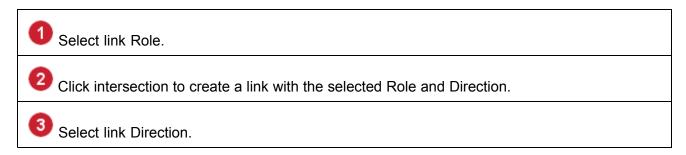
When the matrix has been populated, the intersection of each row and column shows if there is an existing link between the items in that row/column, and provides the possibility to create one if none currently exists. Existing links show an arrow in the intersection indicating the direction of the link. Clicking on the intersection pops up details about the link.



Refer to the screenshot below when following the procedure to create a link using the Matrix view:



Matrix toolbar in link creation mode.



#### To create a link in the Matrix view

1. Populate the rows and columns of the matrix by building queries that will display the source and target Work Items (or Work Items and repository revisions) in the matrix rows and columns.

# Tip

If you filter Work Items for one type in Navigation you may not get the query results you expect in the Matrix. You should generally select the top-level **Work Items** node in Navigation rather than one of the Work Item types listed in the sub-nodes.

2. On the toolbar, click Link Mode to switch link modes.

- 3. In the **Role** combo box, select a role for the link.
- 4. Select the directionality of the link using the **Direction** buttons.

#### Tip

You can do this step either before or after creating the link in the matrix (see next step).

- 5. Locate the intersection of the Work Items you want to link (or the Work Item and the repository revision you want to link, and click on that cell. A green arrow appears in the cell showing the directionality of the link. The actual link is not yet created.
- 6. Click **Save** to create the link. The green arrow in the cell changes to black.

#### Create links in Multi Edit view

# ALM REQ QA PRO

The Multi Edit view provides the possibility to create, browse, and edit flat or hierarchical groups of linked Work Items. When editing a Work Item in the Multi Edit view, the toolbar contains the (Insert Link) icon which you can use to link the edited Work Item to another Work Item, a repository revision, or a URL.

#### How links are handled in a Live Plan

# ALM PRO

There are two kinds of links recognized by the **Live Plan** feature: **depends** and **parent**. The planning.xml file in Administration assigns existing Work Item link roles to these two categories. Live Plan ignores links with roles that are not either **depends** or **parent**.

#### **Dependencies**

The dependent Work Item always links to the one it depends on. For example, if **A** depends on **B**, then **A** holds the link: **B \( A**. This kind of link ensures that **A** does not start before **B** ends. However, this does not influence the planning of **B**. Consider the situation when High priority task **A** depends on Low priority task **B**. Task **A** will be planned after task **B**.

#### **Parents**

Children always link to the parent. Therefore the **contains** link role should not be used; the **parent** role with the opposite direction should be used instead. The Live Plan always shows only the leaf nodes of the containment hierarchy. For example, if Requirement has child Tasks, then just the **Tasks** are shown, not the **Requirement**. The parent's planned start/end are calculated as the minimum/maximum of its children's planned starts/ends values. The Remaining/Initial Estimate value of the parent are not taken into account in this case.

## **Suspect links**

Work Items have a **Suspect** attribute that affects links. A configuration option Auto-Suspect, when enabled, sets the Suspect attribute to true on links between Work Items, indicating a potential inconsistency should any of the linked items be modified.

You can toggle the <code>suspect</code> attribute on and off using the **Suspect** check box available for each linked Work Item in the Linked Work Items section of the Work Item Editor. You can toggle the <code>Auto-Suspect</code> option using the Auto suspect control in the <code>Document Properties</code> panel of the <code>Document Sidebar</code> (for <code>Document-based Work Items</code>). The default state of the check-box can be configured in <code>Configuring Linking</code>.

#### **Note**

The **Suspect** attribute is only applied to links to Child Work Items. (Not to Parents or Structural links within a document.)



#### Structure document links are not Suspect

In the example below, a Requirement (DOCU-473) was added to the document, positioned as a child (using "**Tab**"), and linked to DOCU-461 using the is refined by option. The document was then Saved.

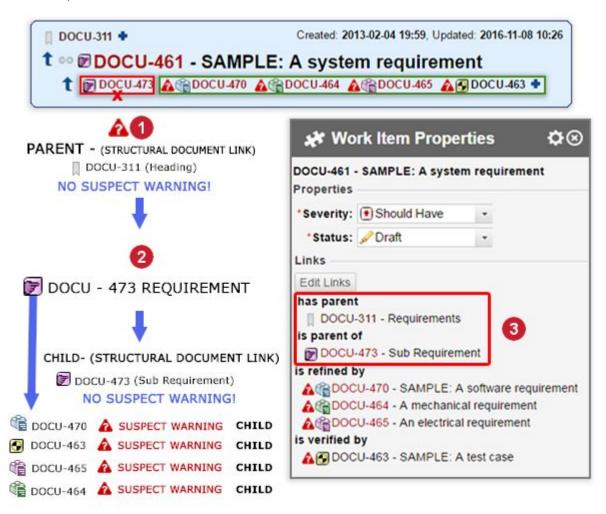
Structural links and their relationships are visible directly within the document and should be updated along with the parent.

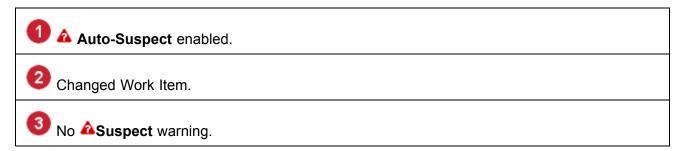
Below is a child structural Document that WILL NOT be subject to Suspect warnings:



The **Description** in the DOCU-461 Requirement Work Item was changed, and then saved using Save and Suspect. All the external child links for DOCU-461 are marked as ASuspect, but the newly created structural child link [MOCU-473] IS NOT.

How Suspects work:





# Work Item hyperlinks

The **Hyperlinks** section the Work Item editor provides a means to link a Work Item to any kind of online resource via a URL. For example, you might link to relevant Wiki pages or Documents, or to repository folders and files in your Polarion portal, or to pertinent information on the web or your

organization's intranet. When creating a hyperlink to any resource, you can specify whether it is an internal or external resource.

Hyperlinks are supplemental to the traceability links you create between Work Items. For traceability, you should always link Work Items to each other with an appropriate link role, and not use the Hyperlinks feature. Hyperlinks are disregarded by analysis features such as Traceability and Impact Analysis. If you want Work Items and Revisions to be available for analysis processes, do not use hyperlinks to link them as generic URLs.

# **Usage Examples**

Example 1: For a Work Item which is a Task requiring the delivery of some document and storing it in the Subversion repository (i.e. not keeping it just as an attachment to this Work Item, but let it live in some other location), you can create a hyperlink to the actual storage of the document once it is created. For example: http://myserver.company.com/repos/documents/text.doc.

Example 2: Link to any resource on the Internet. For example, link to Internet pages that are relevant to topics discussed in comments, so that everybody can easily refer to them. So a comment might say something like "this feature should be essentially like Their Product's Foo Bar functionality". Then a hyperlink can be added to linking to, for example, http://acompetitor.com/their product/features/foobar.html.

Example 3: Link to some resource in your Polarion repository and open it in the Repository Browser, for example, http://polarionhost.company.com/polarion/repo/someFolder leads to the Repository Browser's info view of the specified folder. Any hyperlink URL that appears to be a repository URL (starting with values specified in the system configuration property repo or repo.ui) is rendered as a repository URL and the link opens in the Repository Browser tool.

# Hyperlink roles

Hyperlink Roles are meant to help categorize hyperlinks, something like the Title for attachments. Currently Polarion does not rely on them in any way. Roles are completely configurable in the The intent behind the default configuration is that refers internally is for links to the Subversion repository, while refers externally is for linking to the Internet or systems outside Polarion/SVN. Hyperlink Role is really just a label.

Hyperlink Roles are meant to help categorize and label hyperlinks, something like the Title for attachments. Currently Polarion does not rely on them in any way. Roles are completely configurable in the **Administration** interface ( Work Items Enumerations hyperlink-roles-enum).

The default configuration contains two roles:

- Refers internally: For links to the Subversion repository.
- Refers externally: For linking to the internet or systems outside Polarion/SVN.

#### Hyperlink types

There are two types of hyperlinks:

- URL: Links to an online resource that is not in the Subversion repository.
- Repository URL: Links to a resource in the Subversion repository.

#### Create a Hyperlink

You can create a hyperlink in the Work Item Editor when the Work Item is in edit mode or by clicking the **Edit** button in the **Hyperlinks** section when the item is in View mode.

You must specify three things:

- The hyperlink role (see Hyperlink Roles ).
- The hyperlink URL type (see Hyperlink Types ).
- The actual URL of the hyperlink.

The user interface presents combo box controls for setting the role and the type.

When hyperlinking to an online/external resource, select **URL** in the **Type** field and enter the URL in the text box to the right of the combo box.

When hyperlinking to a Subversion URL, select **Repository URL** in the **Type** field. The **1** (Select Repository URL) icon appears at the right side of the URL text box. Click this icon to display a picker dialog in which you can browse the repository, and select the URL you want as the hyperlink target. The selected URL appears in the URL text box.

# **Export Work Items**

# **Export formats and procedure**

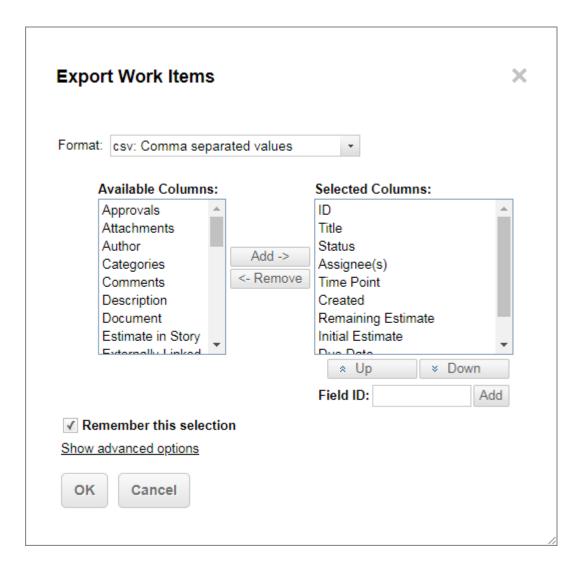
You can export a set of Work Items to various formats:

- Microsoft Word document
- Microsoft Excel
- Comma-separated Values (CSV)
- XML, with options for transforming to HTML, PDF, RTF, or plain text
- Microsoft Project

You can select a subset of all existing items for export using a search guery in the Table, Live Plan, or Road Map views. Supported Microsoft Office versions are listed in the installation guide documentation bundled with Polarion distributions.

#### Export Work Items to a document or other supported format

- 1. In the Table view of the Work Items topic or a Document, query to retrieve the set of Work Items you want to export.
  - (Depending on your license, it may be possible to do the above in other views. The Table view is present with every license.)
- 2. Click and choose Export to display the Export Work Items dialog.



3. In the **Format** field, select the target file format for the export output.

The dialog presents different export options depending on the content being exported and your selection in the **Format** field:

- **Template**: If shown, you may choose the template to be used to format the export output. Note that your choice of an export template may be overridden by a custom plugin implemented by your organization.
- Available/Selected Columns: If shown, you can select from a list of available data fields (columns in tabular formats), and add them to a list of selected fields/columns which will then have values from the selected fields included in the export output.
- csv: Comma separated values: If this format is selected, you're given an option to Remember this selection. When selected, Polarion remembers the fields and columns you chose for future csv: Comma separated values exports.
- 4. If the **Advanced options** link appears for the selected format, it leads to additional format-specific export options such as the Charset. Check the settings to be sure no changes are needed.

5. Click **OK** to begin exporting. You will be prompted to specify a location for storing the exported file locally.

#### Note

#### **Export template override:**

Organizations can optionally implement restrictions via the plugin API to influence the template used for export to Microsoft Word and Excel, with restrictions on individual exports by individual users, to include a watermark in the exported documents and/or to conform to other required policies.

If such a plugin is in use, the Export Work Items dialog shows a message under the Template field when Word or Excel is selected as the target format: "The selected template may be substituted with one that complies with company policy."

## **Linked Work Items in exports**

When you export Work Items to CSV or Excel Round-trip format, Work Items linked to and from the exported items can optionally be included in the exported document. Items linked via derived links, items cited in Subversion commit messages for example, can also be included in the export to CSV or Excel Round-trip.

You include linked Work Items in the export by adding one or more of the following from the **Available Columns** to the **Selected Columns** list in the **Export Work Items** dialog:

- Externally Linked Work Items
- Linked Work Items
- Linked Work Items Derived

#### Repository Revision in exports

When you export Work Items to CSV or Excel Round-trip format, you can optionally include revision numbers of any linked repository revisions in the output. Revisions linked to linked Work Items will also be included. Revision numbers in the exported document are formatted as hyperlinks which lead to the revision in your Polarion portal, displayed in the Repository Browser.

You include linked revisions in the export by adding Linked Revisions, from the Available Columns, to the **Selected Columns** list in the **Export Work Items** dialog.

#### **Export the hierarchical table**

# ALM REQ QA

If the Tree presentation option is selected in the Table view, then when exporting to MS Word the tree structure appears in the exported Word document. Hierarchy is indicated in the document using heading-level paragraph styles **HeadingN** for paragraphs containing the titles of the Work Items. After export, you can generate a Table of Contents in Microsoft Word, and it will display the same hierarchy as the table in the Tree display mode.

Before exporting items when viewing them using the **Tree** presentation option, you should specify the sorting of the Work Items, including child levels. Remember that you can customize the fields that appear in the table using the Customize Table option on the Refresh menu of the Work Items Table toolbar. If the existing fields are not sufficient for the sort order you want, you can add one or more Custom Fields and include them in the sort.

The Tree presentation of Work Items in the Table view may be exported only to these formats:

csv: Comma separated values

doc: Microsoft Word template

xlsx: Microsoft Excel

# **Export the Matrix view**

# ALM REQ QA

There are two report options for exporting data from the Matrix view:

- MS Excel Table
- MS Excel Traceability Report

#### **Excel Table**

You can export data from the Matrix view as a table to Microsoft Excel (normal document, not Excel Round-trip). The export format is XML for MS Excel 2003. Exported are rows and columns visible in the page, and only links with roles visible in the page are exported. A maximum of 256 columns (255 column Work Items or Revisions) can be exported due to limitations of Excel. If the number of columns found is greater, you will see a message and you should formulate a query to return a smaller column count.

When the Matrix view is current, the **Export** item appears on the **Operations** menu. Once you have used the search controls to obtain the data you want to export, use this menu command to launch the Export process. In the Export dialog, **choose xls: Excel Table** in the **Format** field. You have the following additional options:

- Export link roles Choose to display or not display the link role in the Excel cells. Yes makes sense only if links of all roles are displayed. When links of all roles are exported, but role names are not exported, then only one link from all links between two Work Items in the same direction is displayed.
- Export Work Item titles Check to export Work Item titles. If unchecked only Work Item IDs are exported.
- Export Revision messages (Revision matrix only) Check if both revision names and messages are exported. If unchecked, only revision names are exported.

The following notes apply to the exported data in Excel:

• The exported document contains the export date, the queries used to obtain row and column Work Items or Revisions, and in the cases when one particular link role is selected, the role of the link.

- Headers of rows and columns are hyperlinks to the Work Item or revision details in the Polarion portal.
- Links are indicated by an arrow like character (possibly with a role name next to it).
- Suspect links are marked by the ? character next to the link-indicator character.
- Direction of a link between a Work Item and a Revision is always from the Revision to the Work Item, regardless of whether the link is created via a revision message or directly in a Work Item.

# **Excel traceability report**

You can export traceability data from the Matrix view to Microsoft Excel. The export format is XML for Microsoft Excel 2003. The traceability report has two columns: the first represents rows of the matrix, the second contains list of all linked Work Items or Revisions that are present in a column of the matrix. Only links with the role selected in the control are exported.

Export is possible even when the portal does not display the matrix due to a very large number of items retrieved by the search query. A message in the portal informs you if this is the case, and the Link Role combo box is filled with link roles defined for current project or for the global level.

When the Matrix view is current, the **Export** item appears on the **Operations** menu. Once you have used the search controls to obtain the data you want to export, use this menu command to launch the Export process. In the Export dialog, choose the xls: Excel traceability report in the Format field. The following additional options are presented:

- Export link roles (Work Item matrix only): If checked, the link role is displayed in the report cells. Checking this option makes sense only if links of all roles are displayed. When links of all roles are exported, but role names are not exported, then only one link from all links between two Work Items in the same direction is displayed.
- Export Work Item titles If checked, Work Item titles are exported. Otherwise, only Work Item IDs are exported.
- Export Revision messages (Revision matrix only) If checked, revision commit messages are exported, otherwise only the revision names are exported. If the matrix does not retrieve revisions, this option is ignored.)
- Include direct links (Work Item matrix only) If checked, items linked by direct link are exported when the search parameters retrieve only Work Items. The option has no effect if the matrix retrieves revisions.
- Include backlinks (Work Item matrix only) If checked, backlinked items will also be exported when the search parameters retrieve only Work Items. The option has no effect if the matrix retrieves revisions.

### **Export to XML and transformed formats**

# ALM REQ

You can export Work Items from your projects to an XML structure used as input for transformation to various file formats by means of XSL templates stored in the Polarion system. The procedure is the same as export to other formats:

- Run a query to retrieve a set of Work Items for export.
- Invoke the **Export Work Items** dialog box from the menu in any Work Item where **Export** is present on that menu.

In the Export Work Items dialog's **Format** field, select any option that begins with or contains xml or xml. If multiple templates are present, select the desired template in the **Template**field and click **OK**. Either save the exported XML document to your local file system, or opt to open it in whatever application is registered for the file format created by the selected template.

The Export Work Items dialog provides the following XML export options in the Format field. Some both export the selected Work Items to XML, and also invoke XSL transformation processing to transform the output to different file formats. The options are as follows:

- html: XML Export: Content is exported to XML and transformed via XSL processor to HTML.
- pdf: XML-FO Export: Content is exported to XML and transformed via XSL and FO processors to PDF.
- rtf: XML-FO Export: Content is exported to XML and transformed via XSL and FO processor to RTF.
- txt: XML Export: Content is exported to XML and transformed via XSL processor to plain ASCII text.
- xml: XML Export: export to XML transformed via XSL processor to a different (or the same)
   XML structure.

The above options are present when the Work Items table is in the Flat (table) presentation mode. If the Work Items table is in Tree (hierarchical) presentation mode, a subset of these options is presented with only those options that are supported for the Tree mode.

#### Note

**Custom XSL Templates:** You can develop custom XSL templates that transform exported Work Items according to your own requirements. For information, see: Customize XML Export, and XML Export Schema.

PDF and Non-English Characters: In order for PDF to be exported properly when Work Items contain non-English characters, the \*-fo.xsl template that is used for the export (.polarion/tracker/export\_templates) must explicitly name the font-family for all text in the resulting FO file. This is most easily done by setting the font-family attribute of the fo:root element. For example: font-family="Arial" (used in the default Sample-fo.xsl).

If this font-family specification is not provided, FOP replaces non-English characters with # in the resulting PDF output.

**Export for Excel Round-trip** 

ALM REQ QA

The export option xlsx: Microsoft Excel exports Work Items in a round-trip format for Microsoft Excel, enabling an external user to edit Work Item content in Excel. A Polarion user can then import the edited Excel sheet back to Polarion, updating the Work Item content from the externally edited Excel document and/or creating new Work Items defined externally in the exported Excel document. If Work Items are exported from a Document, it is possible to allow changing of the Document structure externally in the exported Excel worksheet.

For information, see Using Excel Round-trip.

# **Export to Microsoft Project**

# **ALM**

Polarion also supports export of Work Items to Microsoft Project enabling sharing of the current Polarion project plan with users of Microsoft Project.

For information, please see Export Work Items to Microsoft Project.

#### Work Item actions

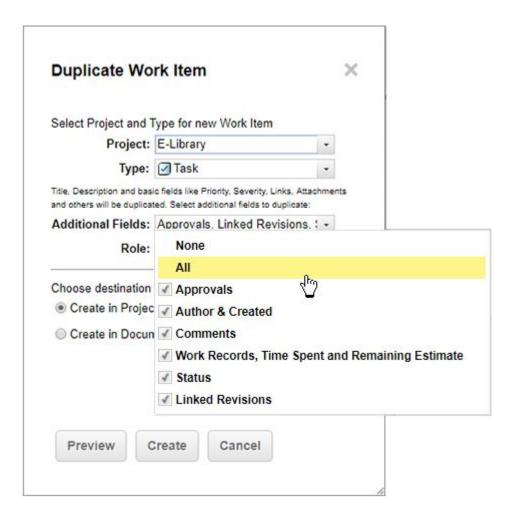
# **Duplicate Work Items**

The menu in the toolbar of the Work Item Viewer/Editor provides access to several useful actions you can perform on Work Items.

You can duplicate a Work Item in several ways:

1. Duplicate as tracker-based Work Items in the same or a different project.

This will copy selected Work Item fields. (Selected using the Additional Fields option in the **Duplicate Work Item** dialog box.)



2. Duplicate a Work Item within a document lets you quickly copy a saved Work Item and selected data within the same document or to an entirely different document - even one that resides in another project. (Provided you have the CREATE NEW Work Item and MODIFY CONTENT permissions for the target document and project.) You can create either a new Work Item based on the clipboard data or a Referenced Work Item.

#### Fields are duplicated:

- When duplicating a Work Item within the same document.
  - Only the fields that are visible in the document.
  - Applies to both saved and unsaved Work Items.
- When duplicating a Work Item from one document to another.

You can choose one of the following options from the Duplicate drop-down list.

- Basic will copy the Title, Description and Test Steps values.
- Advanced will copy the Basic option fields plus the Priority, Severity, Links, Attachments and Custom Fields.
- 3. Copy and Paste an unsaved Work Item within the same document.

(This will only copy its highlighted content. No additional data will be copied.)

# Things to consider while duplicating Work Items

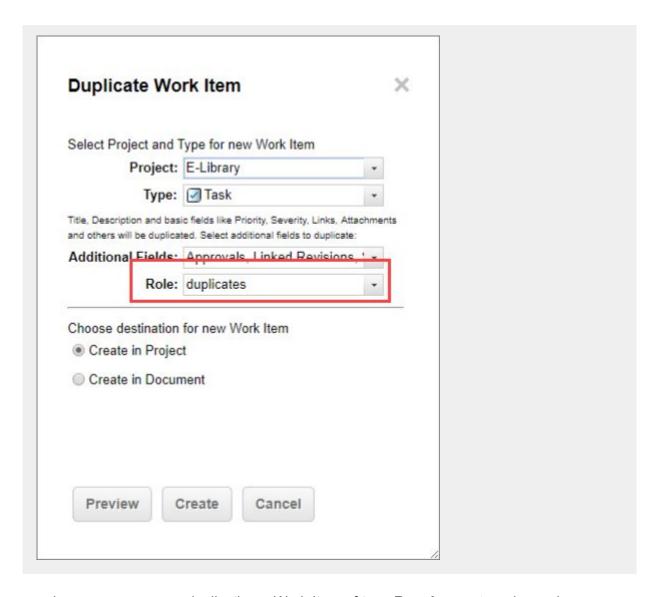
- You can duplicate a single or multiple Work Items. (If your license enables the Bulk Edit feature.) If multiple selected Work Items are all the same type, they can be duplicated as a different type. For example, the type in the source project might be **Requirement**, and a corresponding type in the target project might be **User Story**.
- When duplicating a single Work Item, there is an option to **Preview**. A preview appears in the editor and is not created until Create is clicked.
  - (There is no **Preview** when multiple Work Items are selected.)
- By default, duplicated items have a link to the original item with a "Relates to" link role. Existing links between source Work Item(s) are duplicated as links between the same duplicated items, including structure links if items belong to a Document.
- Votes and Watches on the source Work Items are NOT duplicated in the target Work Items. (But they do appear in the preview.)
- If you duplicate items into a different project, be sure you have CREATE NEW Work Item and MODIFY CONTENT permissions for the target project.

#### Link roles for duplicated Work Items

When duplicating a single Work Item, you can select the role of the link that will lead from the duplicate to the original. You can select only the link role from those allowed by the Work Items configuration between the type of the duplicate Work Item and that of the original. The Role list in the Duplicate Work Item dialog box provides one additional option ind addition to the roles: - not selected -. Choosing this item in the list results in no link between the duplicate and the original items.

#### Tip

When the Duplicate Work Items dialog box opens, the preselected link role in the **Role** list is the one that is set as default in the Work Item Link Role enumeration in the project configuration, if any is specified.



For example, suppose you are duplicating a Work Item of type **Requirement**, and you choose **User Story** as the type of the duplicate. The resulting item will be of the **User Story** type. The available link roles in the **Roles** list will then be the ones that the link roles configuration allows to exist between a User Story and a Requirement - **relates to**, **depends on**, **implements**, or **– not selected –**, for example. Assuming the default link role for this project is **depends on**, it will be the preselected role in the list.

#### Work Item bulk copy in Table view

The maximum number of Work Items that may be duplicated at once is 100. You can select more, but you will need to process them in chunks of 100 or less. See Bulk Editing a Large Number of Work Items for more information.

If more than the maximum is selected, the Duplicate action appears in the Viewer/Editor section rather than the button described above.

## Duplicate a Work Item from the Work Item list in Table View

You can duplicate existing Work Item(s) in the same or a different project. You can duplicate them as a regular Work Item, or associate them with a Document.

- 1. Select the item to duplicate in the Work Item Table view list.
- 2. In the Work Item Viewer/Editor toolbar, click when by Duplicate in the drop-down menu. (The **Duplicate Work Item** dialog appears.)
- 3. In the **Project** list, select the target project for the duplicate Work Item. The current project is selected by default.
  - In the **Type** list, choose the type for the duplicate Work Item. It need not be the same type as the item you are duplicating.
- 4. If you want to create the duplicate Work Item as a regular item in the target project, select **Create** in Project. If you want to create the duplicate Work Item in a Document in the target project, then select Create in Document option, and then select the Space and Document names in the **Space** and **Document** lists, respectively.
- 5. (Optional) Click **Preview** to preview the Work Item in the bottom section of the screen then Create .
- 6. Click **Create** to create the Work Item(s) without a preview.

### Note

The value of the **Status** field is copied to the new item only if no initial workflow transition is defined for the item type. If an initial transition is defined in the workflow configuration, it executes upon creation of the duplicate item, and the new item enters the initial workflow state (Draft, Open, or whatever, per the workflow configuration).

If the source Work Item is one that has been approved using an electronic signature, any new Work Items derived from it through reuse will not show any indication of the source item's signature, and the new items are *not* approved or electronically signed. The new item(s) must also be reviewed, approved, and e-signed.

#### **Duplicate a Work Item within a Document**

Users with the CREATE NEW Work Item and MODIFY CONTENT permissions for the target document and project can easily copy a Work Item within a document. They can even change the Work Item's type. Multiple Work Item Types can be defined for the document, but should be done before duplicating Work Items within a document.

- 1. Highlight the text of a Work Item(s) to copy or click the whitespace so that a yellow bar appears beside its icon.
- 2. Press Ctrl+C, right-click on the browser and click Copy. Alternatively, click on the Work Item's icon, click Copy and then click Item.



(Images and attachment previews within the item's rich text fields, for example, **Description** will also be copied. They will appear once the Work Item is saved.)

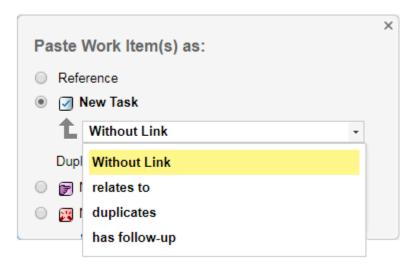
- 3. Open a document in the project you would like to copy the Work Item to.
- 4. Click on where you would like the copy created.
- 5. Press Ctrl+V or right click and click Paste.

The **Paste Work Item(s) as:** dialog box appears and the Work Item Properties sidebar opens.



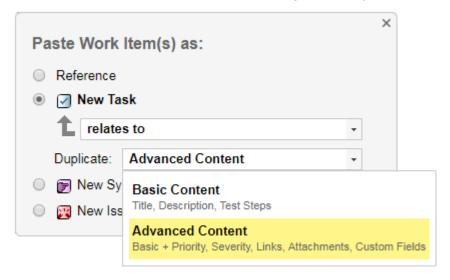
- A Referenced Work Item, not a copy.
- 6. Select a **New** Work Item. (Reference will only create a link to the original Work Item.)

  (If a Work Item type is greyed out, you lack the permissions needed to create it.)
- 7. Select whether the new Work Item should be linked to the original.



(Link roles will vary depending on the Work Item type selected and the link role configuration. See Configure Work Item Linking for details.)

8. Select what values to duplicate from the original. (They can be edited again later.)



- The **Basic** option will copy the **Title**, **Description** and **Test Steps**.
- The Advanced option includes the Basic option plus the Priority, Severity, Links, Attachments, Custom Fields.
- **Save** the document. 9

#### Change the Work Item type

You can quickly change a Work Item type to another type. For example, if the Work Item is currently a Task, you might change it to the Defect or Change Request type.

- 1. In the Table (or other view) select the item you want to change to a different Work Item type.
- On the Work Item Viewer/Editor toolbar, click the **Actions** button.

3. On the drop-down menu, choose **Change Type to**, and on the sub-menu, choose the type to which you want to convert the selected Work Item.

#### **Note**

You cannot change the Work Item type if the Type field is read-only in the project configuration.

# Create and link a new Work Item in a single operation

While focused on any Work Item, you can quickly create a new Work Item and a link, all in a single operation. You can create two basic link types this way, each having several possible link relationships:

- Linked Work Item: Create a new Work Item and an outgoing link from current item to the new item.
- Linking Work Item: Create a new Work Item and an incoming link to the current item from the new item.

# **Linking Work Item**

In the default project configurations in project templates, a new **Linking Work Item** can have any one of the following relationships to the current Work Item:

- 1. Relates to the selected item.
- 2. Depends on the selected item.
- 3. Duplicates the selected item.
- 4. Is the child of the selected item (has parent).
- 5. Has follow-up for the selected item.
- 6. Implements the selected item.

These relationships are configurable in **Administration Work Items Form Menus**.

## Create a new Linking Work Item

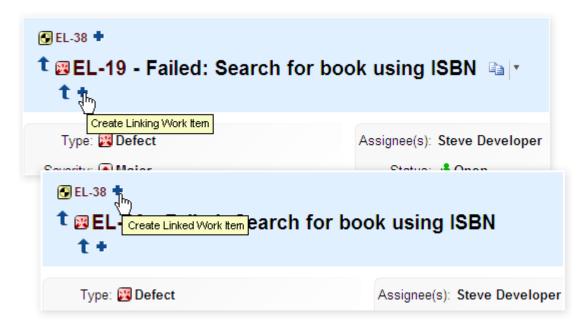
- 1. In the Work Item browser, select the Work Item in which you want to create a new linking Work Item.
- 2. In the Work Item editor toolbar, click to drop down the **Actions** menu.
- 3. In the menu, choose **Create Linking Work Item**. A sub-menu appears which has all currently configured options for the new Work Item type and the role of the link to the new Work Item.
- 4. In the sub-menu, choose the type and the link relationship for the new Work Item you want to create.

A new empty Work Item of the selected type is created and opened for editing in the Work Item editor.

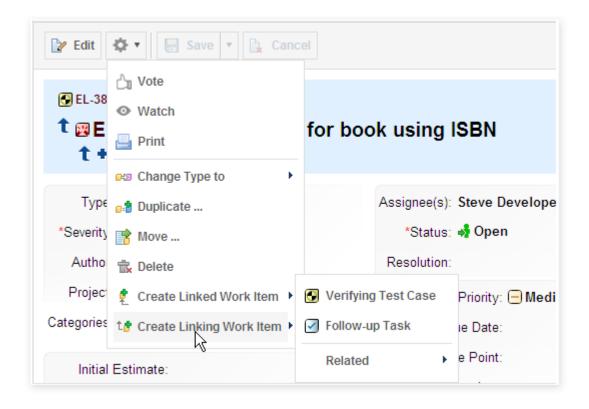
5. Fill in the Work Item data as needed and click Create to create the new Work Item in the system.

# Tip

You can also create new linking/linked Work Items using the icons on any Work Item's title.



(Or use the Actions menu.)



#### **Linked Work Item**

In the default project configurations in project templates, a new **Linked Work Item** can have any one of these relationships to the current Work Item:

- Is related to the selected item.
- Is depended upon by the selected item.
- Is duplicated by the selected item.
- Is parent of the selected item.
- Is follow-up of the selected item.
- Is implemented by the selected item.

These relationships are configurable in \*Administration \* Work Items \* Form Menus.

#### Create a Linked Work Item

- 1. In the Work Item browser, select the Work Item in which you want to create a new linked Work Item.
- 2. In the Work Item editor toolbar, click the **Actions** button to drop down the **Actions** menu.
- 3. In the **Actions** menu, choose **Create Linked Work Item**. A sub-menu appears which has all options for new Work Item type, and role of the new Work Item link.

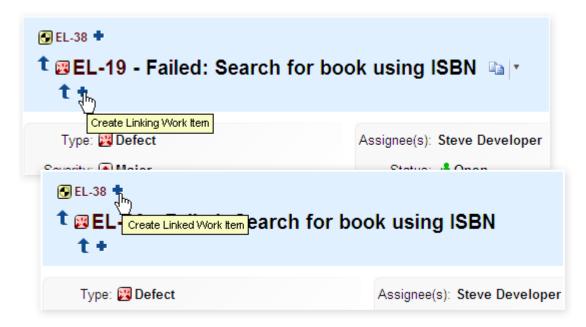
4. In the **Actions** sub-menu, choose the type, and the link relationship for the new Work Item you want to create.

A new empty Work Item of the selected type is created and opened for editing in the Work Item editor.

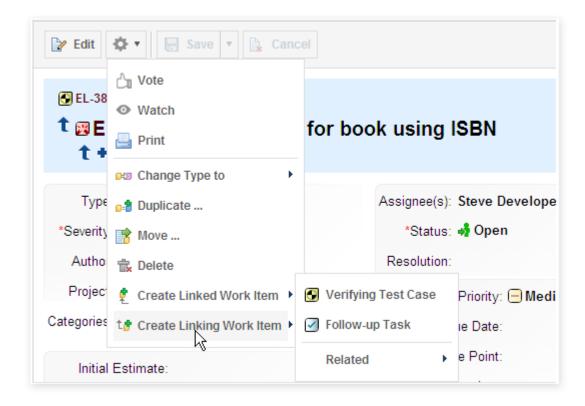
5. Fill in the Work Item data as needed, and click **Create** to create the new Work Item in the system.

# Tip

You can also create new linking/linked Work Items using the icons on any Work Item's title.



(Or use the Actions menu.)



# **Copy Work Item IDs**

When you select multiple Work Items in the Work Items table, an item "Copy IDs" appears on the Actions menu in the toolbar of the lower (detail) pane. It enables you to copy the IDs of all the currently selected Work Items to your clipboard. The IDs are rendered in the following format:

```
id: (projectID/WI-1 projectID/WI-2 projectID/WI-3)
```

This can be useful if you need to construct a text-based Lucene query for use in a script or script-base Widget. You can test it by clearing the current query in the Work Items table and pasting the copied string directly into the Query Builder field.

#### Caution

The copied string works only in text-based queries. If you paste the copied string into the "ID" element in the Query Builder it will not work. If the current query contains visual query elements, you can, however, append the string from your clipboard directly into the Query Builder field after the visual elements.

#### **Delete Work Items**

You may encounter situations where you believe some existing Work Items should be deleted. There are some important considerations before taking this decision. Generally, the following would be considered best practice for undertaking deletion of Work Items:

Do not delete the item if:

It is a Test Case Work Item that still remains within an existing Test Run.

- The Work Item is currently linked to any other Work Item(s).
- The Work Item was linked in the past (you can determine this from its History). The history of the formerly linked item(s) will contain links to the Work Item you propose to delete. If you delete it, traceability which might be needed at some future time will be broken.

Rather than deleting such Work Items, it is recommended to set its severity to the lowest value, and add an appropriate comment.

If the Work Item has never been linked to any other (you can verify this in its History), then you can delete it. Deleting a Work Item triggers a workitem-deleted system event which results in notifications being sent to the notification targets configured for that event in Administration. (See Notifications, Events, and Targets.) All deletions and removals of Work Items appear in the **Activity stream**.

#### Note

If the Work Item you want to delete is defined in a Document, it is recommended to delete the item via the Document Editor rather than the Table or other tracker view. See Work Item Recycle Bin.

Your user role must have the DELETE permission, and access rights for the repository folder that contains the Work Item. By default, the **project\_admin** role has this permission. (The System Administrator also has these permissions.)

To delete a Work Item:

- 1. After logging in with necessary permissions, open the project containing the Work Item(s) to be deleted.
- 2. Locate the item you want to delete in the Table view of the Work Items topic and select it. You may select multiple items (as for Bulk Edit) to delete all the selected items at once.
- 3. In the Work Item Viewer/Editor toolbar (lower part of the page), click on **Actions** and choose Delete.

(Note that if more than 20 items are selected for bulk editing, the Delete action appears along with several others in the Work Item detail pane and the Actions menu button is disabled.)

#### Tip

If the license you are using provides the Multi Edit view, the **Remove** button appears when a Work Item is selected for editing in this view. Use this icon for deleting a Work Item when working in Multi Edit.

## **Move Work Items**

#### **Understand Work Item storage types**

In order to understand how to move Work Items, some understanding of the underlying storage types is necessary. You can move Work Items from one type of storage to another.

XML Storage:

Work Items created in the integrated tracker exist in XML storage. Think of this as "standard" or "normal" storage for Work Items.

• **Document**: Work Items rendered in the user interface exist in XML at the lowest level and are stored in a container which maintains their structure. For example, rendered in a Document-like way and in the various views of the Document's Work Items (such as Table) in a tracker-like way.

## Tip

While there is no limit to the number of Work Items you can have in a LiveDoc, we recommend keeping it under 5000.

Module (legacy): Work Items in Polarion version 2010 and some earlier versions exist in
modules—special containers that maintain structure and sorting. They are deprecated as of
version 2011, but may still exist in projects created with earlier versions. In terms of storage,
Modules are the basis for Documents. The underlying storage is quite similar, the user interface
is quite different.

Moving of Work Items from one type of storage to another is probably not something you would do frequently, but you may occasionally encounter a need for it. Users of pre-2011 Modules in particular need to use Work Item moving to migrate from Module to Documents. Another scenario could be Work Items such as requirements or test cases that were originally defined in an office document and created in a Polarion Live Document during import from Word. You might prefer to manage such items in standard tracker storage, so you would move them out of the Document.

Keep in mind the following information when planning to move Work Items:

- Structure, if any, is preserved.
- A Work Item cannot be moved to a Document unless its type has been specified in the Document's presentation configuration. For example, a Test Case type item cannot be moved to a Document in which only the Requirement type is defined. You would need to add the Test Case type to the Document's presentation configuration before the move.
- Document headings are not moved out of Documents or between Documents. Only Work Item title and description are moved (along with properties).
- When a Work Item is moved to a Document from a Module, and the link roles for the type in the Document are different form those of the type in the Module, the link roles will convert to the Document roles.

## Move from a Module to a Document

You can move Work Items stored in legacy pre-2011 Module to a Live Document ("Document"). You need to create the target Document before initiating the move. This migration requires an intermediate move to XML storage, then from XML storage to the target Document. Please contact Polarion Technical Support if migrating legacy document content is an issue for you.

#### Move between Documents

You can move Work Items stored in one Document to another Document. For example, both requirements and test cases might have been defined in the same office document that was imported into Polarion, so both Work Item types exist in one Document, and you want to move the test cases to a separate Document.

# When moving Work Items between Documents

- A Document structure role can only be used for Derived document structure links. (It's not possible to create links with this link role for a Work Item that's contained within a Document.)
- When moving a Work Item from one Document to another, all existing links that use the Reserved Document structure link role are removed.

### qiT

Have a dedicated role for your Document structure role. (One that cannot be used for normal Work Item links.)

#### Move Work Items to another Document

- 1. Create a Document to receive the moved Work Items if a target Document does not already exist. It can be in the same space as the source Document, or another space.
- 2. Open the Document containing the Work Items you want to move, and open the Table view to see the items in tracker format.
- 3. Select the items you want to move (check the box to the left of each item, as you would for Bulk Edit). You can run a query to reduce the number of items in the table to make it easier to browse to and select the ones you want to move.
- 4. In the Work Item Editor toolbar, click , and choose Move. The Move Work Item(s) dialog appears.
- 5. In the dialog, select the **Into Document** option. The **Space** and **Document** lists become enabled.
- 6. If the target Document resides in a space other than the one shown in the **Space** list, select the correct space in the list.
- 7. Select the target Document in the **Document** list, and click **OK**.

See also Structuring a Document's Work Items.

#### Move from Document to XML

1. Open the Document containing the Work Items you want to move, and open the Table view to see the items in tracker format.

- 2. Select the items you want to move (check the box to the left of each item, as you would for Bulk Edit). You can run a query to reduce the number of items in the table and make it easier to browse to and select the ones you want to move.
- 3. In the Work Item Editor toolbar, click , and choose Move. The Move Work Item(s) dialog appears.
- 4. In the dialog, select the **Out of Document** option and click **OK**.

#### Move from XML to a Document

You can move Work Items from standard XML storage in the track to a Live Document ("Document"). For example, you might discover that someone created new Work Items directly in the tracker which should really be part of a Document.

- 1. If the target Document to receive the moved Work Items does not already exist, create a new Document.
- 2. In **Navigation**, select the **Work Items** topic. Optionally use one of the sub-topics to filter for a single Work Item type.
- 3. In the Table view, select the items you want to move (check the box to the left of each item, as you would for Bulk Edit). You can run a query to reduce the number of items in the table to make it easier to browse to and select the ones you want to move.
- 4. In the Work Item Editor toolbar, click , and choose Move. The Move Work Item(s) dialog appears.
- 5. In the dialog, select the **Into Document** option. The **Space** and **Document** lists become enabled.
- 6. If the target Document resides in a space other than the one shown in the **Space** list, select the correct space in the list.
- 7. Select the target Document in the **Document**, list and click **OK**.

# **Use Excel Round-trip**

# **Excel Round-trip overview**

When you export any set of Work Items using the **xlsx: Microsoft Excel** option, the output is written to a special round-trip format for Microsoft Excel. An external user may optionally edit Work Item content in Excel and/or create new Work Items for import into Polarion. A Polarion user with the necessary permissions can then import the edited Excel workbook back to Polarion, updating the Work Item content from the externally edited Excel worksheet, and/or creating new Work Items defined externally in the exported Excel workbook.

If the Work Items were exported from a Document with the necessary export options, it is also possible for an external user to change the Document's structure externally in Excel. The structural changes are reflected in the online Document when the Excel worksheet changes are re-imported to Polarion.

Excel export may be run from any scope: repository, project group, or project. Re-importing of an Excel workbook which contains modified Work Items can also be done in any scope. However, re-importing of Excel workbooks which contain new Work Items is only allowed in the project scope.

Exported Work Item fields which are writable in the portal may be optionally locked in the **Export Work** Items dialog during the export procedure so that users of the exported Excel file cannot easily change the field value. If an advanced Excel user does manage to change the value of a locked field in Excel, the change is ignored when the Excel file is imported back into Polarion. (Note that fields which are read-only in the portal display the locked lock icon in the dialog, and are not unlocked when clicked.)

Images are not currently supported in the Description field, or other rich-text fields, of Work Items modified or created in an exported Excel round-trip worksheet. If images are included, they will be lost when the workbook is re-imported into Polarion. A warning is displayed and the user performing re-import may opt to cancel the operation.

Also, any advanced text formatting and lists (bullet, etc.) implemented in the exported Excel document are lost upon re-import to Polarion.

Mathematical formulas contained in rich test fields of Work Items are subject to the same limitations as images, because they are rendered as images when not in edit mode.

For procedures, see:

- Exporting for Excel Round-trip.
- Re-importing an Excel Round-trip Document

## **Excel Round-trip templates**

Formatting of the Excel Round-trip output is controlled by an export template. Polarion provides, and by default uses, an Excel template named Basic.xlsx. Additional templates are provided in the original Polarion installation and are available in the **Template** field of the Export Work Items dialog when exporting for Excel Round-trip. An administrator can download any export template and modify it locally to create a customized version of the original template, and/or additional templates. When uploaded to the portal, the template names appear in the **Template** field where users exporting Work Items can select which one they wish to use for the round-trip export to Excel.

An export template may also specify which columns should be locked in the exported Excel file when an export operation uses the export template. This both saves time for end users, and provides control to prevent external or otherwise unauthorized users from inappropriately changing exported data, while permitting them to view it. Even if a user of an exported template-based Excel file manages to change data in a locked column, the change is ignored on re-import to Polarion.

Excel export for offline Test Case execution supports export of Test Cases with multiple Iterations of Test Steps. By default, the export template is configured to append a label for each Iteration to the Title column in the exported workbook. It is possible to configure which field displays the Iteration number in the property Append Iteration Label to Field in the hidden **Polarion** sheet. The default is the **Title** field.

For more information, see: Customize Export Templates.

The default templates can contain the same expandable variables as used in PDF or Word export, as well as one for a query (in cells and header/footer). For information on these variables, see Configure PDF Export.

#### **Comments in Excel Round-trip**

Native Excel comments present in cells are imported to Polarion as new Work Item comments. The Polarion tab of the Excel Round-trip export template contains the property New Comments Column which contains a format for the Comments column header. The default title for imported comments is the value of the header of the comments column.

The worksheet user can create new comments in the exported worksheet by adding a column having a header <code>comments</code> by <code>[USER'S NAME]</code> (where <code>[USER'S NAME]</code> is the commenter's name), and then filling in comment text in the appropriate row. Example: <code>comments</code> by <code>John Doe</code>

When the worksheet is re-imported to Polarion, the **Comments by** cells will produce new comments in the Work Items. The title of imported comments contains the text of comment column header (in the imported worksheet) plus the name of the comment author. For example: <code>comments by [USER NAME]</code>.

If Work Items containing Document comments are exported for Excel Round-trip, and the Work Item **Description** field is allowed by the exporting user to be editable in the exported Excel workbook. A placeholder {comment:id} is inserted into description text in the exported Excel workbook. This preserves the comment marker for the round-trip process and should not be removed from the exported Excel document.

### **Define new Work Items**

The user of an exported Excel workbook can define a new Work Item by adding a new row in the table of exported Work Items on Worksheet one. The user should **leave the ID column blank**. The Excel Round-trip importer creates IDs for newly defined Work Items during the round-trip import process.

To insert a row into the Work Items table in the Excel round-trip document, select a cell in the table, right click on the row number and select **Insert**. A new row is inserted before the current row. Note that in a newly exported Excel document, one empty row is present. This row is locked and is present only so the user can insert new rows above it for new Work Items. It remains as the last row in the table, and its presence does not affect anything in the re-import to Polarion process. If you encounter a table with no empty first row present (perhaps exported from a Polarion version where the exporter did not create an empty row), you can add a first row, but it will not be editable, as just described. Simply add another row above it and proceed to define a new Work Item.

Creation of new Work Items via Excel Round-trip import is only allowed in the project scope (i.e. not in project group or repository scopes). Be sure you have the correct project open when you import Excel workbook in which new Work Items are defined.

### **Note**

You can only import the Work Item **Type** defined by the **Excel Export Template** or the default Work Item **Type** of the project you're working in.

You will not be able to select types from the Excel drop-down menu.

Workflow configuration for projects often includes some initial workflow action which is executed for newly created Work Items. For example, a value may be set for the **Status** field. If no initial action is specified for the Work Item type in the project workflow, then the user-specified values for all editable fields will be set in Polarion when the Excel workbook is re-imported into Polarion. However, if an initial workflow action **is** configured, then any field values set by the initial workflow action will override user-specified values set in the Excel sheet. For example, suppose the Excel user creates a new Work Item in Excel, and sets the **Status** field to **Accepted**. If the project's workflow configuration has an initial action which sets the value of the **Status** to **Draft** for new Work Items of the new

item's type, then the workflow action's setting overrides the Excel user's setting, and the Status field will be set to **Draft** after the re-import operation.

# **Export Work Items for Excel Round-trip**

# ALM REQ QA

For a general overview of exporting Work Items, please review: Export Work Items.

- 1. Follow the procedure described in the Basic Export Procedure section.
- 2. In the Format field, select xlsx: Microsoft Excel
- 3. In the **Template** field, choose the template Polarion should use to format the data in the exported Excel workbook.
- 4. Optionally change the items in the **Selected Columns** list.
- 5. Optionally lock non-read only columns in the exported Excel worksheet by clicking the lock icon next to the column name in the Selected Columns list.

# Fields and Types of custom fields that are exported as read-only

Fields	Types of Custom Fields	
Approvals	Enum (Multi valued)	
Assignee(s)	Feature Selection	
Attachments	pvSCL	
Categories	Test Steps	
Comments		
Created		
Document		
Externally Linked Work Items		
Has Suspect Links		
Hyperlinks		
ID		
Linked Data Resources		
Linked Revisions		

Fields	Types of Custom Fields
Linked Work Items	
Outline Number	
Planning Constraints	
Planned End	
Planned In	
Planned Start	
Project	
Repository, Project	
Status	
Туре	
Updated	
Work Records	

## **Note**

Custom **Enum** fields can be edited during the round-trip process, but only if they do not **Multi** valued flag in their definition.

# Tip

If the Work Items being exported to Excel are contained in a Document, the Export
Work Items dialog provides the advanced option Allow Document structure
changes in Excel. If checked, then an external user will be able to modify the
structure of the online Document in Excel.

After export, all columns will be editable in the exported Excel workbook regardless of their locked/unlocked status in the export dialog. All externally made structural changes will be reflected in the source Document when the exported and modified Excel workbook is re-imported to Polarion. The only exception is Work Item fields that are always marked as locked (i.e. read-only) by Polarion in the export dialog... Linked Work Items, for example. Such fields, when exported with the option to allow structural changes, will not be locked in the exported workbook, and can be changed by the user. However, any change to these fields will be ignored by the importer when the workbook is re-imported to Polarion.

Structure in the exported workbook is shown by indentation. The first column is left-aligned to that indentation is visible.

- You can optionally view the export template before finishing the export by clicking on the **Show template** link. You are then prompted to download and save the template file, or download and open it in Microsoft Excel.
- Columns used in the template for some calculation cannot be removed from the Selected Columns list because they must be present in the exported file in order for the calculation(s) defined in the template to work. Such columns appear in Italic font.
- In addition to the **Add** and **Remove** buttons, columns can be added to or removed from the list boxes by double-clicking a column name in either list.

## Re-import an exported Excel document

# ALM REQ QA

An exported Excel workbook that has been modified can subsequently be re-imported to Polarion. All the Work Items modified in the document are updated with the changes from the document, a new entry is logged in history, and a workitem\_updated notification event is triggered for changed Work Items causing the notifications to be sent as defined in the notifications configuration.

If new Work Items were defined in the Excel workbook, they are created in Polarion by the importer. The following points are useful to note:

- If Auto-assignment is configured for the project, the new items are automatically assigned and notifications are sent to the assignee(s).
- It is possible to import the Status field when importing new items. If that field is empty upon import, then the following rules apply:
  - The initial workflow action (defined in the project's workflow configuration) is executed for the new items.
  - If the initial workflow action is not specified in the workflow configuration, then the value is assigned as described in: Using the Workflow Designer in the paragraphs about initial workflow action.

An Excel workbook can only be imported into the Polarion project from which it was originally exported. If imported to a different project, the import fails with a message, and a log file, viewable in the browser, is generated which contains details. It is possible to export Work Items from the project group scope, but such export is "one way"... Excel workbooks exported from a project group and then modified cannot be re-imported.

To re-import a modified Excel workbook:

- 1. Open the project from which the Excel workbook was exported, and open the Table view of the Work Items topic in Navigation.
- 2. In the Table view, click \* Import to launch the Import Work Items dialog.
- 3. In the Format field, select xlsx: Microsoft Excel.

- In the File field, click Browse and select the file you want to re-import.
   If you want to overwrite any conflicting information in the portal with content from the Excel workbook, check Overwrite Conflicts.
- 5. Click **Start** to upload the local file and launch the import process.

If the import succeeds, the **Import Work Items** displays the message **Import finished**. If the import fails, the message **Import failed** appears. In both cases the **Show Log** link is present. Click it to display a log of the import process in a new browser instance. The log reports any conflicts recognized during import and what was done with them.

# Potential re-import conflicts and how Polarion treats them

Field Value (Polarion)	Field Value (Excel)	What Polarion does
A	Α	No change and no conflict reported.
A	С	Change done in Excel is reimported into Polarion. (Updates automatically. No conflict reported.)
В	С	Change in Polarion conflicts with a change in Excel. (Polarion Identifies it as a conflict. Can be overwritten.)
B (Where A is changed to B in Polarion after a file has been exported to Excel.)	А	Polarion will recognize that the value in Polarion is more recent than the Excel value and will not overwrite it, or report it as a conflict.

# Work Item translations

### **View Work Item translations**

Polarion can be configured to support multiple human language translations of Work Items. For example, an international company could write requirements in English, and provide translations for Japanese-speaking suppliers in Japan and French-speaking key customers in France, Canada, or Africa.

When correctly configured, a Title and Description field for one or more additional languages appear in the Custom Fields section of the Work Item form in tracker views, and when Work Items are contained in Documents, users can switch the language. After an administrator configures your system for multi-language support, your translators can supply a translation of Work Item titles and descriptions previously written in the default language. After translation, users who access Work Items can see the content in their preferred language.

The basic process for adding support for multiple languages in Work Items follows (some steps require Administrator permissions):

Enable multiple language support in the system configuration (in Administration).

- For each translated language, define custom fields to store translated text of Work Item titles and descriptions (also in Administration).
- In projects existing prior to configuration for multiple languages, update existing Work Items to supply translated texts for Title and Description for each language configured.

Information on configuring multi-language translation for Work Items can be found at: Configure Multi-language Support.

## **Translate Work Items**

The assumption is that Work Items will first be written in some default language, and translation(s) will be done afterwards. Translation is always performed in the Tracker view. If a Work Item exists in a Document, the translator must switch to that view.

To translate a Work Item contained in a Document:

- 1. Open the Document and select the Work Item to be translated.
- 2. Click the Work Item's icon in the left margin and choose **Open Item**. The Work Item opens in the Tracker view.
- 3. Scroll to the Custom Fields section of the Work Item form. You should see fields for the title and description of the target language.
- 4. Update the translation fields with your translation text and click **Save**.

# **Switch languages in a Document**

When the project is configured for multi-language Work Items, a menu item Switch Language appears under the menu of the Document Editor. You can choose any of the configured languages from this menu.

When you choose a language, the translation of the Work Item titles and descriptions replaces the previously selected language. Several points are important to remember when viewing translations in a Document:

- Translated text is read-only. You can, however, edit Work Item Properties in the Document Sidebar.
- You can print the translated document or export it to PDF so that title and/or description is translated in the output. PDF export can be configured so that a translated language is written into header and/or footer when the translated Document is exported.
- History is translated, but translation does not appear in historical comparisons.
- If a translated title or description field is not found when you switch the language, then the content of the default language field is shown.

# **Warning**

When opening a Document in the read-only translated view, all Work Items contained in the Document are loaded. If there are many Work Items, it can take some time for the Document to open. This differs from the normal view, in which large Documents with many Work Items open quickly due to a background process that loads Work Items as needed as the user scrolls (assuming that the background processing is enabled in the system configuration).

# **Documents and Pages**

# ALM REQ QA PRO

The **Documents and Pages** topic provides access to project artifacts such as specification documents, reports, and informational content. You can access and create the following types of artifacts:

- LiveDoc documents, referred to in Help as Documents, are special document-like containers for both free-form text and granular Work Items. For example, in a requirements specification Document you have the possibility to mark content elements as Requirement-type Work Items that can be linked for traceability, workflow controlled, assigned, planned, scheduled, reported, and managed throughout the artifact lifecycle. Documents provide an ideal interface for document-oriented and nontechnical authors of specifications.
- Pages contain information that does not need to be tracked and workflow-controlled through a process as Work Items.

*Info Pages* enable you to create general information such as internal documentation, guidelines, meeting notes, preliminary specifications, etc. The content can have rich text formatting, and include images, tables, and more.

LiveReport Pages are intended for building reports visually using a catalog of *Widgets*, which enable nontechnical users to build robust online reports that retrieve, aggregate, and render information from one or more projects, or an entire repository.

#### Tip

Widgets are available with both types of Pages, and work the same way in both. The default layouts differ slightly to facilitate the use cases, but you can build reports or purely informational pages using either type, according to your own preference.

Classic Wiki pages use Polarion's legacy free-form content management technology. Content
for these pages is implemented using Wiki mark-up language, and can include Velocity scripts,
Javascript, and HTML.

## Caution

Classic Wiki pages are currently supported for backward-compatibility with existing Wiki pages created prior to Polarion version 2015. That version introduced Pages, which are intended to replace Classic Wiki, which will eventually be phased out of the platform. You should use Pages for all new reports and information pages, and plan to reconstruct any reports and information pages you will need in the long term using LiveReport and Info Pages.

Polarion provides fine-grained access control in the Administration area. Administrators can control users' access to Documents, Pages, Work Items, and even Work Item data fields. You may be able to view some types of content, but not modify it. If you find you cannot view or modify information that you need, discuss your user permissions with your team leader or Polarion administrator.

# **Test Runs**

# ALM REQ QA PRO REV

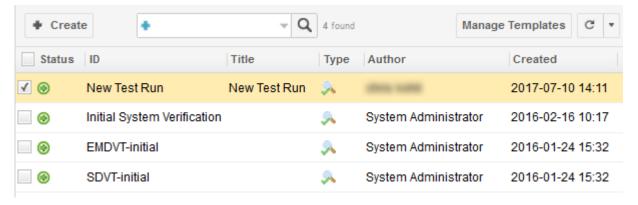
The **ATest Runs** topic provides access to Test Runs and Test Run templates.

In this topic you can:

- View, edit, export to PDF, and delete existing Test Runs.
- View the history, properties, and execution records of any existing Test Run.
- Create new Test Runs.
- Manage Test Run templates.

#### Test Runs page

The Test Runs page has two parts. The top pane provides a table of existing Test Runs. It also provides a toolbar enabling you to:

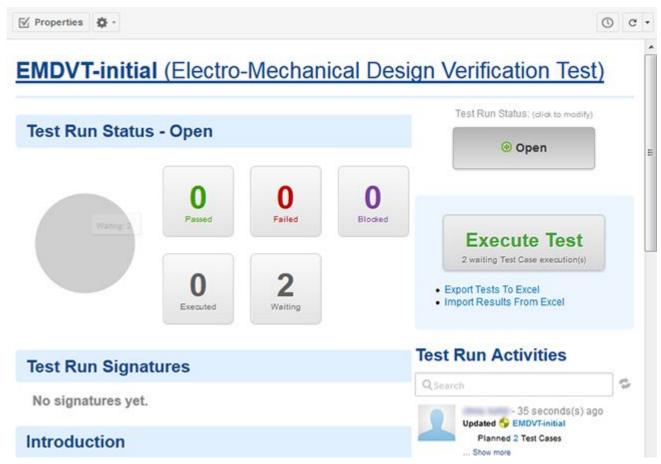


Right click on a column to hide or view additional columns.

Create a new Test Run.

- Search existing Test Runs.
- Manage Test Run templates (including Test Parameters in templates)
- Refresh the table of Test Runs.

Clicking on any Test Run selects it in the top pane, and loads the detail of the selected Test Run in the bottom pane.



Click the (Actions) icon for available Actions.

# The bottom pane:

- Displays information about the status of the Test Run (passed or failed).
- Provides links to passed and failed Test Cases, errors, and executed Test Cases
- Provides information about the test environment and build tested, including a link to that build.
- Summarizes and links to other tests related to the same build, if any.
- Lists problems found during execution of the Test Run, if any.
- Provides access to the Test Run's history.

• Provides access to any Attachments. For example, if the Test Run comprises a set of JUnit tests, a document containing the specifications for those tests might be attached.

The following actions are available from the toolbar of the Test Run detail pane:

- View Test Run properties, including Test Parameters defined in Test Runs. Market Properties.

- Edit the Test Run page. (Actions) Customize Test Run Page.
- Select the Test Cases to be executed for the Test Run. 

  (Actions) 

  ✓ Select Test Cases.

  For more information see: Select Test Cases.
- Save the current Test Run as a template on which to base other Test Runs; ☑ (Actions) → ☐ Save As Template. If you are viewing the page of a Test Run Template, you can create another customizable template by clicking ☑ (Actions) → Duplicate Template.
- Export the selected Test Run detail to PDF. (Actions) → Export to PDF.
- Print the selected Test Run detail on an available printing device. (Actions) Print.
- Delete the selected Test Run. (Actions) → Delete.

  You can delete multiple Test Runs by first checking the boxes on relevant rows in the Test Runs table. When multiple Test Runs are selected, a button appears in the lower pane which will delete all the selected Test Runs in a single operation.

#### Where to find more information

The Test Runs page is one of several features related to Test Management. These features are covered elsewhere in the documentation.

- If you are a user of Test Management features, please see: Polarion for Testing and Quality Assurance.
- If you are an administrator needing to configure Test Management, please see: Configure Testing.

# **Baselines**

# **About Baselines**

# ALM REQ QA

The **Baselines** topic appears in the **Navigation** pane when a project is open. (The topic is hidden when working in the Repository scope.) Browsing and reading Baselines with **Polarion QA** requires the presence of a license for **Polarion ALM** and/or **Polarion Requirements** on the server.

Baselines capture and preserve the state of your project at some point in time. Baselines are typically created for major development milestones and for releases. It is possible to generate comparison reports that compare two baselines and report on the changes that took place between the earlier and the later baseline.

The **Baseline Time Machine** feature enables users using a supporting license to browse any baseline, just as if it were the current project state. All artifacts appear in the state they were when the baseline was created, and queries and reports retrieve only information from the baseline.

## Create a new Baseline

# ALM REQ

You can create a new baseline any time you deem it necessary. Your team or company may have a policy that specifies when baselines must be captured. Basically, you should create a baseline whenever the project reaches some state that might need to be reviewed at some later time. If it turns out you really don't need some baselines, you can always delete them. You must have permissions for any baseline-related activity you want to do: creating, modifying, viewing, reporting, etc.

- 1. Open the Project for which you want to create a new baseline.
- 2. In the **Navigation** pane, select the **Baselines** topic.

The Baselines page displays a table of all existing baselines, if any. There is always a row for the current state of the project.

- Click Create in the page toolbar. The Create New Baseline dialog box appears.
- 4. Enter a name for the baseline in the **Name** field. This name will uniquely identify the baseline.
- Select a revision to associate with the baseline you are creating in one of these ways:
  - In the **Revision** field, select a repository revision. The head revision is the default. It captures the current Head revision of the project, with all artifacts, as of the moment the system creates the baseline.
  - In the Revision field, select Specific. Use the Revision Picker dialog box to select the desired revision.
- 6. In the **Description** field, enter some descriptive text so that others can understand the nature of this baseline later to see if they want to use it in a baseline comparison report.
- 7. Click **Finish** to create the new baseline. It now appears as a row in the **Baselines** table.

# Modify an existing Baseline

# ALM REQ

It is possible to modify the name and/or description of an existing baseline. It is not possible to alter the artifacts captured in the baseline.

1. Open the project that contains the baseline.

- 2. In the **Navigation** pane, click the **Baselines** topic.
- 3. Locate the baseline you want to edit in the Baselines page and click (Edit ) in the Actions column of the baseline's row.
- 4. Edit information as desired and click Finish.

# **Compare Baselines**

# ALM REQ

You can generate an on-screen printable report that compares two or more baselines to each other or any baseline to the head/current revision.

To generate a baseline comparison report:

- 1. Open the project in which you want to compare baselines.
- 2. In the **Navigation** pane, click the **Baselines** topic.
- 3. In the Baselines page, select two baselines you want to compare by checking the box on the relevant baseline's row in the table. If you want to compare a baseline against the head revision, check the Head/Current row and the desired baseline's row.
- 4. Click **Compare** to generate the comparison report.

The baseline comparison report job launched. You can check the status of the job in the **Monitor** topic. When the report generation job is finished, a row is added to the table of reports in the Comparison Reports section of the Baselines page. This section is hidden by default. Click **Show** Comparison Reports to reveal or re-hide the section. The Actions column provides two actions for the generated report: Remove and Show Report.

# View Baseline comparison reports

# ALM REQ QA

### **Note**

Viewing with Polarion QA requires presence of a Polarion ALM or Polarion Requirements license on the server

The selected baseline comparison report displays as a modal dialog after you click the **Show Report** action in the table of generated reports. The report contains several sections:

- **Timepoints**: Compares activity in Timepoints between the revision(s) and/or baseline(s).
- Categories: Compares activity in Categories between the revision(s) and/or baseline(s).
- Work Items: Compares activity in Work Items between the revision(s) and/or baseline(s).

• **Plans**: Compares activity in Plans (changes of Plan fields and added/removed Work Items) between the revision(s) and/or baseline(s).

### Tip

You can expand and collapse each section using the icon at the right-hand edge of its title bar.

# Browse, search and report Baselines

# ALM REQ QA

## **Note**

View with **Polarion QA**: Requires the presence of a **Polarion ALM** or **Polarion Requirements** license on the server.

The **Baselines Time Machine** feature is so-named because it enables you to go back in time and browse, search, view, or run reports in a baseline just as if the baseline were the current project state. Unlike in the project's current state, in a baseline, no artifacts can be changed, and current artifacts that did not exist at the time of the baseline do not appear when browsing and will not be found by searches or report queries.

To enter a baseline via Time Machine:

- 1. Locate the baseline you want to explore on the ►Baselines page (Navigation → ►Baselines).
- 2. Click on the name of the baseline in the **Name** column.

You are now browsing the project in the state it was when the baseline was captured. You can always tell that you are working in a baseline by the banner showing the baseline name, which appears in Navigation below the project name.

To exit the baseline and return to the current project state, click the closer on the baseline banner in Navigation



# **Builds**

# ALM REQ QA PRO

### **Note**

This feature is also available in **Polarion Requirements**, but it is hidden in the default configuration. The topic can be shown in **Navigation** by changing the Navigation topics configuration. For more information, see: Configure Interface Views.

This feature is read-only in **Polarion Reviewer**.

# Scope(s): Project

The **Builds** topic of Navigation enables you to access the current and recent builds of any project. You can find information about the build job(s), create a new build, and access the build results.

The Builds page has two parts. The top half shows a table of recent builds with some basic information about each build listed. It also provides the Create Build for launching a new build, and the **Refresh** button which refreshes the page with any new builds that may have been launched since you accessed the **Builds** topic.

The lower half provides information about the build currently selected in the top half of the page. The Build Results section provides access to whatever artifacts the build is configured to generate (often files distribution, installation, etc.) There are also links that provide access to the build log and base directory.

#### Where to find more information

The **Builds** topic is an access gateway to the results of builds resulting from Polarion's extensive build-management capabilities.

For information on configuring builds and build management, see:

- **Basic Build Management**
- Advanced Build Management.

For more information on working with builds, see:

Working with Builds

# **Quality and metrics**

# **About quality and metrics**

# ALM

The Quality topic provides dashboards with key statistics and results of audits and metrics that can be of interest to QA managers and teams. The information is live based on the actual state of work on the projects that feed data into the dashboard.

The dashboard's information can be viewed for any of three scopes:

- **Global**: Statistics are rolled up for all projects managed with Polarion to provide a high-level view of the current state of quality across your entire organization. To access this scope, open the repository.
- **Project Group**: Statistics are rolled up for all projects in a selected project group (which corresponds to a folder in the repository). This gives you a high-level overview of overall quality for all projects in the group. To access this scope, open the desired project group. Then select the **Quality** topic in the **Navigation** pane.
- Project: Statistics are rolled up for a currently open project to provide a picture of the current state of the project's quality. To access this scope, open the desired project, then select the \times \textbf{Quality} topic in the Navigation pane.



Links in the dashboard for a broader scope enable you to drill down to the narrower scope(s), and into different projects.

For project groups and repositories, the dashboard displays a score with a radar chart resulting from the Process Audit (processaudit) calculation. Data are aggregated from child projects and

project groups. There is also a table listing scores for individual disciplines of the child projects and project groups.

In the project scope, the Quality topic has several sub-topics which load dashboards with different information about the current level of quality in the project. This information includes quality audits and metrics.

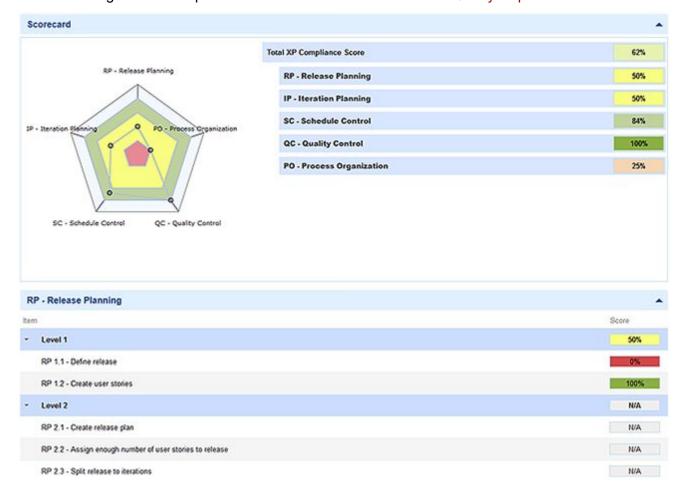
### Process dashboard

# Scope: Project

This dashboard reports statistics about compliance with a process. The data are the result of Process Audit analysis (calculation processaudit). The dashboard contains a radar chart and a table showing project score (a percentage rating, 0% to 100%) in several "disciplines" as measured by Process Audit: Release Planning, Iteration Planning, Schedule Control, Quality Control, and Process Organization. A total score is also displayed, calculated as the arithmetical mean value of individual scores.

In addition to statistical information, this dashboard provides detailed embedded knowledge about the process. For every "discipline" there is a detail table with list of rules and scores. You can click on a rule and see a page with details about meaning of the rule, violations found, how to fix non-compliance, and configuration possibilities.

You can configure some aspects of this dashboard. See: Process Quality Report.



# Work Items dashboard

Scope: Project

The dashboard in the Work Items topic provides some key metrics and statistics about the current state of Work Items in the currently selected project. It shows such things as counts of unresolved and unassigned Work Items. The **Quality Audit** metric in the **Metrics** portlet can be of interest for quality assurance teams and managers.

Work Item metrics and statistics are calculated by the trackeranalysis calculation. It is possible to configure which metrics and statics are displayed, their labels, and also layout of the dashboard page. For information, see Work Item Quality Report.

# **Testing dashboard**

This dashboard provides a view on the current state of unit test coverage and success ratio of unit tests. It also provides drill-down links to details about coverage and to your unit tests. It shows unit test success ratio and coverage for Maven artifacts with Java sources and tests. The detailed Maven unit test and coverage reports are linked.

Data shown in this topic are updated either during the Process Audit calculation (processaudit) or when the artifact is built using the option Update reports together with the build.

For more information, see Complex and Custom Building.



Testing statistics and drill-down from the Testing dashboard.

#### CMMI dashboard

Scope: Project

This dashboard is of interest if your organization implements compliance with the CMMI maturity model. Using actual data about the status of a project, it reflects the current level of compliance with all the indicators defined by CMMI. The dashboard can help organizations learn about CMMI, as all the points for Level 3 compliance are shown in the dashboard.

# **Reports**

# **About reports**

# ALM REQ QA PRO

#### Note

This feature is available in Polarion Requirements, but it is hidden in the default configuration. The topic can be shown in Navigation by changing the **Navigation** topics configuration. For more information, see Configure Interface Views.

This feature is read-only in Polarion Reviewer.

Scope: Project

The Reports topic provides access to a number of reports related to project source code, and enables you to explicitly launch report calculation.

The Reports topic provides access to the following reports:

- Maven Site
- **Unit Test Coverage**
- **Unit Tests**
- Javadoc
- **XRef**

#### Mavin site

This sub-topic provides access to the latest update of any Maven Site build artifacts of the project. You can use the Calculate Report to explicitly invoke calculation of the report to get an up-to-the-minute version.

The **Maven Site** page contains a list of build artifacts, if any have been configured for the project. (See Basic Build Management for information). You can drill down into the generated Maven reports for any listed build artifact by clicking on its name. The following reports are generated:

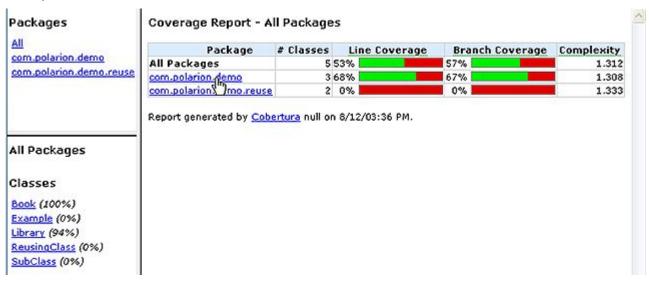
- **Checkstyle**: Report on coding style conventions
- **CPD Report**: Duplicate code detection
- **PMD Report**: Verification of coding rules
- Source Xref: HTML based cross-reference version of Java source code
- Test Source Xref: HTML based cross-reference version of Java test source code

Click on the name of any of these reports to drill down into the content.

# Unit test coverage

This sub-topic provides access to the latest update of the unit test coverage report, which provides information on how much of the project source code is covered by unit tests. You can use the Calculate Report to explicitly invoke calculation of the report to get an up-to-the-minute version.

The Unit Test Coverage page contains a list of build artifacts, if any have been configured for the project. See Basic Build Management for information. You can drill down into the generated code coverage reports for any listed build artifact by clicking on its name. The report appears in three frames, with various links that drill down into more detail... into specific packages or classes, for example.



Top-level unit test coverage report.

### **Unit tests**

This sub-topic provides access to the latest update of the unit tests report, which provides information about and access to the project's unit tests. You can use the **Calculate Report** to explicitly invoke calculation of the report to get an up-to-the-minute version.

The **Unit Tests** page contains a list of build artifacts, if any have been configured for the project. See Basic Build Management for information. You can drill down into the generated unit tests report for any listed build artifact by clicking on its name. The report is built and formatted by Maven (Surefire report).

The report has three main sections:

- Summary: Provides statistics about the project unit tests.
- Package List: Lists the project packages and the test classes for those packages.
- Test Cases: Lists the test cases in the project and enables you to access the sources.

#### **Javadoc**

This sub-topic provides access to the latest update of the project's Javadoc (for projects implemented in Java). You can use the **Calculate Report** to explicitly invoke calculation of the report to get an up-to-the-minute version.

The **Javadoc** page contains a list of build artifacts, if any have been configured for the project. See Basic Build Management for information. You can drill down into the generated Javadoc for any listed build artifact by clicking on its name.

## **Xref**

This sub-topic provides access to the latest update of the project's source code Xref report. You can use the Calculate Report to explicitly invoke calculation of the report to get an up-to-the-minute version.

The Xref page contains a list of build artifacts, if any have been configured for the project. See Basic Build Management for information. You can drill down into the generated Javadoc for any listed build artifact by clicking on its name. The generated report is a framed, cross-linked HTML report of the project source code. You can look at specific packages to see the classes and drill down into the actual source code of any class. The access is read-only. You cannot modify any source code.

# **Monitor**

Jobs are background processes which the server executes. Updating the Live Plan or a dashboard, or running a build are examples of jobs. The Monitor topic provides access to information about currently-running and scheduled jobs. Users with the required permissions can schedule jobs and explicitly launch jobs.

Scheduling and running jobs are, for the most part, a topic for administrators. However, there may be times when, as an end user, you need to check the status or outcome of some job. You can do that in the Monitor topic. The topic is available when working in a project or in the global (Repository) scope.

## Jobs pane

**Jobs** pane of the Monitor page shows any currently-running jobs and various details such as when it was started, and the current status.

For example, if you click the **Update** link of some dashboard and go to the **Monitor** topic, you see the job in the Jobs pane. The Status column indicated whan a job is finished, and you can remove the job by clicking **Remove Finished Jobs**, which is enabled on completion of the selected job.

#### Warning

Jobs consume system resources and can significantly slow the system for other users, depending on the type and scope of the job. It's a good idea to bear this in mind when deciding to explicitly run a job.

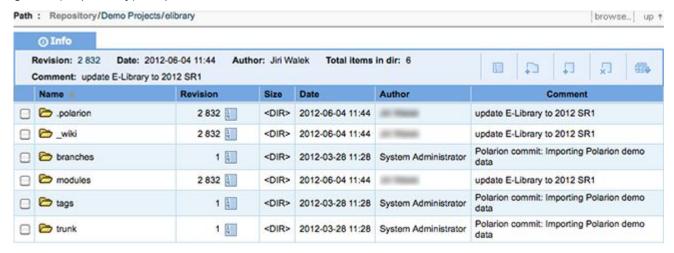
#### Scheduled Jobs pane

The **Scheduled Jobs** pane displays the jobs currently scheduled by the system configuration. If you have permission, you can select one or more jobs in the list and run them immediately. The preceding warning about jobs and system resources applies here too. For more information on scheduled jobs, see: Configure the Scheduler.

# **Repository Browser**

# **Repository Browser overview**

The **Repository Browser** topic provides access to the integrated Subversion repository and provides some basic client functionality including browsing, downloading, and committing locally changed resources (you must have permissions for committing to the repository). The Repository Browser is available in all Polarion products and is present in Navigation in project scope and global (Repository) scope.



#### Browse the repository

By default you begin browsing in Polarion ALM's main internal repository, the root folder, and the Head revision.

The Path at the top of the page shows you your current location at all times. All folders in the Path are hyperlinks. Clicking on a folder name in the Path takes you directly into that folder.

At the right side of the Path bar are controls that enable you to select the Trunk, branches, or tags, select a revision to browse, and navigate up one folder in the Path.

#### **Drill-down into folders**

The table in the **Info** tab lists the content of the current repository folder, which appears as the last element in the Path. Content is either folders or files. Each content type is distinguished by a folder or file icon.

To drill down into a folder or file, click the folder name in the **Name** column.

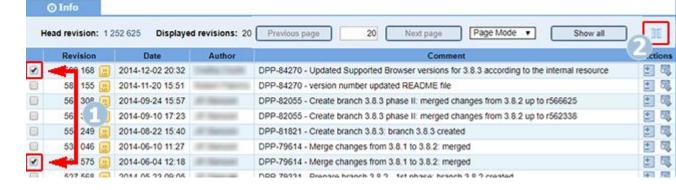
### Select a different revision

You can select a different revision in one of these ways:

- Click the Revision List icon in the Info bar, and select a revision in the list of revisions that appears
- Click Browse in the Path bar (right side of page), and in the Go to path screen, enter the number
  of the revision you want to browse.

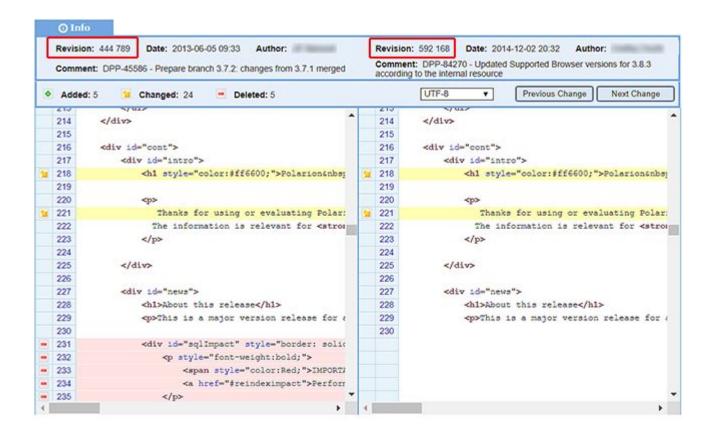
# **Compare revisions**

- 1. Browse into the file for which you want to compare revisions.
- On the Info bar, click Revision list.
- In the table of revisions, select the check box next to the revisions you want to compare. Select only one revision to compare it to the current HEAD revision. Select two revisions in the table to compare them with each other.
- On the far right side of the Info bar, click **Compare**.





The selected revisions of the file open in separate panes. A toolbar above them summarizes the number of additions, changes, and deletions. Two buttons on the toolbar enable you to jump quickly to the previous or next change.



# Repository operations

Using the Repository Browser you can add folders, directories, and files, commit a locally changed file, delete existing content, or download a file or a directory as a Zip file. Use the respective icons that appear in the **Info** tab.

To delete content, select the check box located at the far left of an element's row in the Info table. Select all elements you want to delete, and click **Delete selected elements**.

#### Tip

You can use almost any Subversion client to access the Polarion repository from outside the portal. You can access the entire repository or just certain projects. For this you need the URL of the portal repository and/or project folders within it. You must have access permission for the repository.

By default, the repository root URL is shown in the Project Group Overview in the Repository Home topic. The URL for a project is shown by default in the Project Overview section of project Home pages. (Home pages may be customized from the default and may either show the repository URL elsewhere, or not at all. Consult your portal or project administrator if necessary.)

# Download a local copy of a folder

1. Using links in **Info** and/or **Path**, navigate to the folder you want to download as a local copy.

- 2. In the Info tab, click Download current directory as a zip.
- 3. If necessary, move the downloaded Zip file from your default download location to some other location and unzip it.

It's important to note that the above downloads only the data files and no Subversion metadata and control files. Consequently, the result is *not* a true SVN working copy that can be used with an external client. (The Repository Browser is not intended as a full-featured version control client. If you need such functionality, you should obtain and use one of the many available Subversion client applications.)

## Download a local copy of a file

- 1. Navigate to the head revision of the file in the Repository Browser, and open it so that you are viewing the file content.
- 2. In the **Info** tab, click **Download**.
- 3. If necessary, move the downloaded file from your default download location to some other location.

#### Caution

The Repository Browser downloads only files. It does not download any Subversion metadata or control files. Consequently, the download is *not* a true SVN local working copy that can be used with an external client. The Repository Browser is not intended as a full-featured SVN client. If you need full functionality, you should obtain and use one of the many available Subversion client applications.

#### Commit a locally modified file

- 1. Navigate to the head revision of the file in the Repository Browser, and open it so that you are viewing the file content.
- 2. In the **Info** tab, click **Commit**.
- 3. On the **Commit File** page, click **Choose File**, and then select your local copy of the file.
- 4. Enter a commit message in the **Comment** field and click **OK**.
- 5. Click **OK** on the **Change confirmation** page to complete the commit operation.

#### Add a directory

The Repository Browser enables your to add new directories in the repository.

- 1. In the table of resources, drill down into the directory that will be the *parent* of the directory you will add.
- 2. In the **Info** tab, click **Add directory**.

- 3. Enter a name for the new directory. Enter a comment to replace the default comment, and then click **OK**.
- 4. In the **Change confirmation** panel, click **OK** to complete the operation.

# Add a file to the repository

- 1. Using links in **Info** and/or **Path**, navigate into the folder in which you want to add a new file.
- 2. In Info, click Add file.
- 3. Click **Choose File**, and in your system's file selection dialog box, navigate to and select the local file you wish to add.
- 4. (Optional) Enter a comment to replace the default comment, and then click **OK**.
- 5. In the **Change confirmation** panel, click **OK** to complete the operation.

#### **Note**

You can only add one file at a time using the Repository Browser. If you need to add many files, use an external SVN client that has features for adding multiple files at once.

### **Delete files or directories**

- 1. In the table of resources, drill down into the folder containing the file(s) you want to delete.
- 2. Select the check box on the row of the file(s) to be deleted.
- 3. In the **Info** tab, click **Delete selected elements**.
- 4. Add a comment and click **OK**.
- 5. In the **Change confirmation** panel, click **OK** to complete the operation.

You can delete directories in the same way. Navigate into the *parent* directory, select the check box next to the directory or directories to delete, and click **Delete selected elements**.

# **Working with Documents**

# **Working with Documents overview**

Documents combine the ease of use of desktop word processing applications with the power and scope of a data-driven tool. Your ability to work with Documents is controlled by permissions granted to your user role in projects. You may have different roles in different projects, and your access level may vary according to your role. For example, you may be able to create new Documents in one project, but not in another project. Should you find that some feature seems to be missing, or you cannot perform some operation, check your permissions with the project administrator. For information about what permissions you need to have for different actions, see Permissions and License, especially Permissions for Document Actions.

### Create a new Document

# ALM REQ QA

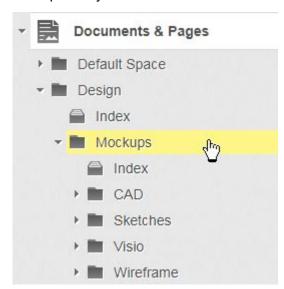
Creating a new Document in your Polarion portal is as simple as creating a new document in your favorite word processor application. Documents for a project are stored in one or more spaces under the **Documents and Pages** topic in **Navigation**. You can locate them easily either by browsing the Documents in Navigation, or using the Navigation pane's filter box, which provides incremental search of your project content.

You can create a new Document in Polarion in one of these ways:

- Create a new empty Document in the Polarion portal using the **New Document** feature.
- Create a new Document by importing an existing Microsoft Office Word™ document, using the **Word Import** feature.

# **Spaces and Subspaces**

You can create a hierarchy of nested folders ( Spaces and Subspaces) for new documents, and optionally save them as Shortcuts in Navigation.



Five levels of nested subspaces and up to 250 artifacts within each of them are officially supported.

# Tip

Plan and create your **Space** and **Subspace** hierarchy *before* creating documents. to avoid any move restrictions.

# Warning

Permissions change when creating a new Subspace within a Space. If you create a 
Subspace within a Space, or Move a Space/Subspace from another Space, they do not inherit the Subversion repository's restrictions, or permissions from their new parent. If you want to restrict access to a newly created Subspace, you need to work with an an administrator to remove the SVN access rights and create a custom rule for permissions (see Configure user Permissions: Define permissions for Artifact Sets).

When creating documents in Polarion for the first time, it's a good idea to plan out how you would like to organize them beforehand. Unlike Word documents, Live Documents and Live Reports are integrated with a vast array of possible widgets, potentially thousands of Work Items, and in the case of Live Documents, a powerful version history.

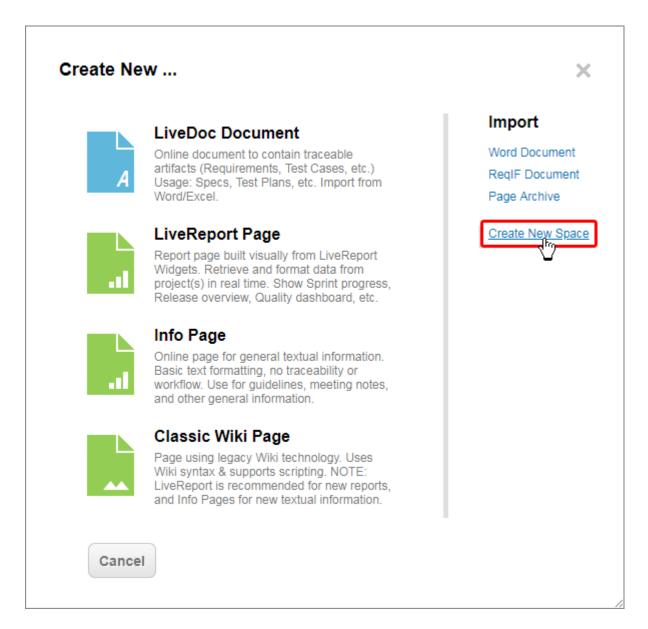
This diversity of integrated external content can lead to move restrictions, so its a good idea to organize your **Spaces** and **Subspaces** from the start, to avoid any potential move restrictions.

#### Note

You can move LiveDoc Documents once they're created, but you cannot move Live Report, Info or Classic Wiki pages.

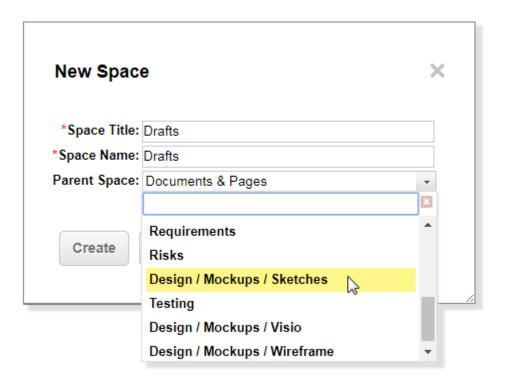
# Create a nested hierarchy of Spaces

- 1. In Navigation, click Documents and Pages.
- 2. Click on the top left.
  - If you clicked on an existing Space or Subspace, click Expand Tools at the top then and click Create New.
  - If you clicked on an existing, top level, Space, click Expand Tools at the top, then and click Create New.
  - If you clicked on the lander, click on the top left.
- 3. Click Create New Space.



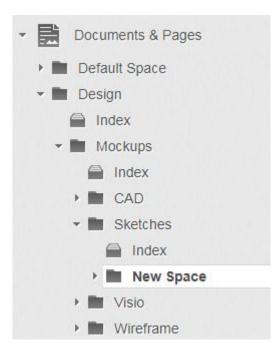
4. Enter a **Space Title**, a **Space Name** and select the **Parent Space** that you would like the new **Space** to appear in.

(If you've already created a **Subspace** within a **Space**, then the hierarchy appears within the list control.)

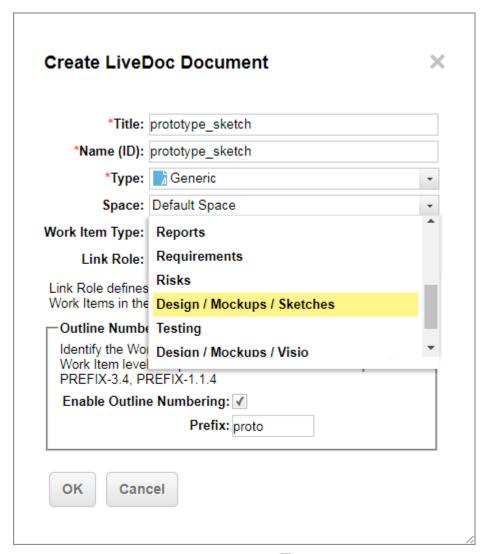


5. Click Create.

The new Space appears in the navigation bar's Space hierarchy.



6. Now when you create a new Live Document, Live Report, Info Page, or Classic Wiki Page, you can select any Space or Subspace in the Space list.



Live Documents can also be moved to a Space or Subspace.

#### Note

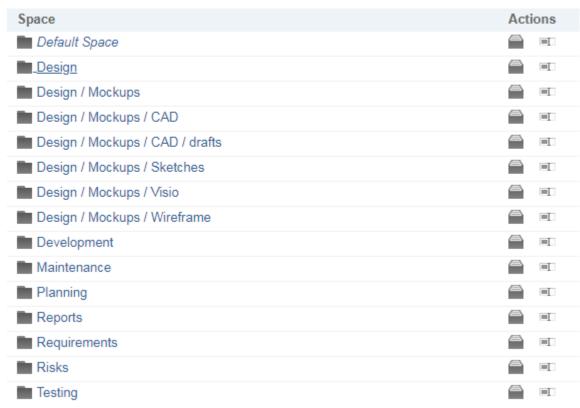
Hierarchical spaces created in the Polarion do not actually create physical nested folders within the repository.

## Rename a Space or Subspace

- 1. Click on **Documents and Pages** in Navigation.
- 2. All existing **Spaces** appear in a list.

## What's here

The nested\_test contains the following space(s):

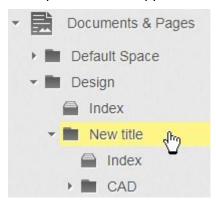


(You can view their hierarchy in **Navigation**.)

- Click <sup>■</sup> to the right of the **Space** you want to rename.
- 4. Enter a new Title.

(The **Space's ID** cannot be changed.)

- 5. Click Save.
- 6. The updated name appears in **Navigation**.



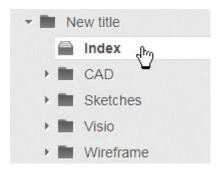
## **Delete a Space or Subspace**

## Warning

If you delete a **Space**, any Documents or **Subspaces** within it are also deleted, and cannot be restored.

## Delete a Space:

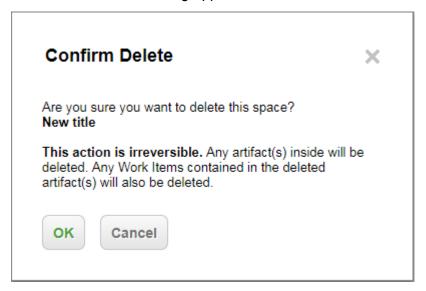
1. Click the **Index** of the **Space** you want to delete.



2. Click and select Delete (Space\_Name).



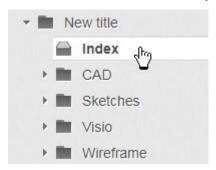
3. The **Confirm Delete** dialog appears.



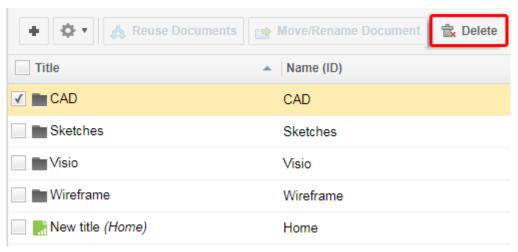
4. Click OK.

## Delete a Subspace:

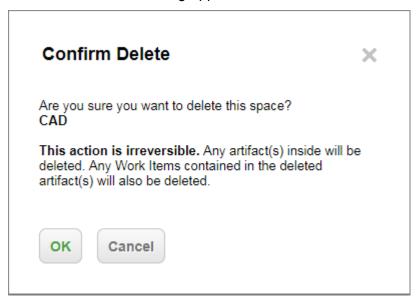
1. Click the Index of the Space that contains the Subspaces you want to delete.



2. Select the check box beside the **Subspace(s)** to delete.



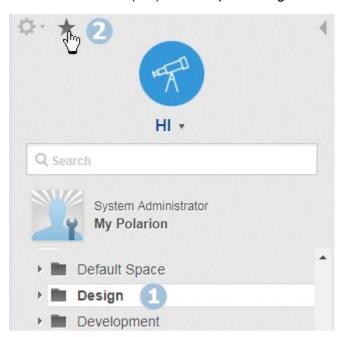
- 3. Click **Delete** on the right.
- 4. The Confirm Delete dialog appears.



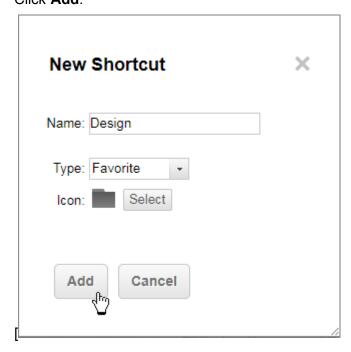
## 5. Click OK.

## Save a Space or Subspace as a Shortcut

- 1. Click the name of the Space or Subspace that you want to save as a Shortcut.
- 2. Click the star icon (\*) at the top of Navigation.



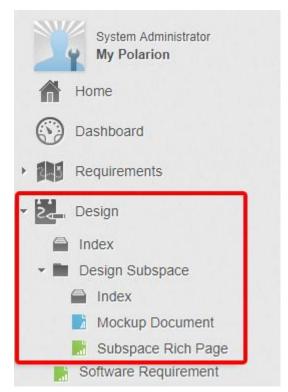
## 3. Click Add.



#### Tip

Optionally click **Select** to choose a different icon.

4. The Space or Subspace now appear in the favorites section of Navigation.



If it contains a hierarchy of Subspaces, all files and Spaces within the hierarchy are also accessible from the Shortcut.

#### Tip

- You can optionally modify the position of the Shortcut, and you can remove it at any time.
- Filtering also works in Space/Subspace hierarchies added as favorites.

## **Create a new empty Document**

Create a new empty Document when you want to create new content such as specifications, or any type of document that contains Work Items, and you do not already have the content in a Microsoft Word document from which to import it.

When you create a new Document, the Document Editor opens enabling you to write and format the content. The most current saved state of the Document is online at all times. You can share the Document's URL with other Polarion users, and other users who have the necessary permissions to access the project and the Document can collaborate on the content. You can also use the Round-trip

for Microsoft Word feature to share the Document with external stakeholders who do not have access to the Document in your Polarion portal.

Either during the authoring process, or when it is complete, you can use tools in the Document Editor to mark portions of the document content as Work Items and link the Document's Work Items for structure and traceability. For more information, see Work Items in Documents.

#### To create a new Document:

- 1. Open the project in which to create the new Document.
- 2. In Navigation, click Documents and Pages. The topic's home page opens.
- 3. In the toolbar at the top of the page, click Create New and choose LiveDoc Document. The Create LiveDoc Document dialog appears.
- 4. In the Create LiveDoc Document dialog, fill in the fields:
  - Title: Specify a human-readable Document title. Characters other than ASCII may be used.
  - Name (ID): Specify an identifier (referred to as ID) unique to the current project and space. Used by the system for indexing and other purposes. Use only ASCII characters in IDs.
  - Type: Choose from a list of available Document types. This field is present only if Document Types are configured for the project.
  - Space: Select the Space or Subspace where the new Document is located once it has been created.
  - **Work Item Type**: Select the main type of Work Item the new Document should contain. You can add more types after the Document is created. The list control contains the Work Item types defined in your project configuration. You must select one of the listed types only administrators can create new types.
  - Link Role: Select the default link role that should be applied to hierarchically structured Work Items in the new Document.
    - For example, if you split a paragraph marked as a requirement so you have two separate requirements, and you then indent the second paragraph, this option controls what relationship is applied between the indented requirement paragraph and the preceding requirement paragraph. Typically the indented paragraph is a child of the preceding, so the default selection has parent is what you want. Note that this is just the default: you can apply any configured link role later on when editing the Document.
  - If you want the Document's headings to have outline numbers, select the Enable Outline **Numbering** box and optionally enter a prefix value in the **Prefix** field.
- 5. Click **OK** to create your new Document. The new Document loads in the Document Editor ready for editing. Before beginning to work on the new Document, you may want to check the configuration that controls how the Document displays Work Items. Click the Work Item type icon in the Document Editor toolbar and choose Configure. For more information, see Multiple Work Item Types.

### Add remote links to a LiveDoc

Polarion supports the linking of remote data from other tools, like SIEMENS' Teamcenter.

Linked data can be used to:

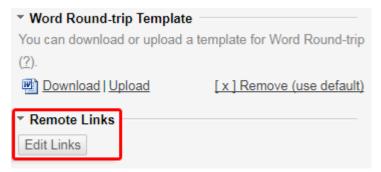
- Create links for Work Items or LiveDocs from objects that reside on external OSLC enabled tools.
   (Act as a Linked Data consumer.)
- Link Polarion LiveDocs to objects that reside on external OSLC enabled tools. (Act as a Linked Data provider.)

This is generally done by creating or selecting a remote object in a so-called delegated UI provided by the Linked Data provider.

Once Linked Data Friends are configured by an administrator, they appear as an loon in the Linked Work Items section of a Work Item.

#### Add an external Linked Data item

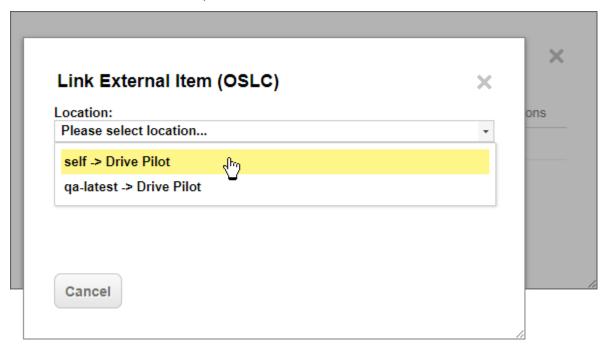
- 1. Select the target **□Project**.
- 2. Click **Documents & Pages** in the navigation bar.
- 3. Select an existing LiveDoc or create a new one.
- 4. Click then **Document Properties**.
- 5. Click **Edit Links** at the bottom of the sidebar.



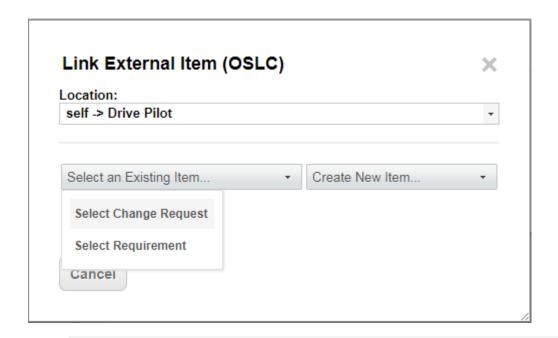
6. Select the link's role and click lake Link External Item (OSLC).



- 1 The external link's role.
  2 Select an external item.
- 7. Select a location from the drop-down list.



- 8. Click Allow.
- 9. Click **Select an Existing Item** and select an option.
  - Or **Create New Item** to create a new item in the external application.



#### Note

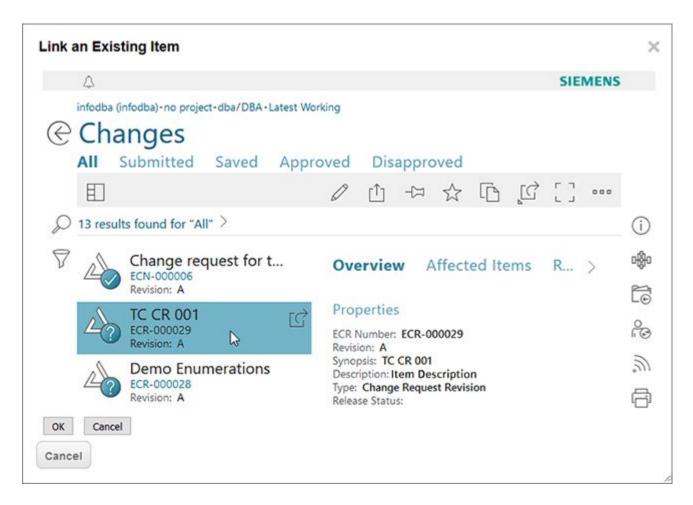
**Create New Item** launches the external OSLC application in a pop-up window.

The workflow to create a new Item varies depending on the application.

## 10. Example: Existing Item from Teamcenter:

Select Teamcenter in the Location list control, and click Select an Existing Item.

Teamcenter appears in a pop-up window.



- 1. Select an item from the list on the left. An overview of its details appears on the right.
- 2. Click OK.
- The item is linked to the Polarion LiveDoc.

## **Import Microsoft Word Documents**

## Import Microsoft Word documents overview

# ALM REQ QA

You can use the Import from Word (or Word Import) feature to convert existing specification content, such as requirements or test cases, written in a Word document into a Polarion LiveDoc Document. This allows your organization to take advantage of Polarion's process automation and management capabilities, while retaining the familiarity and ease of use of a simple desktop-based office document for authors.

By bringing the content of your Microsoft Word documents into Polarion, your authors give up very little in terms of how they write and format content, and your organization gains much in terms of efficient and controlled process, full-lifecycle traceability, transparency, and team collaboration and communication. Remember that after a Microsoft Word document is imported, you can always export

it back to its original format to work on it offline, or enable others to do so using the Round-trip for Microsoft Word feature.

Let's take a moment for a brief overview of this functionality. A typical, but by no means the only use case is to bring existing requirements specifications into Polarion. In so doing, you can have Polarion automatically differentiate between document content that is just general information, and content that should be recognized as requirement type Work Items which can be tracked, assigned, planned, managed with workflow, and linked for traceability across functional boundaries such as Quality Assurance and Development.

This automatic recognition can be based on keywords in the Microsoft Word document, or on styles defined and applied in the document such as bullet, heading, and paragraph styles, or some combination of both of these. Keywords can also serve as triggers for actions specified in import rules and applied to recognized Work Items. For example, the word **must** in your document might the trigger for an action to set a value in the Severity field to **must have** or any value defined in the project configuration for that field. You define Work Item recognition Rules using easy-to-understand graphical controls when you import a Microsoft Word document into Polarion as a LiveDoc Document.

You can save any set of rules you define as an Import Configuration which you can then quickly apply to other documents when you import them, so you only need to define Work Item recognition rules for similar documents once. You can of course have multiple Import Configurations for different types of documents. For example, you might have one set of rules in an Import Configuration which you apply only when importing software requirements specifications, and another when importing functional requirements specifications, and yet another for test case specifications.

## Important compatibility and other notes

Microsoft Word documents to be imported must be saved in Word 2007 or later format, with the .docx extension. The older .doc format is not supported.

Import file size is limited to 50 MB unless configured otherwise by administrators.

#### **Tip**

If the default **paragraph** style in a Microsoft Word document is customized, that will not affect its appearance when imported into Polarion, but the custom styling will still be applied when the document is exported back into Word for round-trip collaboration.

### Tip

While there is no hard limit to the number of Work Items a LiveDoc Document can have, the recommend best practice is to keep it under five thousand.

### **Launch Word import**

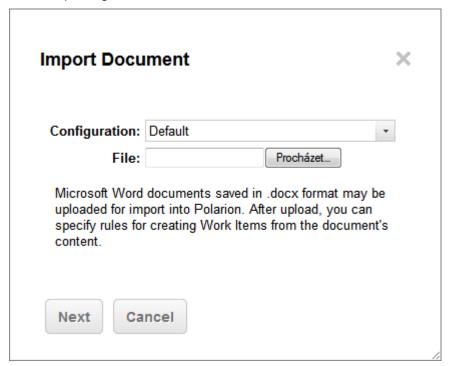
This section tells you how to begin importing a Microsoft Word document. Subsequent sections in the chapter explain how to define rules to automatically recognize Work Items, set severity level and/or other data values from document content, and then work on the new LiveDoc Document after importing.

Documents can only be imported into an existing project. If you do not yet have a project created for your Documents, you should read about planning new projects and creating new projects before

going further with Word import. You must have user permissions allowing you to create and save content in the project where you will import Microsoft Word documents.

To begin importing a document:

- 1. Open the project into which you want to import your document content.
- 2. In **Navigation**, click **Documents and Pages**. The topic's home page loads in your browser.
- 3. In the toolbar at the top of the page, click the **Create New** button.
- 4. In the **Import** section of the **Create New** dialog, click the clink labeled **Word Document**.
- 5. Choose an Import Configuration in the **Configuration** list. If no Import Configurations have been saved by you or other users, the only item in the list is **Default**. Use it for the current import. You will be able to define Work Item recognition rules and save them as a new Import Configuration later in the import process. Import configurations enable you to reuse Work Item recognition rules when importing other Microsoft Word documents.



If additional Import Configurations exist in the list, select any which you know to be applicable for documents of the kind you are importing. Additional Import Configurations appear in the list only if a user has created and shared them, or the project template from which the project was instantiated provides some additional import configuration.

- 6. Click the Choose File button and use your file system navigation to select the document file you wish to import. The file format must be .docx.
- 7. Click **Next** to upload the document and begin the import process.

When your document is uploaded, Polarion converts it to a LiveDoc Document and displays a Preview. Note that nothing is committed to the repository yet, and the import can be canceled using the Cancel Import link in the Preview.

The Preview contains a panel where you can optionally specify a name (ID) and title, specify the Document type, select the destination space, and define and save rules that control the recognition of portions of the content as Polarion-managed Work Items (see the screenshot in the next section.)

- 8. In the rules panel, optionally change the default Document name and title in the **Name** and **Title** fields, respectively, at the top of the panel.
- 9. If your project is configured to support Document types, you can optionally select a different Document type in the **Type** field. If no Document types are configured, the field is not present.
- 10. In the **Space** list, select the **Space** where you want the LiveDoc Document to reside.

If you don't want to define automatic Work Item recognition rules at this time, you can remove any existing rules in the Preview, and check the preview of the LiveDoc Document to make sure that nothing is marked as a Work Item, and then click the **Import** button to finish the import process and create the new Document. You can then work on this document and mark Work Items manually while editing the Polarion document.

If you do want to have content recognized and marked as Work Items on import, you now need to define Work Item import rules so that Polarion will automatically create Work Items in your new LiveDoc Document when you complete the import process.

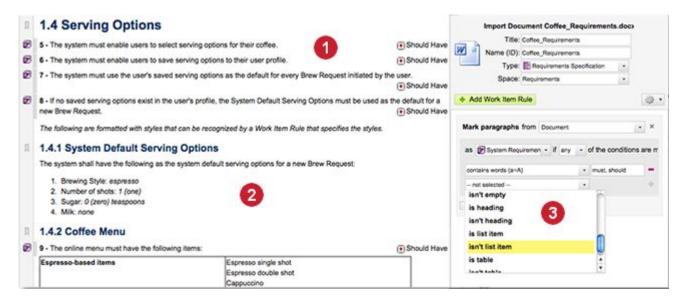
#### **Note**

If the project is configured to support **Document workflow**, then after the import operation, the **Status** field of the new LiveDoc Document will be set to the configured initial **Status** value, and any configured initial workflow action will be invoked.

For example, if the initial configured Status value is **Draft**, the new Document will have that status, even if the source Document was not a draft, but a finished specification.

#### **Configure automatic Work Item recognition**

Polarion can create Work Items based on content elements in a Microsoft Word document according to Work Item import Rules you specify during the Preview stage of the import process. This topic assumes that you have successfully initiated import of a Microsoft Word document and are now seeing the Preview:





Your goal is to have Polarion recognize that some content in the imported document is just general text, while other content should be recognized as Work Items of some particular type — requirements or test cases, for example. While the Document is in Preview mode, you can experiment with the Work Item recognition Rules without actually creating any Work Items in the repository.

#### **Advance Preparation**

Before you begin importing Microsoft Word documents, there are several things to set up and to know about.

- Target project: You should already have a project set up in Polarion, and you should know what Work Item types are defined in the project's configuration.
- Document types: You should know whether your project configuration defines different types of Documents. If it does, you will be able to choose the type into which your content is imported. If not so configured, consider whether you your project should be configured for different Document types before importing your content.
- Work Item types: You should decide which Work Item type or types you want to create in Polarion when importing a Microsoft Word document.
- Content to Work Item mapping: you should review your Microsoft Word document and identify what elements should become Work Items or Work Item attributes such as description after the document is imported. This is key successful Work Items recognition from your document's content. The next section discusses how to prepare a document for import to Polarion.

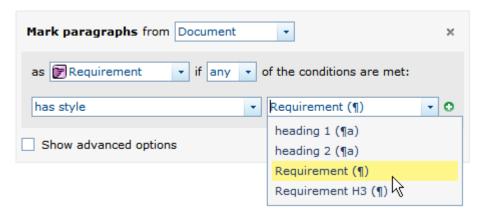
#### **Prepare the Document**

Carefully go through the Microsoft Word document you plan to import and make sure you know what content should be managed as Work Items in Polarion after import.

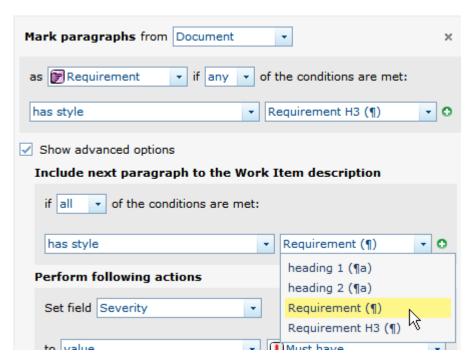
Consider a requirements specification document as an example. What elements of your document should Polarion recognize and mark as Requirement type Work Items? Should it recognize some words? Should it recognize some paragraph or list item styles, or entire sections under headings? Will you need to define just one simple Work Item Rule to achieve the recognition you want, or will you need a more complex Rule, or multiple Rules? While it's possible devise quite complex strategies that get very granular, you can actually achieve a great deal with fairly simple Rules.

For example, Polarion's default import configuration recognizes the words **must** and **should**, it will mark paragraphs containing those words as Requirement type Work Items, and it sets the Severity level to **must have** or **should have** respectively. The project configuration must have the Requirement Work Item type and the severity levels defined. This configuration is predefined in Polarion's project templates that support requirements management. Therefore, if you create a project based on such a template, and edit your Microsoft Word document so that only the content of Requirements contains the words **must** and **should**, you can very likely achieve quite a usable result when you import the document and use the default Work Item Rule.

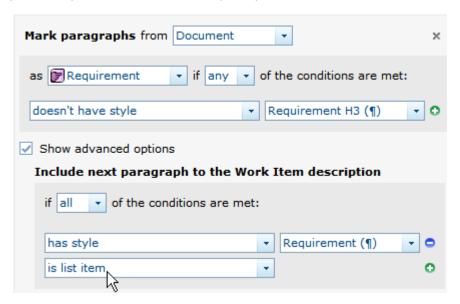
Another easy to implement mapping strategy is to make sure all requirements in your Microsoft Word document have a particular style. For example, you might define paragraph style named **Requirement** and apply it to all paragraphs that should be recognized as Requirement type Work Items. When you subsequently import the document, create a Work Item Rule that specifies that elements in the document with that style applied to them should be recognized as Requirements.



Another mapping approach that is still fairly easy is to define a heading style named something like **Requirement H3** in addition to the paragraph style named **Requirement**. On import, you can create a relatively simple Rule with multiple conditions that says to mark content styled **Requirement 3** as a Work Item, and include in the description any following paragraphs styled with the **Requirement** style.



Suppose that you have a bullet list after a paragraph styled as **Requirement**, and you also want to include the list in the Work Item description. To do that, add another condition to the Rule that says to include paragraphs styled as list items which occur after a **Requirement 3** styled heading. This gives you a very powerful, but still fairly easy-to-define Work Item Rule.



#### Tip

Remember that in the import Preview, you can experiment with Rules and see what their effect will be before you actually create any Work Items in the repository. The Rules pane contains the **Preview** button in the lower left corner. Clicking that button loads a preview of the Document with the current Work Item Rules applied as they are currently defined. You can continue experimenting with Work Item Rules until you are satisfied with the result you see in Preview.

Remember also that you can save Work Item Rules as Import Configurations, which can be used when importing other Microsoft Word documents with similar characteristics.

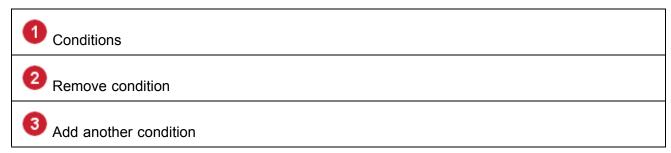
#### Define a simple Work Item Rule

A simple Work Item Rule is one using only basic options and specifying one or more conditions. A Rule's components are like an instruction sentence. For example:0

Mark paragraphs from [whole document or specific section] as a [Work Item of this type] if [any/all] of the conditions are met.

— after which come one or more conditions: [contains words] [words], [has style] [style name], [contains text matching regex] [regular expression], to name a few examples of conditions.





Let us see how to define a simple Work Item Rule with two conditions that will cause paragraphs anywhere in the imported document to be marked as a Requirement type Work Item if they contain any of the word shown, or if they are a list item.

Take the following actions:

- 1. In the Mark paragraphs from field, select the content element or elements of the original document to which Polarion should apply Rules to recognize Work Items. Select **Document** to allow Polarion to apply Work Item recognition Rules to all elements.
- 2. In the list following the word **as**, select the type of Work Item to be recognized in the content source. The list contains the Work Item types defined in the project configuration.
- 3. On the first conditions line, select a condition from the list of conditions. If your selection requires more input to complete the condition, an appropriate text, list, or other field will appear on the line. Fill in the additional information.
- 4. If you need another condition in this Rule, select it in the list on the second conditions line and as before, fill in any additional information.
- 5. If you need more conditions, add new condition lines by clicking the + icon. Fill in the data as needed. If you make a mistake, you can remove any condition line from the Rule by clicking the icon at the end of the line.

If you need one or more additional Rules to achieve the Work Item recognition you want, you can click the Add Work Item Rule button near the top of the Rules pane. A new simple Rule is added each time you click. Repeat the above steps for each simple Rule you add.

### **Define advanced option Rules**

Each simple Rule you define can have Advanced options added to it to make it even more robust and selective. Advanced options are disabled by default. To enable these options for any Rule, select the **Show advanced options** check box. The Rule panel expands to show the advanced options.

Advanced options fall into the following categories which appear as labeled sections in the Rules panel:

- Paragraph inclusion conditions: Specify conditions for including paragraphs that follow a content element marked as a Work Item per the conditions defined in the simple Rule. For example, suppose the simple Rule specifies that elements styled as Heading 3 are to be recognized as Work Items. You would probably want the paragraphs that follow the heading included in the Work Item description, so you can specify one or more conditions under Include next paragraph to the Work Item description. You might specify paragraphs that have some style, and/or that are list items to include after the heading.
- Actions: Define actions which set the value of Work Item fields when document content is recognized as a Work Item according to the Rule. Each Rule's Advanced options contain one action by default. If you don't want it, you can remove it by clicking the X icon in the upper right corner of the action's panel. If you want more actions, you can click the Add button at the bottom of the Actions section.
  - Each Action applies to a selected Work Item field of recognized Work Items. Select the field in the **Set field** list. Depending on the field selected, a control appears which enables you to set the value of the selected field. The **execute** field enables you to specify whether the specified value will be set always, or conditionally. If conditionally, fields appear enabling you to define one or more conditions in a manner similar to conditions for other Rule parameters.
- Additional options: Opt to have the first paragraph of any block of content the Rule marks as a Work Item imported as the title of the recognized Work Item. The first paragraph of such a block

might be a heading. You can also specify which fields, if any, you want to appear at the beginning, and at the end of recognized Work Items in the LiveDoc Document. You must specify the field ID in **Fields at start** and **Fields at end**. If you want the values of more than one field to appear at the start or end of recognized Work Items, specify multiple IDs separated by a comma.

## Import Work Items from tables

Work Items can even be extracted from tables within a Microsoft Word document. You can target all the tables within a document or select only the sections, even subsections to include.

#### Tip

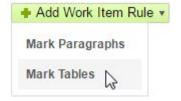
Import Rules for content contained in tables may require the use of regular expressions (regex). Some information and examples are available in the reference topic Import Rules for Tables.

Multiple Rules can be created to extract different Work Item types such as Requirements or Tasks. Severity or priority levels can also be automatically extracted.

In the example below, requirement keywords within a table are detected and assigned different Work Item severity levels using the advanced table Rule options.

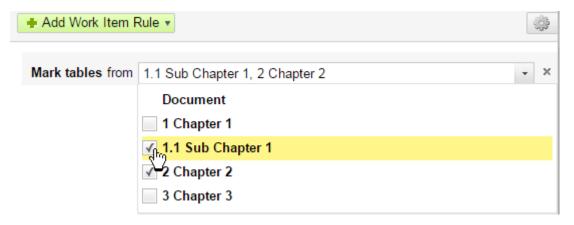
To import Work Items from within tables:

- 1. (Optional) To extract Work Items exclusively from tables within the document, close (\*) the section Mark paragraph from.
- 2. Click + Add Work Item Rule and select Mark Tables.

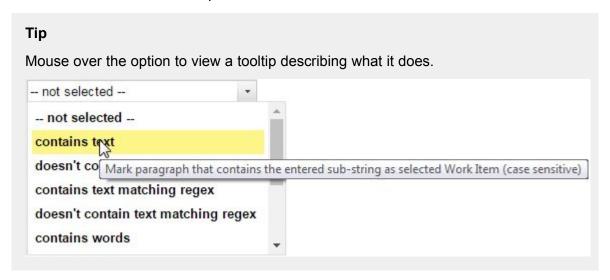


3. Define the scope of the tables to include in the **Mark tables from** drop-down menu.

(Clicking on Document will apply the Rule to the entire document.)



- 4. Select whether to invoke the Rule if **any** or **all** the conditions are met.
- 5. Select whether to apply to Rule to **Whole Tables** or only the **Header Rows**.
- 6. Under **Import Rows**, select a Work Item type in the **as** list. Types listed vary depending on the project configuration.
- 7. Select the Rule's behavior in the drop-down list.



8. Enter the condition details in the field that appears on the right.

If contains text is selected like the example above, enter the text that the Rule should search for.



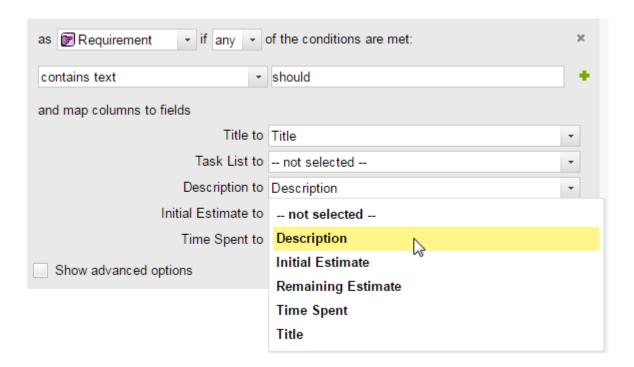
Click to add additional conditions.

#### Tip

To assign different Work Item attributes to each condition — Severity, for example rather than adding it here, create an entirely new Rule by clicking \*Add after the initial Rule is defined. See optional step below.

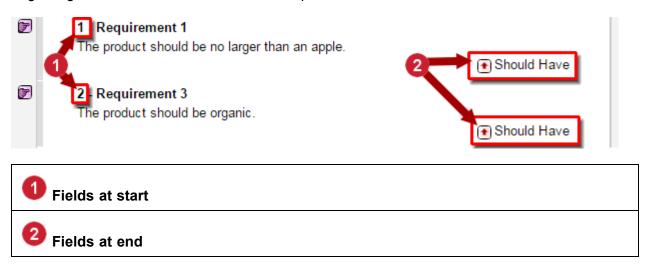
10. Polarion will automatically detect the target area's columns and try to match the text to existing Work Item definitions.

Any Column headers that cannot be recognized can be assigned manually or excluded by leaving them as **not selected**.



11. (Optional) Select the **Show Advanced Options** check box.

The **Fields at start** and **Fields at end** can be customized to change what attributes appear at the beginning and end of the work item in the imported document.



12. (Optional) Click - Add then select an option in the Set Field to enter additional Work Item attributes like Severity or Priority.

The available options can be customized by administrators.)

13. (Optional) The available options under the **To** field change depending on what was selected in **Set Field**.

In this example, **Severity** is selected to convert table cells that contain the word **should**, to requirements with a severity level of **Should Have**.



- 14. (Optional) Specify when the Rule should be implemented in the **Execute** list. The default is **Always**.
- 15. Click **Preview** to see how the Rule affects the document.

### Tip

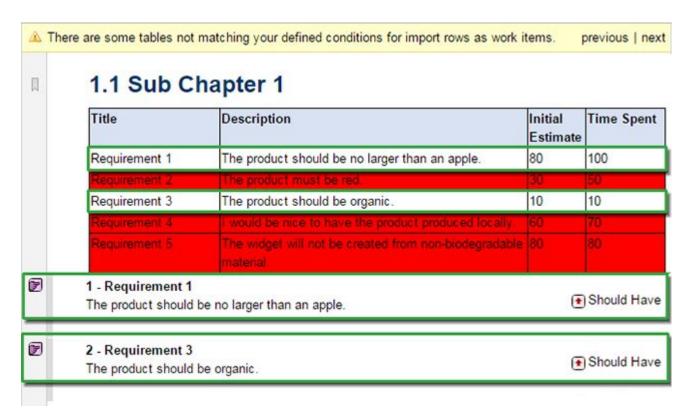
It is important to understand that the document has not been fully imported yet, and nothing has been written to the repository Clicking Preview displays what the Rule will mark as Work Items when you finally click Import.

This way, you can thoroughly test Rules and Rule combinations without creating, and later having to delete, unwanted Work Items.

Any table rows that do not have a Work Item Rule applied to them are highlighted in red.

Table rows that contain data that the Rule(s) recognize are converted to work items that appear below the table.

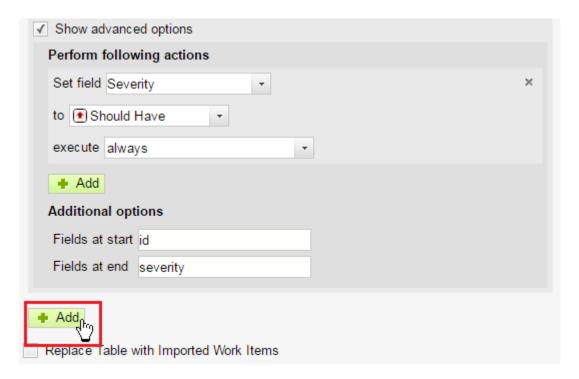
Detected table data looks like this:



16. Check the **Replace Table with Imported Items** box to **replace** the table(s) with the extracted Work Items.

If left unchecked, the created Work Items will appear directly below the table in which they were detected.

17. (Optional) Click \*Add above Replace Table with Imported Items to add another table Rule.





18. Click **Import** when you are satisfied from the preview that all the information you want shown as Work Items is correctly shown.

#### Save Work Item Rules for reuse

Once you have defined a Work Item Rule or a set of Rules and run Preview to verify that they cause Work Items to be marked correctly, you can save your Rules as an Import Configuration that can be reused when importing similar documents in the future, thereby eliminating the need to specify the same Work Item Rules again.

- 1. In the Rules pane, click and choose **Save**. The **Save Import Configuration** dialog appears.
- 2. If you are saving Rules to a new configuration, select Create new configuration and enter a name in the Name field.
- 3. If you want to save your current Rule definitions to overwrite an existing configuration, select Update existing configuration, and select which configuration you want to update from the list of existing configurations.
- 4. Click **Save** to save the new or updated configuration. If you saved a new configuration, it now appears in the list of Import Configurations, which users can select when importing Microsoft Word documents.

## Use a saved import configuration

If you or a colleague saved a set of Work Item Rules as an Import Configuration, you can reuse it to import a document in some instances. The content of the document you are importing must conform to the rules defined in the Import Configuration so that Work Item content is correctly recognized.

- 1. With the imported Microsoft Word document in Preview mode, click ( ) and choose **Load**.
- 2. In the dialog, select the Import Configuration whose Work Item recognition Rules you want to reuse for the current import operation.
- 3. Optionally, click the **Preview** button and check the document preview to verify that the document's Work Items are marked as expected.
- 4. Click the **Import** button to import the Microsoft Word document reusing the selected Import Configuration.

See Manage saved import configurations for more information.

## Set the severity from Document content

Having the Work Item recognition process set the severity level of recognized Work Items automatically is commonly-needed, and is just one example of the actions that can be invoked and applied to recognized Work Items during Work Item recognition. You can specify this in the Actions section of the Advanced options of a Work Item Rule. There, you can select the Severity field, and specify the conditions for setting a value for the field when a Work Item is recognized according to the rule. For more information, see Define Advanced Option Rules.

#### **Table of contents**

The import feature also imports Table of Contents from the source document. Keep the following points in mind:

- Outline numbering is turned on by default, and the TOC of the resulting Document will contain outline numbers in the TOC.
- If the source document contains a table of contents, a table of contents will be generated in the
  resulting Document during import. A manual TOC is replaced by an automatically generated and
  updated TOC in the Document. Up to five heading levels are recognized and rendered in the
  TOC of the Document. Properties of the original TOC are saved and used in any subsequent
  Word Round-trip export.
- If the source document contains a TOC, the import preview shows how it will look in the
  Document after import. The TOC preview is updated when the import preview is updated. TOC
  items in the import preview are not clickable.
- Headings that are recognized as Work Items do not appear in the Document TOC, in both import
  preview and the LiveDoc Document. If the source document contained a TOC but all headings
  are recognized as Work Items, then the LiveDoc Document's TOC is initially empty.

#### **Documents with OLE objects**

Polarion can display OLE Object thumbnails during import. However, some additional third-party image converter software must be installed and configured by an administrator before you can import such documents. Information for administrators is provided in the topic Configure OLE Object Support.

#### **Unsupported content**

If the source document contains images or other content in formats not supported by the import from Microsoft Word and image conversion features, the document is imported into Polarion without the unsupported content. If your project administrator configures the wordImport.ignore.[CLASS] system property for the content types in your Microsoft Word documents, a placeholder image and/or text informing you of the omission is inserted into the Polarion LiveDoc Document at the same point where the original, unsupported content was in the original document.

See Advanced System Tuning wordImport.ignore.[CLASS] for more information.

## RegIF Import/Export

## RegIF Import/Export overview

# ALM REQ

Wikipedia defines Requirements Interchange Format (RegIF) as follows:

"RegIF is an XML schema that is used to exchange requirements, along with its associated metadata, between software tools from different vendors. The requirements exchange format also defines a workflow for transmitting the status of requirements between partners. Although developed in the automotive industry, ReqIF is suitable for lossless exchange of requirements in any industry."

In Polarion, you can import RegIF files into Polarion LiveDoc Documents, and export LiveDocs to a RegIF file using the RegIF Round-trip feature. This feature can be used to exchange requirements between different Polarion servers as well as between external tools and systems.

#### Supported versions and file types

Polarion currently supports RegIF versions 1.0.1 and 1.1, and RIF versions 1.1a and 1.2. The following file types are supported:

- .regif or .xml: Plain RegIF/RIF file without embedded objects.
- .regifz or .zip: Archive that contains one or multiple RegIF or RIF file(s) with embedded objects, and the embedded objects.

#### Note

Image Magick must be installed and configured to see the thumbnails of imported OLE Objects. See the section titled Include thumbnails for OLE DOORS and RegIF imports in the Image Magick documentation for details.

## Tip

Embedded image or attachment previews are preserved during the ReqIF Import/Export process.

## Warning

Tables are not supported by ReqIF.

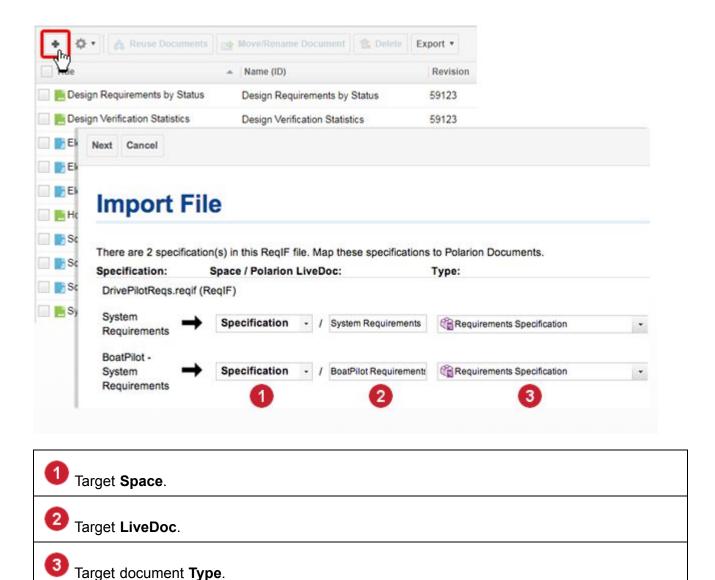
Mathematical formulas are not supported for RegIF import/export.

#### **Import RegIF files**

You can import ReqIF/RIF files singly, or several at once. ReqIF/RIF files may contain one or multiple Specifications. If a file contains multiple specifications, you map them to different Polarion Documents. When importing a ReqIF file for the first time, a Polarion Document is created for each specification in the ReqIF file. Subsequent imports of the same ReqIF file into the same Space automatically map to the respective Polarion Document, the content of which is updated and synchronized with the content of the ReqIF file.

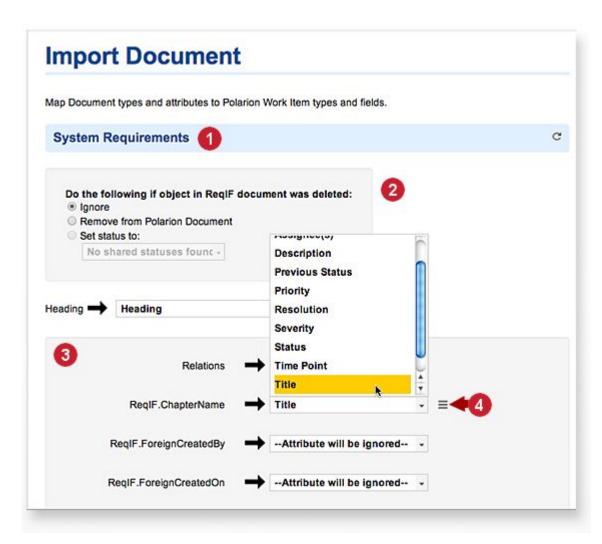
- 1. Open the project that should host the imported requirements. Be sure you have the necessary permissions to create new Documents.
- 2. In Navigation, expand **Documents and Pages** and expand the node of the space into which you want to import, (or you can create a new space), and select **Index**.
- 3. In the Index page toolbar, click **●** (Create New).
- 4. In the dialog's **Import** section, click the link labeled **ReqIF Document**.
- 5. In the next dialog, select the interchange file stored on a local or network drive and click **Start**.
- 6. Wait for the initial import to finish. If the file is large the process can take a while.
- 7. When the **Import Document** page appears, select the specification(s) you want to import. Map all selected specifications in the ReqIF file to a Space (the current Space is used by default), and a LiveDoc Document. See the figure below.
  - If you import an updated version of a ReqIF document you previously imported, the target LiveDoc value will be prefilled and can't be changed.
- 8. When your specification to Document mapping is complete, click the **Next** button.

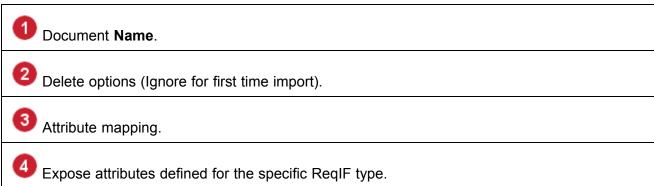
The first part of the RegIF import operation is mapping specification(s) to Polarion Document(s):



#### Map Attributes, Types and Fields

The second part of the ReqIF import operation is mapping specification attributes to Polarion Work Item fields and types. Once the initial import phase is complete, the **Import Document** page is updated and presents options for mapping ReqIF document types and attributes to Polarion Work Item types and fields, plus delete options for each specification.





The **Import Document** page contains a separate section for each Document to be created by the importer. Each section is demarcated by a header, which displays the Document name. The first subsection is delete options. These control what happens if an object has been removed from the ReqIF specification since the last Polarion import. You can ignore this section when importing a specification for the first time.

A ReqIF specification may contain different object types. For each of these object types you must configure the Polarion Work Item type to be created. Because some requirements tools don't support

different types but implicitly define object without text as headings (e.g. IBM DOORS), the "virtual" types "Heading" and "Requirement" will be shown if there is only one object type in the specification.

Below the Object Type is a list of all attributes defined for the specific RegIF type. You should map each of those attributes to an attribute of the selected Polarion Work Item type. Each attribute has a control that exposes value mappings (see figure above). You can define value mappings for all kinds of attributes, but it is mandatory to map all available values of an enumeration attribute. (You will not be able to start import until this is done.)

When all mapping is complete, click the **Start Import** to execute the import of the configured specifications. When the import process is finished, an overview page opens, which contains a table that lists and provides access to the newly created Polarion Documents.

#### **Export Documents to RegIF**

You can export Polarion Documents to ReqIF files of any supported type. You can export a single Document or multiple Documents to a single RegIF file. Each exported Document corresponds to a specification in the exported RegIF file. As with import, some configuration and mapping is necessary during the export process.

There are two ways to launch export to RegIF:

- 1. From the Index page of any space. Export from here will always be to a new RegIF file you cannot export to update an existing, previously imported RegIF file this way. You can export multiple Documents to a single ReqIF file.
- 2. From the Actions menu of a Document in the Document Editor. You can only export the open Document. However, you will have several options for the RegIF output:
  - Update existing ReqIF specification: Write changes into an existing specification in an existing RegIF file.
  - Add as new specification to an existing ReqIF file: Create a new specification in an existing RegIF file.

#### **Note**

DOORS (and some other requirements tools) have the limitation that links can only exist among items in the same RegIF file. Thus, if you want to import links created in Polarion to such a tool, you should first export the LiveDoc that contains the Work Items that are target of the links and then export the LiveDoc that contains the linking items to the generated RegIF file.

**Create new RegIF file**: Export to a new RegIF or RIF 1.2 file.

To export from a space's Index page:

- 1. Navigate to the Index page of the space that contains the Polarion Document(s) you want to export.
- 2. In the table, check the row(s) of the Document(s) to be exported.
- 3. On the page toolbar, click the **Export** button and choose **Export to RegIF File**.

- 4. Map types and attributes as described in **Mapping Types and Attributes**, below.
- 5. Click the Start Export button to launch the exporter process.

To export a Document in the Document Editor:

- 1. In **Navigation**, expand **Documents and Pages**, browse to the Document you want to export, and double-click to open it. If there are many Documents, you can use the search field at the top of the Navigation panel.
- 2. With the Document open in the Document Editor, click the Actions Menu button and choose RegIF Round-trip & Export Document.
- 3. Select the desired option for the ReqIF file output and click **Start**.
- 4. Map types and attributes as described in Mapping Types and Attributes, below.
- 5. Click the **Start Export** button to launch the exporter process.

### **Map Types and Attributes**

When presented with the **Export Polarion LiveDoc Document** page, you must configure export options and mappings in order to complete the export process. The first section for each ReqIF specification presents several options:

- Export new Work Items and update document structure means that new Work Items will be added to the ReqIF file, and Work Items will be moved within the file (change parent, change position). Usually you do that only if you "own" the reqIF file; that is, the file was initially exported by you.
- Update ReqIF metadata means that attributes and available values for enumeration attributes
  will be added according to the mapping configuration. You definitely should do that only if you
  own the ReqIF file.
- DOORS compatibility mode means there will be some additional processing to make the ReqIF file compatible with DOORS. You should use this option only if the exported ReqIF file will be directly imported into DOORS without using any additional tools (e.g. Requisis Rex, Atego Excerpt or Agosense). Currently this setting:
  - Instructs Polarion to wrap images as OLE objects (increasing the size by orders of magnitude)
  - o For RIF: Adds a special DOORS-dependent section to the exported file.
  - For RIF: Controls naming of default attributes (Object Heading + Object Text vs. Title + Description)

Mapping basically is the same as for import except that some default attributes will be added automatically if you are creating a new ReqIF file. For ReqIF, the defaults are added according to the suggestion of the ReqIF Implementers Forum. For RIF, the defaults are Title and Description or Object Heading and Object Text when running DOORS Compatibility mode.

When **Update Metadata** is enabled, you can add any supported Polarion attribute to the ReqIF file. The configuration of the ReqIF fill will be adjusted accordingly.

When all mapping is complete, starting export will trigger download of the generated RegIF file. Give this some time, as some browsers may not show generation and download progress. You will see an message when the process is finished, and any errors during export will be reported.

## Tip

An alternative path to reach the mapping and export options page is via space **Index** page **Export** Export to ReqIF File. That option will export all selected Polarion Documents into a new RegIF file, and the configuration page will include an interface to configure the file name and type.

#### Reuse mapping configuration

When importing or exporting from/to RegIF, you can save the mappings you define as templates. You can then reuse the mappings for subsequent imports/exports. Each of the section headers on the mappings page has a drop-down actions menu, which provides actions for saving the current mapping as a template, or loading a previously saved template. Mapping configuration templates can be created in both global and project scopes.

## Import File

Map Document types and attributes to Polarion Work Item types and fields. System Requirements Reload Polarion meta-data. Do the following if object in ReqIF file was deleted: Load template Ignore Save as template Remove from Polarion Document Set status to: No shared statuses founc -

## **Browse for Documents**

Heading -

# ALM REQ QA PRO REV

--Select Work Item type--

To locate Documents, you can browse in the **Navigation** panel, or in the **Index** page of every Space.

To browse in **Navigation**, expand **Documents and Pages** to see the Spaces. Expand any Space's node, scroll down to locate the Document you want to access, and click on its Navigation node to load the Document in the Document Editor. If the Space contains Subspaces, you can expand them to see their contents. If there are too many Documents, Navigation cannot display them all and you will need to Search to locate the Document(s) you want.

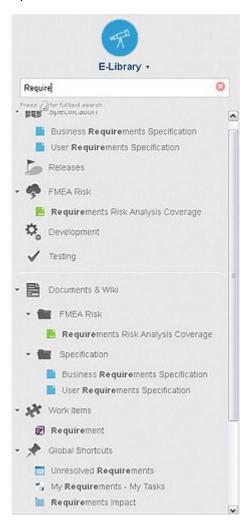
You can also browse for Documents in the Index page of any Space. Expand a Space node in Navigation, look for **Index** and click it. The Space's Index page lists all the Documents and Pages in the Space in a table format. You can sort the table on any of the columns by clicking the desired column headers.

## **Search for Documents**

# ALM REQ QA PRO REV

If there are too many Documents and Pages for all of them to be listed in Navigation, a message appears directing you to use Search. In this case, you need to know at least some part of the Document name, or title (if different from name), or some phrase in its content.

The Navigation panel has a Search text box near the top. As you type, the Navigation panel is filtered and shows only content in each topic that matches the text you enter. You can enter the first letters of any word in the Document name, and look for the result in the automatically expanded **Documents and Pages** topic. If you are working in a project, then only the project is searched. When you identify the Document you want to access, click on its node in the Navigation panel to open it in the Document Editor.

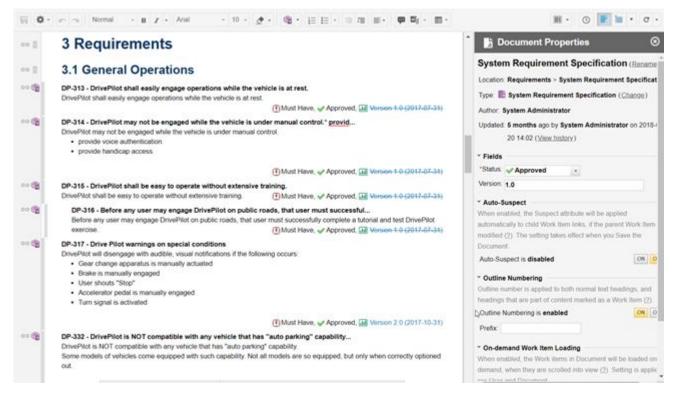


You can expand the search to the entire portal by pressing Enter/Return in the Search box. On the Search Results page, you can filter the results to show only Documents by clicking the Documents link in the page header.

# The Document Editor

# ALM REQ QA PRO

After creating a Document, either in the portal or by importing, you can edit it in the Document Editor. You will see much that is familiar to you from desktop office applications, or if you have used an online tool like Google Docs. The major difference, conceptually, is that some of your Documents' content can be marked as Work Items.



This approach provides the benefits of both office documents, and data-driven process and project management for the organization. People like business analysts and requirement engineers whose work typically focuses around documents don't have to change their paradigm or give up much in terms of functionality and ease of use. People on the technical side get the workflow and data-driven management capabilities they need. Executives and others responsible for compliance issues get the information they need. The organization as a whole benefits from improved efficiency, transparency, and communication.

This section focuses on various features and capabilities you will encounter while editing Documents using the Document Editor. See also Document Editor Toolbar.

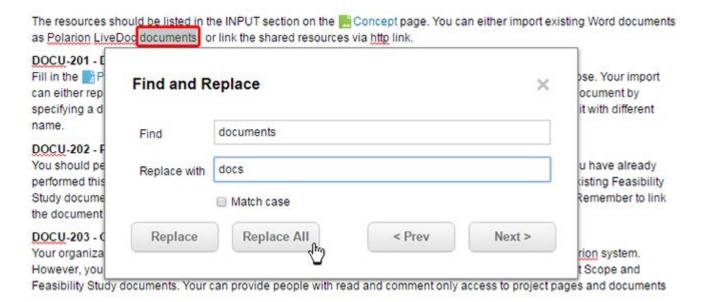
# Find and Replace

The Document Editor has its own Q Find and Q Find and Replace that even work for text within the Work Items contained in a Document.

They behave a little differently if On-demand Work Item Loading is set to **ON** or **OFF** in the **Document Properties** sidebar of the Document you're working with.

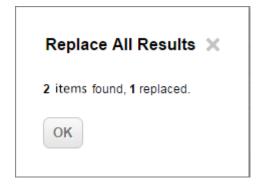
# With on-demand Work Item Loading OFF

1. Click Q Find and Replace in the Actions Menu or use the Ctrl + H keyboard shortcut.



(Clicking **Ctrl+ F** with On-demand Work Item Loading **OFF** will launch the Browser's native *Find* dialog.)

- 2. Enter an item to search for in the **Find** field.
- 3. (Optional). Select the **Match case** check box ( Match case ) to only display results with the same capitalization.
- 4. Cycle through the results by clicking **Next>** and **<Previous**.
- 5. Click **Replace** to replace the highlighted item with the text in the **Replace with** field or **Replace All** to replace all occurrences within the document.
- 6. If **Replace All** is clicked another dialog appears showing the results.



# **Note**

If there is a discrepancy between the number of items found and replaced, it is because non-editable items, such as referenced Work Items, or TOC entries, cannot be replaced.

# With On-demand Work Item Loading ON

Because On-demand Work Item Loading only loads data for that a user has scrolled into view, using your browser's native Find (Ctrl + F) generally leads to incomplete search results. To ensure that all data in a Document and its Work Items are searched when On-demand Work Item Loading is turned on, Polarion has its own Find function. You can still use Ctrl + F, or Q Find icon in the Q **Actions** menu to launch it

#### Caution

If you use Ctrl + F, click on the Document first. If the focus is on the Navigation panel, sidebar, or browser window, the browser's internal Find command will be launched instead, and the search results are likely to be incomplete.

#### Find:

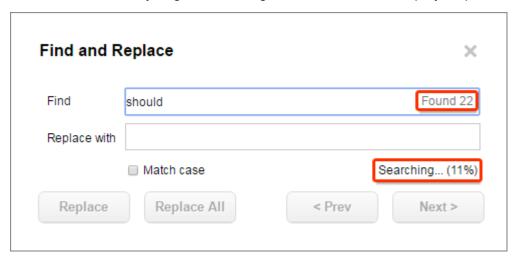
1. Click the Q Find in the Actions Menu or Press Ctrl + F.

DOCU-201 - Document the Project Scope. Fill in the Project Scope document provided, or you can import an existing Word docur Find × Find docu 6 of 79 Match case < Prev Next >

- 2. Polarion automatically begins searching the document, and displays a percentage of its progress. When complete, the total number of matching results is displayed and the current occurrence highlighted.
- 3. Cycle through the results by clicking **Next>** and **<Previous**.
- 4. (Optional). Select the Match case check box ( Match case ) to only display results with the same capitalization.
- 5. Ctrl + H turns an open Find dialog into Find and Replace.

# Replace & Replace All

- 1. Click the Q Find and Replace icon in the Actions Menu or the Ctrl + Hkeyboard shortcut.
- 2. Enter the text to replace in the **Find** field.
- 3. Polarion automatically begins searching the document, and displays a percentage of its progress.



4. Once complete, it lists the total number of times the item appears and the occurrence number that's currently highlighted.

MYPR-203 - Garner management buy-in and necessary approvals for the project



- 5. (Optional). Select the **Match case** check box ( Match case ) to only display results with the same capitalization.
- 6. Cycle through the results by clicking **Next>** and **<Previous**.

Click **Replace** to only replace the currently highlighted item, or **Replace All** to replace all matching occurrences.

# **Internet Explorer limitations:**

- With Internet Explorer, **Find and Replace** does not automatically jump to the first occurrence. Just click **Next>** to do so.
- Text that cannot be edited is counted and tracked, but not highlighted.

# **Copy and Paste**

Because Documents are artifact-aware, that is, they contain Work Items which are tracked and managed with other tools, the common actions of cutting/copying and pasting of text require some special considerations and processing behind the scenes. You should keep the following points in mind when performing any type of cut-copy-paste actions in Documents.

In general, cut/copy and paste between different Documents (also between different projects) will just copy the rich text to the destination, except when copying Work Items.

- Headings are preserved, but new Heading objects are created.
- If copy and pasting from an existing Work Item, the target document will the option to select a Referenced Work Item or a New Work Item appears.
- If copy and pasting from an *Unsaved Work Item*, the target document will **contain a new Work** Item with only the text selected copied.
- Tables and their content are preserved
- Table of Contents included in cut/copy is removed on paste. The TOC will need to be recreated in the target Document.
- Wiki content inserted in the Document is copied and pasted without alteration. If the Wiki content source contains relative paths, these could become invalid in the target Document and you would need to modify them manually.
- Links are copied/pasted without alteration. If links contain relative paths, the paths might not be valid in the target Document and you would need to update them manually.
- If cut/copied text contains comment markers, these are dropped on paste to the target Document.

# Paste Images from the clipboard

With some limitations (see Note below) you can paste an image from your clipboard into the Document body or into the descriptions of Work Items in the Document. If there is a mixture of images and text in the clipboard, then only the name of the image file is pasted, and not the image itself. When you past an image, an attachment is automatically created with file name screenshot-[timestamp].png, where [timestamp] is the date-time when the image was pasted. Supported image formats are: BMP, JPG, PNG, and GIF.

To paste an image from your clipboard, simply locate the insert point where you want to image to appear in the Document and use your browser's *Paste* keyboard shortcut or menu command.

# **Note**

Quality of the pasted image can vary depending on whether you copy the image from your file system, a web browser, a photo viewer, or image management program, as well as which supported browser you are using.

# Clipboard operations within a LiveDoc Document

While working on a Document you may decide to change the structure and flow of the Work Items. You might want to cut an entire Work Item and paste it elsewhere in the Document, or copy a Work Item, past it as the basis for a new item, modifying the text after pasting. To facilitate these actions the Document Editor provides **Copy** and **Cut** commands on the Work Item icon menu which appears to the left of the Work Item title, just outside the page border.

The **Cut** command selects the entire Work Item and cuts it to the clipboard, visually removing it from the Document. You can use Undo to restore the cut item to the Document.

The **Copy** command selects the entire Work Item and copies it to the clipboard, leaving the Work Item content highlighted in the Document.

After copying or cutting a Work Item using the respective menu command, move the insertion point to the line where you want to paste the item and use your browser's Paste command or keyboard shortcut to paste from the clipboard.

#### Tip

You can always use your pointing device to select text or a Work Item, and use your browser's keyboard shortcuts to cut, copy, and paste the selection. The menu items described above merely make it easy to accurately copy or cut an entire Work Item.

# Spell check

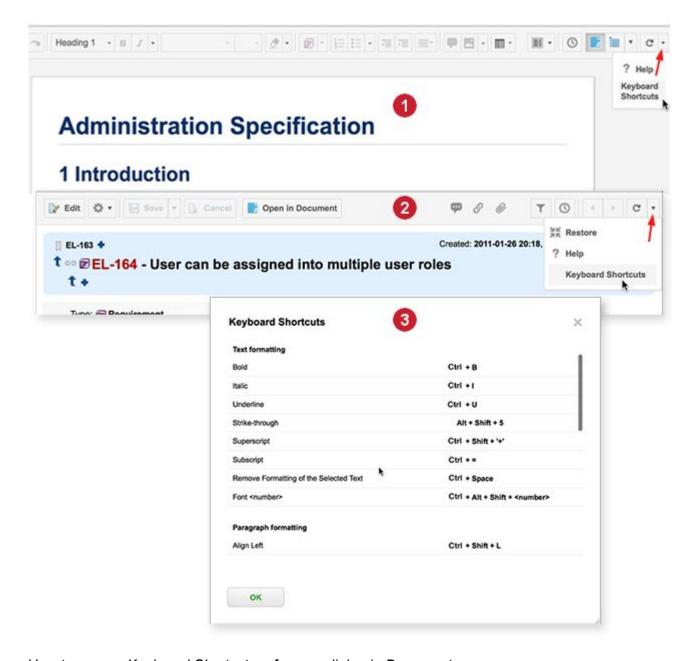
Polarion LiveDoc Documents use your browser's built-in spell checking feature to check spelling in text and Work Items. Be sure you are familiar with this feature of your web browser and that it is enabled.

# **Spell check and Performance**

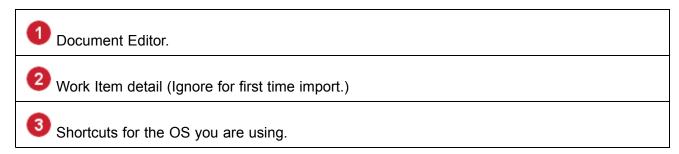
A known issue in very large Documents is that browser-based spell checking may negatively impact performance of the Document Editor or result in memory-usage errors. This has been observed most often with Microsoft Internet Explorer and Firefox. If you encounter such problems with a Document, you may need to disable your browser's spell check feature while editing it.

# **Keyboard shortcuts**

Keyboard shortcuts in the Document Editor are similar to those of applications such as Microsoft Word and Google Docs. The same shortcuts apply in the Description field and other rich-text type fields in Work Items. A comprehensive reference dialog, context-sensitive for your operating system, is available via the **Keyboard Shortcuts** item on the Help menu of the respective editors.



How to access Keyboard Shortcuts reference dialog in Documents



### **Work Items in Documents**

#### **Work Items in Documents**

Any time during the authoring process of a Document, you can mark some parts of its content as Work Items. Marking content creates a Work Item artifact in the underlying repository, enabling the team to take advantage of all the traceability, workflow, reporting, and other management features. Document authors can continue to work with content in a straightforward document-like editor.

# Tip

While there is no hard limit to the number of Work Items you can have in a LiveDoc, the recommended best practice is that you keep it under 5000.

By default, Documents can contain Work Items of the type named **Heading**, and one other type specified by the user who creates the Document. Work Items created in Documents are tracked through their lifecycle and managed with workflow. They are also accessible in the Work Items topic for those who prefer a more tool-based interface.

After a Document is created, one or more additional user-defined Work Items types can be configured, enabling users to mark Document content as a Work Item of any of the configured types. For example, a team might choose to have requirements and test cases in the same Document. See Configure Document Work Item types for specifics.

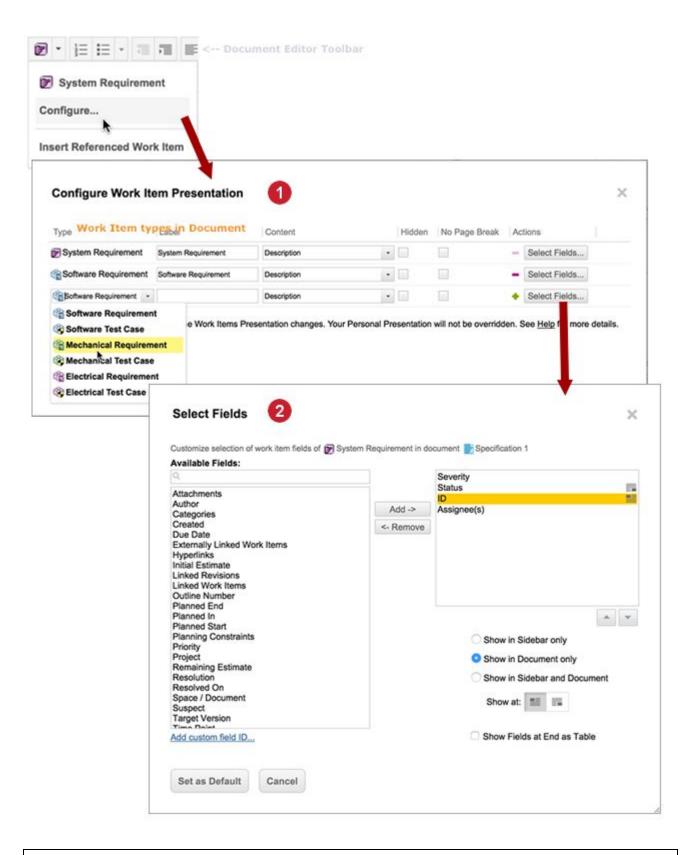
To make changes in a Document, you must have permissions to modify Documents, to create new Work Items, and to modify Work Item fields. If you are denied permission to modify any fields, some restrictions on creating new Work Items are imposed. For details, see Work Item Field Permission Restrictions: New Work Items

# **Configure Document Work Item types**

By default, Documents contain only one Work Item type — Requirement, for example. You specify the type when creating a new Document. In many cases, one Work Item type per Document is enough and you need do nothing more after you create a Document. However, Documents can contain more than one Work Item type. For example, you might want a Document to contain Work Items of a type named Requirement and Work Items of a type named Test Case. If Requirement was the type specified when the Document was created, then you need to add the second type, Test Case, to the Document configuration before you can mark any content as a Test Case.

- 1. In the Document Editor toolbar, click the Work Item Type drop-down list control in the Document editor toolbar (see figure below).
- 2. On the drop-down menu, choose **Configure Presentation**. The **Configure Work Item Presentation** dialog box appears (see figure below). The dialog box presents a table showing the Work Item types currently supported in the Document.
- 3. On the last row of the table, click the **Type** list and select the Work Item type you want to be able to define in this Document.
- 4. Fill in the fields in the remaining columns as desired (consult: Configure Work Item Presentation Dialog if necessary).

- 5. If you wish to add support for another type, click the → icon in the **Actions** column. A row is added to the table.
- 6. Repeat these steps until you have added support for all the Work Item types you want in the current Document. Click **OK** when finished.



Add a Work Item Type



Configure fields to show in **Work Item** properties

See also Configure Work Item Presentation Dialog, Display Data Field Values and Change Work Item Type.

# Referenced Work Item types

You can insert Work Items contained in some other Document into a Document you are currently working on. These referenced Work Items may be contained in another Document in the same project, or in a different project. Once inserted in the current Document, they can be read but not modified. It is possible that referenced Work Items might be of a type not currently specified in the current Document's configuration. The type might not even exist in the current project's configuration. Polarion handles these scenarios in different ways, which you should be aware of.

- If a referenced Work Item is from same project as the Document you are working on, but thatDocument is not configured for the referenced type, then Polarion automatically adds the type of the referenced Work Item to the current Document's configuration. By default, this added type is marked as **Hidden** in the **Work Item Presentation** dialog box. When so marked, the type does not appear in the Mark/unmark text menu and users cannot mark Document content as the newly added type. If you want users to be able to do that, then you can deselect the **Hidden** option in the Work Item Presentation dialog box.
- If a referenced Work Item is from different project, and its type is not defined the configuration of the current project, then Polarion automatically adds the type to the current Document's configuration, the added type is marked as **Hidden** in the **Work Item Presentation** dialog box, and it cannot be unmarked unless the same type is added to the current project's configuration by an administrator. The type icon from other project is not shown.

For example, suppose there is a requirements project that has a requirements specification Document that is configured to contain Work Items of a type named Requirement. Suppose also another project that has a test specification Document configured to contain Work Items of a type named Test Case. Now you want to insert some Work Items of the Test Case type, by reference from the test specification Document, into the requirements specification Document. However, the requirements specification Document' is configured to contain only one Work Item type — Requirement. Furthermore, the Test Case type is not even defined the requirements project configuration. When a referenced Work Item is inserted via the Mark/unmark text menu. Polarion automatically updates the Document configuration of the requirements specification with the referenced Work Item type — Test Case in this instance — allowing items of that type to appear in the Document. The properties of the added type (Label, Content, Fields, etc.) can be edited in the **Document Presentation** dialog box.

#### Mark Document content as Work Items

- 1. Select all the text you want included in the Work Item. If you select multiple paragraphs, they will all be included in the same Work Item.
- 2. On the Document toolbar, click the icon representing the type of Work Item initially configured for the Document.
- 3. In the menu that now appears, select the desired Work Item type. If only one type is configured, only that type appears. The selected text is marked as a Work Item, and the icon for the specified

type appears just outside the left-hand border of the editor. The artifact in the underlying repository is not created until you save the Document.

If you change your mind, you can unmark the content by clicking on its type icon and choosing **Unmark** from the pop-up menu. If you want to unmark the item and delete both Document content and the underlying artifact, choose **Delete** instead.

# **Modify Work Items**

After a Work Item is created, you can edit its description, and other attributes such as fields and links provided that your user profile is granted permission to MODIFY Work Items, and Work Item fields. If you are denied permission to modify any fields, you cannot edit Work Item content in the Document Editor. You must use the Table view of the Work Items topic, where you will be able to modify only those fields you have permission to modify. If you find you are restricted from editing Work Items, contact your administrator to review your permissions.

# Tip

To open a Work Item in the Table view, select it in the Document Editor, click the icon at the left edge, and choose **Open** on the menu. You can also open the Table view form the Document Editor toolbar.

#### Structure a Document's Work Items

Creating hierarchically structured Work Items is as easy as using different indent levels. For example suppose you have two paragraphs marked as Requirements, and the second paragraph is a subrequirement of the first. Simply indent the second paragraph. Behind the scenes, Polarion automatically takes care of linking the two Work Item artifacts with the correct link role, and people viewing them in the integrated Tracker will see the links and the relationship.

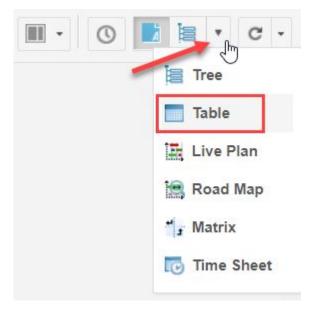
If you change your mind and decide the two paragraphs are not hierarchical, simply unindent the child paragraph and save the Document. Again, all linking operations of the underlying artifacts are handled automatically. You simply keep working with your document.

Suppose you have two paragraphs marked as Requirements and you decide that the second Requirement is neither a subrequirement, nor a separate requirement, but rather part of the first. Simply merge the two paragraphs just as you would with paragraphs or bullet-list items in Word. The content of the second Requirement is automatically unmarked as a Work Item, any structural linking is removed, and the content appears as part of the first Requirement. If the Document was saved after the second Requirement was created, then after the merge the underlying Work Item artifact is still maintained in the underlying repository and appears in the Recycle Bin pane of the Document sidebar. This enables maximum flexibility during the Document's development process. For example, you may later decide that a merged Requirement actually belongs in a different Document. You could delete the relevant paragraph from the combined Work Item, and move the original Work Item (now in the Document's Recycle Bin) to the other Document. When the Document is near completion, you can clean up any Work Items in the Recycle Bin that you are sure you no longer need. For information, see Recycle Bin.

You can also move content marked as a Work Item within the document via Cut/Paste. It works just like moving content in Word. Simply select the entire text of the marked Work Item, cut it to your clipboard, and paste it back into the Document wherever you want it.

# Modify structure using the Tracker interface

You can also modify the structure of the items using the Table view of the Work Items topic, which you can open from the Document Editor toolbar.



Using that interface, you can:

- Move an existing Work Item so that:
  - It is the first or last child of another Work Item in the same Document.
  - It is placed before or after another Work Item in the same Document, under the same parent item.
- Create a new child Work Item as the last child of the current Work Item in the same Document.
- Create a sibling Work Item on the same hierarchical level as the current Work Item in the same Document.

To move an existing Work Item:

- 1. Select it in the Table view.
- 3. In the Move Work Item(s) dialog, use the **In Document Structure** option to specify where you want to move the current Work Item.

#### **Note**

The move operation moves any existing subtree, not just the selected item(s).

Both contained and referenced Work Items can be moved within a Document using the Table interface.

See also Move Work Items.

To create a new sibling Work Item:

- 1. Select an item in the Table view that is on the same hierarchical level as the new item you want to create.
- 2. On the Work Item Editor toolbar invoke 🗫 🖚 🖁 Create Sibling Work Item
- 3. On the submenu, select the type for the new item. The menu enables you to select only the type(s) configured for the Document that contains the currently selected Work Item. For example, if the Document is configured to contain only Requirements, that will be the only submenu item. If it is configured to contain Requirements and Tasks, both will appear as submenu items.

#### **Link a Document's Work Items**

You can link the Work Items defined in a Document to other items in the same Document, in a different Document, or stored directly in the Tracker. Links can denote structure and provide traceability. For example, in a requirements specification Document, Work Items of the Requirement type may be structured with parent-child relationships where one item is the parent of one or more others. This type of structure is most easily and naturally done with hierarchical headings and/or indentation — of paragraphs or list items, for example. For information, see Structure a Document's Work Items.

Linking for traceability may require linking Work Items defined in one Document to Work Items defined in a different Document, or created directly in the integrated Tracker. For example, you would probably want to link Work Items of the Requirement type, defined in a requirements specification Document, to items of the Test Case type that verify the requirements. The Test Case items might be defined in another section of the same Document, in another Document, or created directly in the Tracker. For more complete information, see Link Work Items.

The following sections explain how to link Work Items to each other in the same Document, how to link Work Items in one Document to items in a different Document, and how to link Work Items in a Document to Work Items created directly in the Tracker.

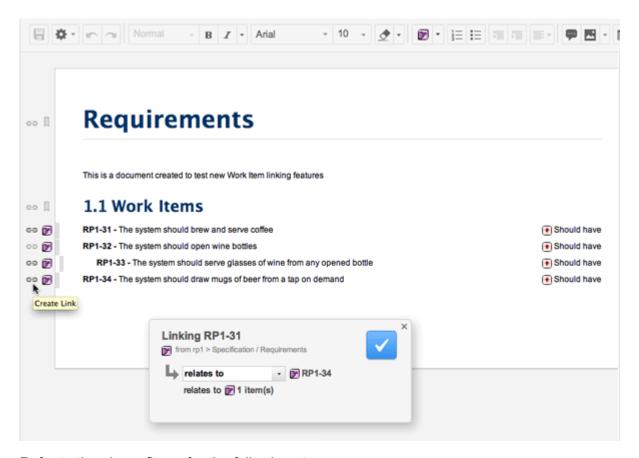
# Tip

No links are created in the system until you save the Document in which you originate the link(s). If you accidentally create a pending link to the wrong target item, you can click the **X** icon in the link detail panel and then confirm the prompt to cancel changes. You can then begin the linking operation again.

When linking to multiple Work Items, you can specify a different link role for different target items. For example, you might link from a Requirement to a Test Case with a link role named **verifies**, and to another requirement with one named **refines**. You select the link role in the link detail panel after clicking the link icon on each target Work Item.

# Link within a single Document

You can link one Work Item in a Document to another single item, or several target Work Items in the same Document all at once. For this operation, you only need one browser tab.



Refer to the above figure for the following steps.

- 1. Click on the icon in the left margin of the Work Item you want to link to some other Work Item. The link detail panel appears.
- 2. In the link detail panel, select the link role for the link you are creating. This describes the relationship of the source item to the target item(s) — relates to or depends on, for example.
- 3. Locate the first Work Item you are linking to, and click the Link icon to the left of it.
  - If you want to link to multiple Work Items, hold down Ctrl (Windows) or Command (Mac and others), and click the <sup>eq.</sup> (Create Link) icon of each target Work Item. The panel reports the items linked each time you click.
- 4. Save the Document. The links are not created in the repository or logged to the Document history until you save.

You can view the items to which any Work Item is linked by clicking on the item in the Document and opening the Work Item Properties pane of the Document sidebar.

#### **Link across different Documents**

You can link a Work Item in one Document to another single item in a different Document, or to several target Work Items in a different Document all at once.

# Tip

This linking is easiest if you open 2 browser tabs or windows: one displaying the Document containing the Work Item(s) you want to link (referred to as Tab 1), and the other (Tab 2) displaying the Document or Tracker view containing the Work Item(s) you want to link to.

To link Work Items in different Documents:

- 1. In one browser tab or window (Tab 1), open the Document containing the Work Item(s) you want to link.
- 2. In a second browser tab or window (Tab 2), open the Document containing the target Work Item(s).

That is, the item(s) you want to link to from some item(s) contained in the first Document (in Tab 1).

- 3. In Tab 1, select a Work Item to link by clicking on the (Create Link) icon in the item's left margin. The link detail panel appears in both browser instances.
- 4. In the link detail panel, select the link role for the link you are creating. This describes the relationship of the source item to the target item(s)— **Relates to** or **Depends on**, for example.
- 5. Switch to Tab 2. If you want to link the item in Tab 1 to multiple items in Tab 2, press and hold **Ctrl** or **Command** (depending on your operating system).
- 6. Click the (Create Link) icon beside the first target Work Item in Tab 2. To link to one or more additional items in this target Document, click their link icons while continuing to hold down the **Ctrl** or **Command** key.
- 7. Switch back to Tab 1 and save that Document (the one from which you started the link operation) to create the links in the repository and Document history.

# Tip

You can link new Work Items in a target Document as you create them. When you begin a link from a Work Item in one Document (in Tab 1), then if you switch to a different Document (in Tab 2), and you create new Work Items there, links to the new items are automatically created and added in the link detail panel.

# Tip

You can link a Work Item to one or more items in a historical revision of a target Document. For example, you might link the current revision of a Business Case item to the version of a Requirement in a functional specification Document that has been saved in a Baseline. To do this, open in Tab 2 the Baseline containing the revision of the Document, and then open the Document containing the items you want to link to.

#### Link from Document to Tracker

You can link a Work Item contained in a Document to one or more Work Items created in the Tracker that are not contained in any Document. You can link to a single target item, or multiple target items at once. As in the previous section, you should use 2 browser tabs.

- 1. In one browser tab (Tab 1), open the Document containing the Work Item(s) you want to link.
- 2. In a second browser tab (Tab 2), open the **Table** view (**Navigation** Work Items), and run a guery to retrieve the target Work Item(s) and list them in the table.
- 3. In Tab 1, select a Work Item to link by clicking on the (Create Link) icon in the item's left margin. The link detail panel appears in both browser instances.
- 4. In the link detail panel, select the link role for the link you are creating. This describes the relationship of the source item to the target item(s)—relates to or depends on, for example.
- 5. Switch to Tab 2. If you want to link the item in Tab 1 to multiple items in Tab 2, press and hold Ctrl or **Command** (depending on your operating system).
- 6. In the table in Tab 2, locate the first target Work Item and click the (Create Link) icon on its row in the table. To link to one or more additional items in the table, click the link icon on their respective table rows while continuing to hold down the Ctrl or Command key.

#### Note

Each target item can be linked with a different link role. Change the role in the link detail panel after **Ctrl** / **Command** clicking an item.

7. Switch back to Tab 1 and save the Document to create the links in the repository and Document history.

### Tip

You can link Work Items using other tools in the Polarion portal. You can use the Linked Work Items section in the Work Item editor (Navigation - Work Items -Table view). If your license provides the feature, you can also link Work Items in the \*Matrix view.

#### See also:

- Link Work Items
- Link a resource
- Link individual Work Items
- Link multiple Work Items at once
- Link Work Items in the Matrix view

Create links in Multi Edit view

# Manage existing links

Use the **Edit Links** button in the **Properties** page of the Document sidebar to launch the **Linked Work Items** dialog box. There, you can manage the links of the currently selected Work Item in your Document. You can use the dialog box as an alternate way to add one or more links to a selected Work Item, as well as to edit some properties of existing links, or remove some existing links. For details about this dialog, please see **Linked Work Items Dialog** 

# **Access Work Item properties**

The Document Sidebar provides the Work Item Properties sidebar, which enables you to edit data fields and access links of Work Items defined in the Document.

You can specify which Work Item fields appear in the Properties section of the panel. When the Properties panel is showing, click the cicon (Pane Settings) icon and choose **Select Fields** on the menu. In the **Select Fields** dialog box, select the check boxes of the Work Item fields you want to be able to edit in the Document, and deselect those of fields that should not be shown there. You can also control the order in which the fields display in the panel by rearranging their order in the dialog box.

Some Work Item fields, both default and custom, may be configured as required fields. A value must be supplied in all required fields before a Work Item can be saved. When you mark a new Work Item in a Document, if any required fields are not among those shown in the Work Item Properties panel, then a default value is automatically and silently set for these fields. This default may or may not be what you actually want, so it is usually a good idea to make sure that all required fields appear in Work Item Properties panel of the Sidebar. You can easily see which fields are required by looking at any Work Item in the Table view of the Document. Fields marked with a red asterisk (s) are required fields. Be sure to check the **Custom Fields** section to see if any custom fields have been configured as required fields.

# Tip

All the fields of any Work Item can be edited in the Table view of the Document, which you can launch from the Document Editor toolbar. Remember also that you need user permissions that enable you to edit Work Items and Work Item fields in order to change Work Item properties in a Document.

- 1. To access properties of a Work Item in a Document, click the Work Item's icon next to the left-hand margin of the Document Editor.
- 2. On the menu that appears, choose **Properties**.

The **Work Item Properties** panel of the sidebar displays properties of the currently selected Work Item in the Document. The data fields of the selected Work Item appear in the **Properties** section. Changes to these fields do not take effect until you save the Document.

The **Links** section appears below the **Properties** section. It shows the items currently linked with the selected Work Item.

### **Locating referenced Work Items**

Not all Work Items appearing in a Document are necessarily contained in it. Some may be contained in another Document and just referenced in the current one. If the currently selected Work Item is referenced from another Document, the Work Item Properties panel of the Sidebar contains the **Documents** section. That section lists the source Document — the one that actually contains the item, and in which it can be edited — plus all Documents that reference the Work Item. Names of Documents are clickable links that open the respective Documents in the Document Editor. For more information on referenced Work Items, see Branch Documents and Work Item Properties sidebar.

# Tip

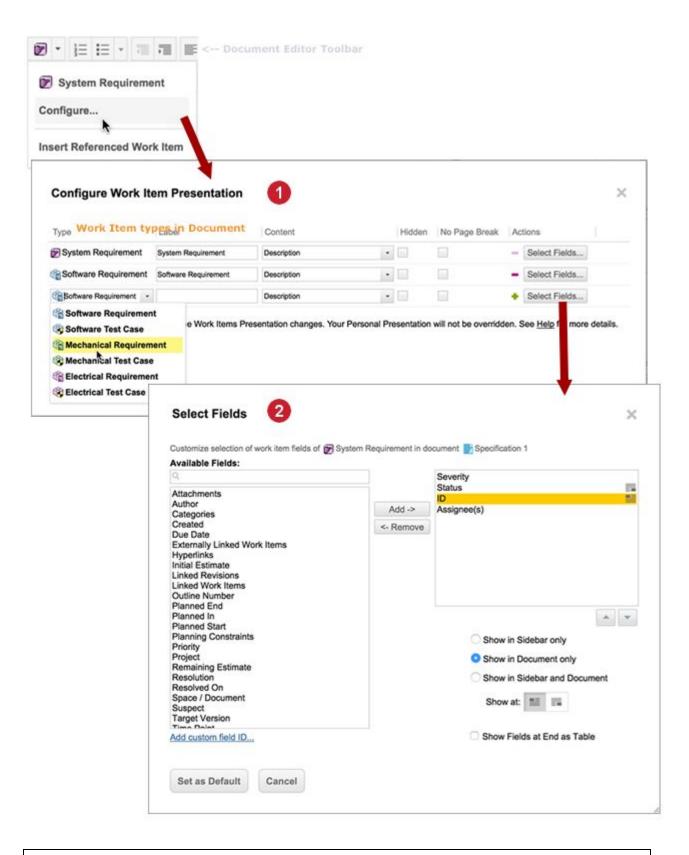
If you find you cannot edit a Work Item, it may be referenced from a different Document. Check the Work Item Properties panel. If the item is referenced, you can only modify it in the source Document, and you must have the necessary permissions for editing Work Items and fields in that Document. If Work Item not a referenced one, and you cannot edit it, you most likely lack the necessary permissions for the current Document.

### **Display Data Field values**

Whenever you mark Document content as a Work Item, Polarion creates a Work Item artifact in the repository. Each Work Item artifact has a number of default data fields. It is also possible to configure custom data fields. You can optionally show the value one or more Work Item data fields along with the text of Document elements marked as Work Items. For example, you might decide to show such data as status, priority, or assignee. When you configure data fields for the Work Item types in your Document, Work Item data appears in your Document in Polarion, and in exported copies, but it is not really part of the Document content. It is actually maintained in the underlying repository and simply displayed in your Document. You can see the same fields in the Work Item detail if you view the item in the integrated Tracker (Navigation > Work Items > Table, for example). Values for the ID and Status fields are preconfigured to appear by default. You can either remove these from the visual presentation, or add additional fields.

# **Configure Document data fields**

Click the Work Item Type drop-down list control in the Document editor toolbar.



Add a Work Item Type.



Configure fields to show in **Work Item** properties

- 2. On the drop-down menu, choose Configure. The Configure Work Item Presentation dialog appears. The dialog presents a table showing the Work Item types currently supported in the Document.
- 3. Locate the row for the Work Item type you want to configure. For example, if the table contains rows with presentation configurations for Requirement, Test Case, and Defect types, and you want to specify the fields to show with Requirement type items, you will need to work with fields in the row for Requirement.
- 4. Click the **Select Fields** button, and in the **Select Fields** dialog, add the fields you want to appear in the Document from the list on the left to the list on the right. You can optionally set whether to display fields at the start or end of Work Items. For more information, see Configure Work Item Presentation Dialog.

# Tip

The **Select Fields** dialog can also be invoked via the (Pane Settings) icon in the Work Item Properties sidebar (enabled when a Work Item is selected in a Document). When invoked from there, the fields configuration can be saved as a user-specific personal configuration only for the current user, or as the default for the Document.

See also Configure Work Item Presentation Dialog and Configure Document Work Item types.

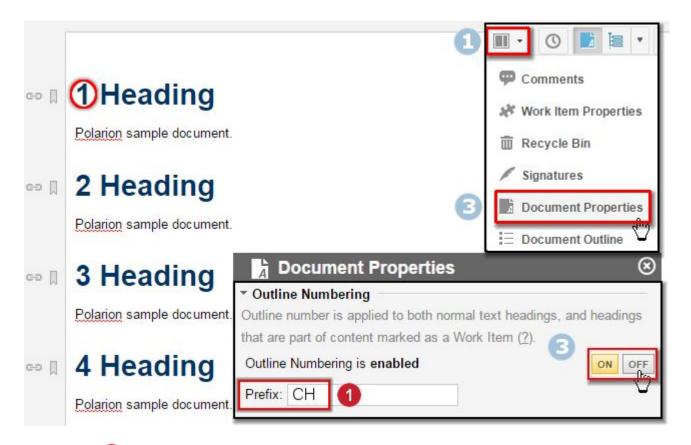
#### **Use Outline numbering**

Outline numbering can be applied to Document headings or Work Items in a Document. Outline numbering in Polarion behaves similarly to outline numbering in Microsoft Word.

# **Outline numbering for Document headings**

The outline numbering feature adds outline numbers to document headings. When Importing a Word document, Outline numbering is enabled by default, but can be disabled before the actual import, as described in the procedure below.

- 1. On the Document Editor toolbar, click ( ) to display a menu.
- 2. In the menu, click ( Document Properties.
- 3. Us the buttons in the **Outline Numbering** section to turn numbering **ON** or **OFF** as shown in the figure below.



4. (Optional.) • Enter a prefix to be shown before the outline numbers.

#### **Note**

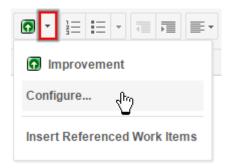
Prefix is only applied to Work Items within the heading, and is visible in **Tree** View of Work Items.

5. Click the 🖥 Save icon to save the Document and apply the changes.

# **Outline numbering for Work Items**

Polarion can also automatically add outline numbers to all Work Items within a Document.

- 1. Select an existing Work Item or locate the insertion point on an empty line.
- 2. Click the Work Item Type selector icon in the Document Editor toolbar.



#### **Note**

The Work Item Icon you seen depends on the Work Item type the Document is configured to contain.

- 3. Choose Configure. The Configure Work Item Presentation dialog box appears.
- 4. For every Work Item type in the table (there is one per row), click **Select Fields**. Then, in the Select Fields dialog box, add Outline Number from the list of fields on the left to the list of fields on the right.
- 5. (Optional) You can edit the order that the selected fields will appear using the keys. (The field at the top of the list appears at the top of the table.)
- 6. Select a layout option:
  - Show in Sidebar only (Outline numbers only appear in the Document Properties, **Work Item Properties** and **□Document Outline** sidebars.)
  - **Show in Document only** Outline numbers only appear in the Document.

This only applies to the Work Item and Document sidebars. Outline Numbers always appear in the **Document Outline** Sidebar.

Show in Sidebar and document (Outline numbers appear in both the document and respective sidebars.)

(An additional **Show at** option appears letting you select whether to show the outline numbers Before or After the Work Item.)

- Show Fields at End as Table Displays any field with selected in a table at the end of the Work Item.
- 7. Click **Set as Default**, then click **OK**.

# Personal outline numbering

You can optionally set a personal, user-specific preference for Work Item outline numbering if outline numbering is not otherwise applied. In this case, you see outline numbers but other users do not.

- Click Select Fields.
- 3. The Configure Work Item Presentation dialog will appear but will contain an additional Set as Personal button beside Set as Default.

#### Note

The **Reset** button resets any field customization to the default settings.

# Work Item Recycle Bin

When some Document content is marked as a Work Item, a Work Item artifact of a specified type is created in the underlying repository. During editing of a Document, authors or editors may decide to unmark some content so that it is no longer a Work Item, or (more likely) to merge one Work Item paragraph or section into the preceding Work Item content. In such cases, the Work Item content either no longer appears in the Document, or it no longer appears as a separate Work Item. However, the artifact still exists in the underlying Repository. This means that it is subject to workflow, planning, auto-assignment, or any other Work Item and process management features and actions. In the early stages of the Document's development, having such **orphaned Work Items** in a Document may be perfectly acceptable because you might still want to do something with them — put them back into the Document later, or put them into another Document, for example.

The Document Sidebar's Recycle Bin panel shows any Work Items that exist in the repository, but are no longer present in the Document. (You will also see these items in the Table view of the Document. This view contains a button **Open in Document's Recycle Bin** which alerts any viewer to its state, as well as providing access to it in the Recycle Bin.) The Recycle Bin provides you with some options for handling these stored but unused Work Items. When you think a Document is complete, or nearing completion, it is a good idea to check the Document's Recycle Bin and clean up any Work Items shown there so as not to clutter the integrated Tracker with Work Items that are not intended to be processed.

# View the Recycle Bin

- 1. In the Document toolbar, click the Sidebar icon to invoke a drop-down menu.
- 2. On the drop-down menu, choose **Recycle Bin** to display the sidebar's **Recycle Bin** sidebar.

### Manage Recycle Bin items

The Recycle Bin panel provides several options for cleaning up Work Items in the Recycle Bin:

- Reinsert Work Item(s) into the current Document
- Move Work Item(s) to another Document in the project
- Disassociate Work Item(s) from the Document but keep them in the project, to be accessed only via the Work Items topic of Navigation.
- Delete Work Item(s) from the Document and the repository.

# Tip

For the procedures in the rest of this topic, you may find it useful to view: Recycle Bin sidebar.

For a reference on what actions send content to the Recycle Bin, see subtopic of the above: Content Handling.

#### **Restore items to the current Document**

- 1. Place the insertion point (caret) at the position where you want to insert the Work Item(s) currently sitting in the Recycle Bin.
- 2. In the Recycle Bin, select the item(s) you want to insert by selecting the check box preceding the ID-Title.
- 3. Click the **Insert** button to insert the selected Work Item(s) into the Document at the current insertion point.
- 4. Save the Document.

#### Move items to another Document

You can move a Work Item from the Recycle Bin of one Document to a different Document. Note that you must have permissions allowing you to modify the target Document and to create new Work Items in it.

- 1. In the Recycle Bin pane of the Sidebar, select the item(s) you want to move by selecting the check box next to each one.
- 2. Click the **Move** button. The **Move Work Items** dialog box appears.
- 3. In the dialog box, select the **Move selected Work Item(s) to Document** option.
- 4. In the **Space** list, select the name of the space where the target Document resides. The default selection is the space of the current Document. The **Document** list is populated with the names of the Documents residing in the selected space.
- 5. In the **Document** list, select the name of the target Document, click **OK**, and finally save the Document.

The selected Work Items are disassociated from the current Document and removed from its Recycle Bin. They are then associated with, and appended to the content of the target Document. If you want the moved item(s) to appear elsewhere in the target Document, you need to open that Document for editing and move them manually.

# Disassociate items from the Document

You can disassociate a Work Item in a Document's Recycle Bin from the Document but still keep it in the project.

1. In the **Recycle Bin** panel of the Sidebar, select the item(s) you want to disassociate by selecting the check box next to each one.

- 2. Click the **Move** button. The **Move Work Items** dialog box appears.
- 3. In the dialog box, select the **Move selected Work Item(s) out of Document** option, click **OK**, and then Save the Document.

The selected Work Items are disassociated from the current Document and removed from the Recycle Bin. However, the data artifacts are maintained in the repository with their same IDs and properties. People using the integrated Tracker will be able to manage them as Tracker-based Work Items.

# **Delete items completely**

You can delete Work Items from the Recycle Bin, disassociating them from the Document and deleting them from the repository.

#### **Note**

Deleting a Work Item does *not* delete its history from the system.

- 1. Select the item(s) you want to delete by checking their boxes.
- 2. Click the **Delete selected Work Items** button and respond to the confirmation prompt.
- 3. Save the Document.

# Tip

You can delete a Work Item without first moving it to the Recycle Bin: click the Work Item icon in the left margin and select **Delete** on the menu. The item is deleted from the Document and the repository. (It is not deleted from the Document history.)

If you want to delete a Work Item that is not defined in a Document, see Delete Work Items.

#### **Filter Work Items in Documents**

You can filter a Document to show only a subset of all the Work Items it contains. For example, a specification might contain requirements for different operating systems. Assuming a custom field named **OS** to designate the operating system, you could filter the Document to show only requirements for Ubuntu Linux, for example, by querying on the custom field.

Filtering is accomplished using the integrated Apache Lucene query engine and query language. The Visual Query Builder tool is available to help you construct simple or complex queries graphically, if you are not familiar with the query language syntax.

- 1. On the Document toolbar, click Actions Filter. The Document becomes read-only, and filter controls appear in the Document Editor toolbar.
- 2. In the **Filter Work Items** field, enter the visual query elements, query language syntax, or a combination of the two as needed to obtain the results you want: os:Ubuntu, or severity:must\_have, for example.

# Tip

You can also filter Work Items when viewing any revision of a Document in the Document history.

If an item shown by the filter is the child of one or more other items, the parent items are also shown in the Document regardless of whether or not they match the filter criteria.

While viewing the Document in its filtered state, you cannot edit the text. However, you can:

- Access the Document history and compare revisions.
- View and edit Work Item properties and links in the Document sidebar.
- Export the filtered Document using Word Round-trip **Export Document**. Only the Work Items shown by the filtering are included in the exported document.
- Print the Document or export it to PDF. Only the Work Items shown by the filtering appear in the printed copy or exported PDF file.
- Reply to any comment that is shown when the filtering is engaged (but not create new comments).
- Access linked Work Items that are hidden by the filter. Linked items appear in the Properties of Work Items shown by the filter and if clicked, open in a new browser tab.
- Return to the unfiltered view of the Document by clicking **Close Filter**.

See also Search Work Items.

# **Comments and threads**

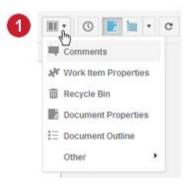
# ALM REQ QA PRO REV

Polarion LiveDoc Documents support comments in the body text, including the text of Work Items. Comments in Documents work in much the same way as in Microsoft Word and other desktop word processing applications. Comments may be added to Documents either online in the portal, or in Microsoft Word files that have been exported for comment and/or collaboration, and then reimported using the Word Round-trip feature.

Document comments are not the same thing as the default Work Item data field named **Comments**. Document comments are only visible in the Document Editor, while Work Item comments are only visible in the Table and other views of a Document's Work Items. Document comments are not exported or imported during Excel round-trip.

In order to add comments and replies, or resolve comments, you must be assigned a user role that is granted these permissions.

Please refer to the following figure as necessary when reading subsequent comment sections:





1 Show sidebar	4 View comments in sidebar
2 Document text	5 Unselected
3 2-way access	6 Selected in sidebar

# Add new comments

1. Place the insertion point (caret) in the position in the Document where you want to insert a comment.

- 2. On the Document Editor toolbar, click the **Add Comment** button. The sidebar displays the **Comments** panel, and a new comment is added ready for editing.
- 3. Type the text of the comment and click the **Save** button on the Document toolbar when finished. A comment icon appears in the Document text at the insertion point.

# View existing comments

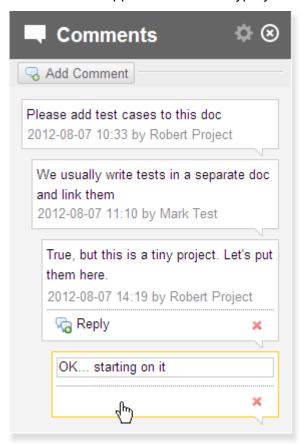
The Document body displays a comment icon at every point where a comment has been inserted.

Hover your pointer above any comment icon to see the comment text in a tooltip. Click a comment icon to view the comment text in the Comments panel of the Document Sidebar.

You can also access comments by browsing the Comments panel of the sidebar. Clicking on a comment in the sidebar scrolls the Document to the location of the selected comment icon in the text.

# Reply to an existing comment

You can use the comment reply feature to create threaded discussions inside a Document. To reply to an existing comment, select the comment in the sidebar **Comments** panel and click **Reply**. A new comment appears indented. Type your reply and click **Save** in the Document Editor toolbar.



#### Delete a comment

1. Select the comment you want to delete in the Document or the sidebar **Comments** panel.

2. In the sidebar, click the **Delete** in the selected comment.

Tip

If there are any replies to the comment you are deleting, all of them are also deleted.

#### Resolve Document comments and threads

When stakeholders reach consensus through comments and comment threads, it can be useful to mark the comments and/or comment threads as Resolved. This hides them, after which individual users can opt to view them. It is also possible to reopen resolved comments if a stakeholder decides that further discussion is needed.

See also: Resolve Approval Comments, Resolve Work Item Comments and Threads.

- 1. Open the **Comments** panel of the Document sidebar using the **Show Sidebar** command on the Document Editor toolbar.
- 2. Locate the comment to mark as resolved, hover your pointer over it, and click **✓ Resolve**. In comment threads, you must mark the top-level comment.

You can view any resolved comments that are hidden:

- 1. Open the **Comments** panel of the Document sidebar.
- 2. Click the (Pane Settings) icon in the header and on the menu, choose **Show Resolved** Comments.

You can reopen resolved comments or threads:

- 1. Open the **Comments** panel of the Document sidebar.
- 2. Show resolved comments by clicking the (Pane Settings) icon in the header and on the menu, and choosing **Show Resolved Comments**.
- 3. Locate the resolved comment you want to reopen, hover over it and click **Reopen**.

Tip

Replying to any comment in a resolved thread reopens the entire thread.

See also: Signature Comments.

# **Required Work Item comments**

As previously mentioned, only Document comments are accessible in the Document Editor. However, it is possible for administrators to configure projects to make the **Comments** field of Work Items a required field. When so configured, Work Items changes that affect the workflow cannot be saved without adding a comment to the **Comments** field. This prevents Document users from saving the Document if they change a field such as **Status** in **Work Item Properties**, because the **Comments** field is not accessible in the Document Editor. Administrators and project leaders should be sure to inform their users and teams if Work Item comments are required to transition the workflow.

If you are prompted to add a Work Item comment after changing Status in one or more of the Document's Work Items, it means you cannot change Status in the Document Editor. You will instead need to use the Table view of the Document, which is accessible from the Document Editor toolbar.

In this situation, if you have unsaved content changes, go back to all Work Items in which you have changed **Status**, and set the value back to what it was before you changed it. This should enable you to save the content changes. If you have no pending content changes, you just can switch to the Table view immediately and opt to cancel changes when prompted.

In the Table view, browse or run a query to find the Work Items for which you want to change Status. For each item, change the Status field as desired and add a new comment to the Comments.

# Comments in Excel Round-trip

If Work Items containing Document comments are exported for Excel round-trip, and the Work Item **Description** field is allowed by the exporting user to be editable in the exported Excel workbook, a placeholder {comment:id} is inserted into description text in the exported Excel workbook. This preserves the comment marker for the round-trip process and should not be removed from the exported Excel document.

# Images and other attachments

You can display images or attachment previews in a document. Simply place the cursor where you want the image or attachment preview to appear in the document an click the button. See Insert Image or Attachment Preview for details.

After inserting an image or attachment preview, you can add a caption above or below it, resize it, or launch a full screen preview of it.

#### Note

Your system or project administrator can optionally limit the file size of attachments. Contact your administrator if you are unable to upload an attachment due to such restriction.

#### Download an attachment

- 1. Open the **Attachments** panel of the Document Sidebar if it is not already open.
- 2. Locate the attachment you want to update. Optionally use the search box at the top of the panel.
- 3. Hover your pointer over the attachment you want to download and click the icon.
- 4. If prompted by your browser, select the location on your local file system where you want to store the downloaded file, and confirm the save action. Otherwise, you can look for the downloaded file in the location where your browser stores downloads.

# Update an attachment

- 1. Open the **Attachments** panel of the Document Sidebar if it is not already open.
- 2. Locate the attachment you want to update. Optionally use the search box at the top of the panel.

- 3. Hover your pointer over the attachment you want to update and click the T (Update) icon.
- 4. Locate the updated version of the attachment file in your local file system, select it, and confirm your selection in your file selection dialog.

#### Note

The file you upload should have the same file name and extension as the existing attachment. If you upload a different extension, you are asked if you wish to overwrite the existing attachment.

5. After the selected file uploads, save the Document to complete the attachment update.

### **Delete an Attachment**

- 1. Open the **Attachments** panel of the Document Sidebar if it is not already open.
- 2. Locate the attachment you want to delete. Optionally use the search box at the top of the panel.
- 3. Hover your pointer over the attachment you want to delete and click the (Delete) icon .
- 4. Save the Document to complete the delete operation.

#### Tip

If you delete the wrong attachment by mistake, you can click **Undo** on the Document Editor toolbar, provided you did not yet save the Document. If you did save it, you can revert the Document to the previous revision in the **History**.

See also: Attachments sidebar.

# **Query for attachments**

You can easily search Work Items to find all that have some kind of attachment. You can refine the search to include only items with attachments of specified file type(s), and also filter selected attachments according to:

- Title
- File name
- Author
- File size
- Date modified
- Content (provided that indexing of attachment content is enabled in the system configuration - see Advanced System Tuning).

To query for attachments in the Table view of Work Items, click the - icon in the Query Builder, select one of the query elements beginning with the word **Attachment**, and select appropriate criteria for the element. Add additional Attachment elements to further refine the query. For more information on queries, see Search Work Items.

# Insert a Cross-reference

Cross-references let you link to **Headings** or Work Items within a Document.

When clicked, they scroll right to their target.

To distinguish them from Live Links, cross-references are underlined with a dashed gray line.

See the 1.1 - Purpose section for details

They are underlined with a solid black line in exported PDFs.

#### Note

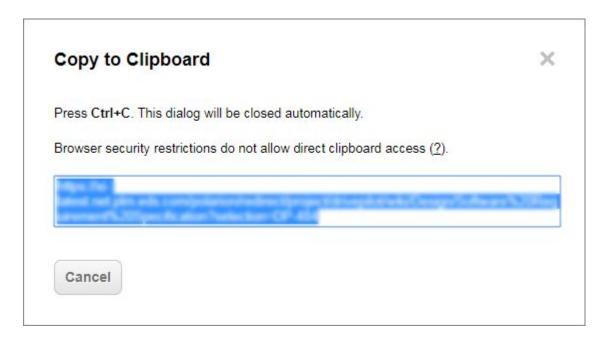
- Unlike Live Links, cross-references are only navigational links and do not reflect an item's Status or Severity.
- The **Copy Cross-reference** option is only visible for saved items.
- 1. Scroll to the **Heading** or **Work Item** that you'd like to reference.
- 2. Referring to the figure below, click the **Heading** or **Work Item** icon in the left margin (1), hover over Copy (2), and finally click Cross-reference (3).



# Tip

Always use the method above to copy cross-references. If you copy them any other way, they may not work as expected.

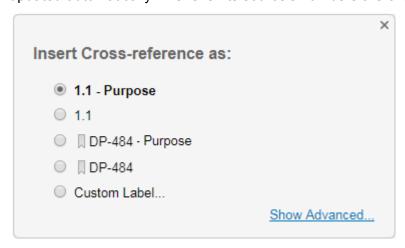
3. Press Ctrl+C, or the equivalent copy-to-clipboard shortcut for your operating system, when the Copy to Clipboard dialog box appears.



- 4. Scroll to, and click where you want to place the cross-reference.
- 5. Press Ctrl+V.

A cross-reference can be added to the Document's body text or to the fields of a Work Item contained within the Document.

6. If the Document has outline numbering **ON** in the <u>Document Properties sidebar</u>, then the default link text included them and the section's title in the link. A reference's outline numbers are updated automatically whenever its source's numbers are changed.



7. Click \* to keep the default outline numbers and title, or select the options to omit the title or replace the outline numbers with the Project ID.

#### Tip

Click **Custom Label** to define your own cross-reference link text.

The cross-reference appears in the selected position.

# 2 Component Design

See the 1.1 - Purpose section for details

#### Note

- If a referenced Work Item or Heading is removed, any cross-references pointing to it change to Missing cross-reference.
  - (It also contains the missing item's ID, so that it can be easily tracked down.)
- If the referenced item is filtered out of the Document, then the cross-reference link remains but does not scroll when clicked.
- For an accurate cross-reference history view it in the Document's Wistory.
- When comparing different Documents, cross-references are highlighted as changed because they contain different URLs.

# When exporting Cross-references:

### **Exporting to Microsoft Word:**

- Cross-references created in Polarion will survive a Word Round-trip, but instead of scrolling in the Word document, they open the referenced object in Polarion.
- Cross-references created in Word are *not* recognized when imported into Polarion.

# **Exporting Cross-references to PDF:**

#### Note

Please use Adobe Acrobat Reader to view PDF documents.

- If a cross-referenced item exists within the same PDF, then clicking its cross-reference scrolls right to it.
- If a cross-referenced item is *not* within the PDF, then clicking its cross-reference opens the item in the Document Editor in Polarion.
- If a Document was filtered before it was exported, cross-references for filtered items are exported as plain text.
  - (They do not scroll to, or open the items in Polarion when clicked.)
- Cross-reference links are underlined with a solid black line in exported PDFs

# The difference between a Link and a Cross-reference:

- A cross-reference scrolls to a heading or Work Item within a Document.
- When a Live Link is created, then clicking Heading or Work Item icon Copy Link opens the linked item in the Work item viewer.
- Cross reference links can contain outline numbers. Live Links cannot.

# Cross-references in branched, variant or reused Documents:

Things to know about cross-references in Branched or Variant or Reused Documents:

- If a cross-reference is pointed to a section within a master Document, it will point to its matching counterpart section in Branched or Variant or Reused versions.
- If a referenced Work Item that is targeted in a cross-reference is overwritten, the cross-reference is updated so that it still scrolls to the same position.
- When a Work Item is opened in the Work Item form, any cross-reference contained within it link to the master Document.
- If you change the title of a cross-referenced Heading or Work item, the cross-reference link text updates accordingly.

(But only within the Document version in that the Heading is changed in.)

# Note

Currently cross-references between Documents, Revisions and Baselines is not fully supported and cannot be rendered as Live Links.

You can repair missing cross-references by inserting the missing item using the **Insert Live/Frozen reference** merge action.

# **Insert a Diagram**

# ALM REQ

Polarion provides the Diagram Editor tool for creating and editing diagrams. You can use it to create and edit diagrams in Documents, including Work Items contained in Documents. You can also use it to create diagrams in the Description field of Work Items you edit in the Work Items Table view.

# Insert or Edit a Formula

You can add  $\sqrt{x}$  Mathematical Formulas to a LiveDoc or Work Item rich text field.

The formulas support LaTeX syntax.

## Tip

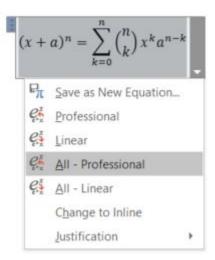
- See Syntax help for a list of supported symbols and examples.
- You can use third-party software or an online editor to create a formula and copy and paste the LaTeX syntax they output into Polarion.

## What you can do with mathematical formulas

- Insert a formula into a Document or Work Item.
- Edit an existing formula.
- Copy and paste formulas between Documents or Work Items.
- Edit a formula that was created in Microsoft Word and imported into Polarion.

#### Caution

Microsoft Word formulas must be saved in **Professional** mode to import correctly into Polarion.



## How formulas appear when exported

- In **Documents**:
  - **ReqIF**: Not supported
  - Microsoft Word RoundTrip: Appear as images in Word, but can be edited by importing them back into Polarion.

(You need to replace them to edit them in Word, but the Word formulas will be editable if imported back into Polarion.)

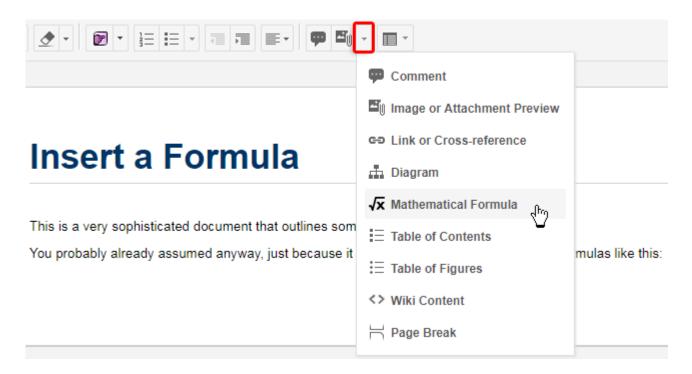
- o **PDF**: The formulas are rendered as images
- In Work Items: (Via Table View.)
  - o **Excel**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)
  - o **HTML**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)
  - o **MS Project**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)
  - o **PDF**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)
  - o **RTF**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)
  - o **XML**: Formulas are replaced with LaTeX syntax. (You can copy and paste it to recreate the formula.)

#### Insert a formula into a Document or Work Item rich text field

- 1. Open the Document or Work Item field that you want to edit.
- 2. Click where you want to insert a formula.
- Click the √x icon.
  - In rich-text fields of Work Items, it appears on the toolbar when the field is in edit mode.

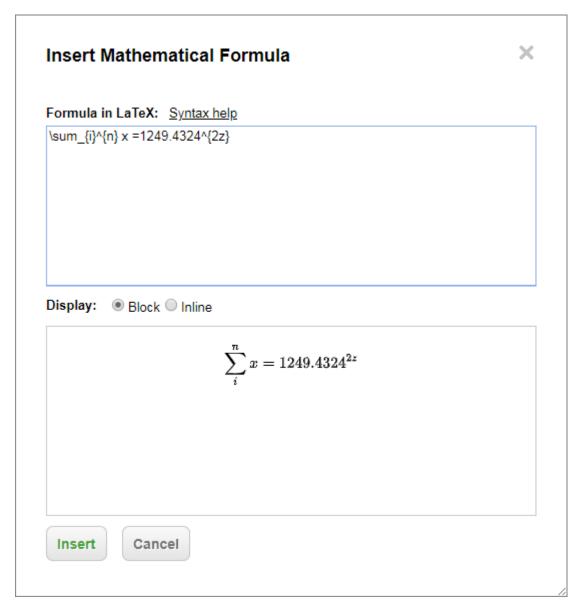


• In a Document, choose **√x Mathematical Formula** on the **Insert** menu of the Document Editor toolbar:



The Insert Mathematical Formula dialog box appears.

4. Enter your formula in the **Formula in LaTeX** field.



A preview appears in the lower box.

## Tip

If your formula does not render as expected, click Syntax help to view a list of supported symbols and examples.

- 5. Select whether you want the formula to appear as a **Block** or **Inline**.
  - **Block**: Centers the formula on the page or field and does not allow for any content on either side of it.
  - Inline: Inserts the formula wherever the cursor was placed and allows for content on either side.

Block:

$$\sum_{i}^{n} x = 1249.4324^{2z}$$

Inline:

Selecting Inline lets you add additional content  $\sum_i^n x = 1249.4324^{2z}$  on the same line.

6. Click Insert. The formula is inserted into the Document or Work Item field.

## Edit an existing formula

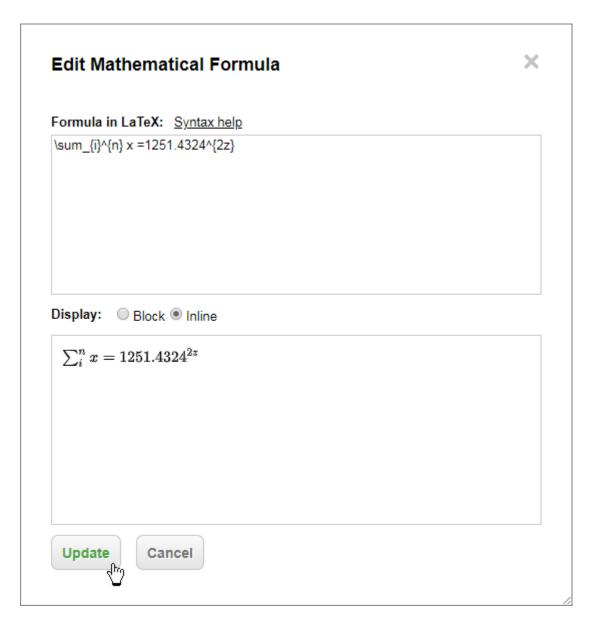
- 1. Open the Document or Work Item field that contains the formula you want to edit.
- 2. Select the formula and click **Edit**. (Or double-click the formula.)

Inline:

Selecting Inline lets you add additional content on the same line.



3. Edit the formula in the **Formula in LaTeX** field, and check the preview to ensure that it's correct.



4. Click Update.

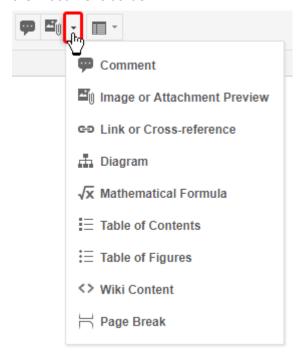
## **Insert a Table of Contents**

You can insert a table of contents (TOC) into a Document. The content of the TOC is based on the headings in the Document. If you insert a TOC into a Document that has no headings, the message No table of contents entries found appears at the point you insert the TOC. The TOC is automatically updated when you add new headings and save the Document. TOC items are links which scroll the page to the respective heading. You can explicitly update or delete the Document's TOC any time, and you can insert a new paragraph before or after the TOC.

The Document Editor allows up to 29 heading levels. However, the TOC displays only up to 5 heading levels.

1. Place the insertion point (caret) at the position in the Document where you want to insert the TOC.

2. In the Document editor toolbar, click the Insert drop-down (see figure below) and choose **Table of Contents**. The TOC is inserted and the (Table of Contents) icon appears outside the Document border.



## **Update the TOC**

The Document TOC is updated automatically when you save the Document. However, you can explicitly update it while editing the Document. Note that you cannot manually edit any text in the Table of Contents.

- 1. Click the TOC icon on the Document background. The TOC actions menu appears.
- 2. Choose **Update** on the TOC actions menu.

## **Delete the TOC**

- 1. Click the TOC icon on the Document background. The TOC actions menu appears.
- 2. Choose **Delete** on the TOC actions menu. The TOC is no longer visible in the Document.
- 3. Save the Document to complete the deletion of the TOC.

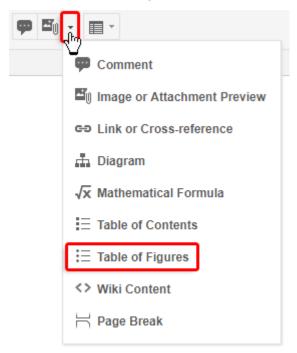
## **Insert New Paragraphs**

You can insert a new paragraph before or after the Table of Contents in a Document.

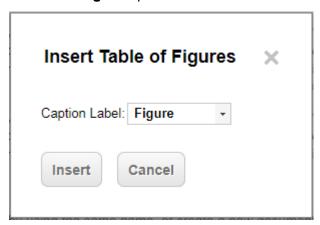
- 1. Click the TOC icon on the Document background. The TOC actions menu appears.
- 2. On the TOC actions menu choose Insert Paragraph Above or Insert Paragraph Below.

## **Insert a Table of Figures**

1. To insert a Table of Figures in a Document, click the Insert drop-down, select **Table of Figures**.



2. Select the Figure option.



The inserted table includes links to images and/or diagrams in the Document that have a caption with the caption label **Figure**. For more information, see Captions .

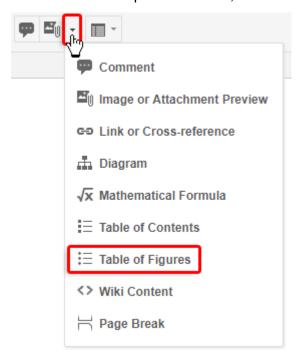
## **Note**

The Table of Figures appears only in documents created by export to PDF and Round-trip for Microsoft Word. A placeholder appears in your Polarion Document.

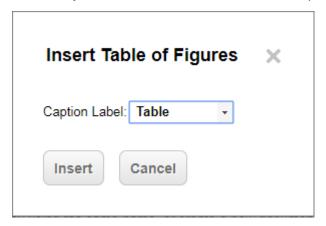
## Insert a Table of Tables

The procedure to insert a Table of Tables is identical to that for inserting a Table of Figures, except that you choose a different caption label option.

1. Click the Insert drop-down control, and select  $\stackrel{:}{=}$  **Table of Figures**.



2. In the Caption Label field, select the Table option.



The inserted table includes links to tables in the Document that have a caption with the caption label Table. For more information, see Captions .

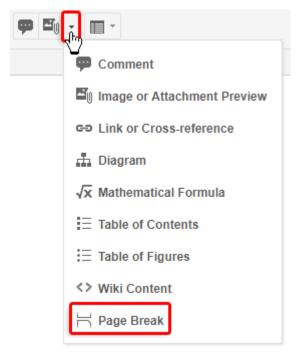
## **Note**

The Table of Tables appears only in documents generated by export to PDF and Round-trip for Microsoft Word. A placeholder appears in your Polarion Document.

## Insert page breaks

You can insert explicit page breaks in a Document. Such breaks also occur in exported documents after export to PDF and Microsoft Word (Round-trip).

- 1. Place the insertion point at the end of the line where you want to insert a page break, or on an empty line.
- 2. In the Document editor toolbar, click the insert drop-down and choose **Page Break**.



3. The icon appears outside the Document border on the line where the page break occurs in exported content.

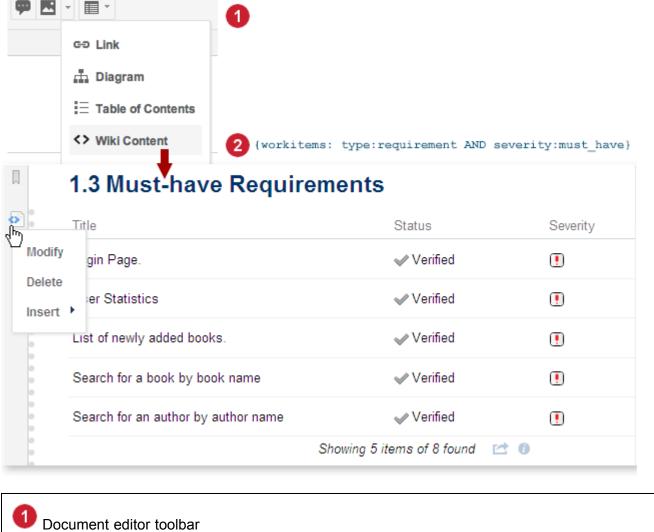


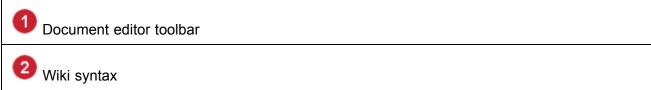
To remove a page break, click the Page Break icon and choose **Delete**.

## Insert Classic Wiki macros and content

In Documents, it is possible to insert textual content formatted with Classic Wiki syntax, or Classic Wiki macros which render content dynamically. For example, you might create a Document that

reports the current status of Work Items in the project using the {workitems} macro with a query parameter to display some subset of the Work Items in the Document — all currently open items, or items with high severity, for example.





After opening the Document for editing, do the following:

 In the Document Editor toolbar, click the insert drop-down (see figure below) and choose Wiki Content.

The **Insert or Modify Wiki Content** dialog box appears.

- 2. Enter the desired Classic Wiki macro or other content with Classic Wiki mark-up into the edit field in the dialog box, and click **OK** when finished.
- 3. Click 🖥 Save on the Document Editor toolbar.

For general information on using the Classic Wiki, including editing of Classic Wiki pages, see Using Classic Wiki. For information on Wiki macros, see Adding Dynamic Content in Wiki Pages.

See also Macros and Wiki Mark-up Support in Word Round-trip.

## **Modify Wiki Content**

You can modify existing embedded wiki content. For example, you might modify the query of an existing workitems macro to retrieve a different set of Work Items, or display more or fewer Work Item fields.

- 1. After locating the content you want to modify, click the Wiki Content icon to the left of the Document border.
- 2. On the menu, click **Modify** to display the Wiki Content dialog box.
- 3. Change the content and desired and click **OK**.

#### **Delete Embedded Wiki Content**

• After locating the content you want to modify, click the Wiki Content icon to the left of the Document border, and then choose the **Delete** item on the menu.

## **Resolve conflicts**

If you and another user edit the same Document at the same time, it is possible that your changes can conflict with changes made by the other user.

Polarion attempts to merge concurrent changes to Documents and does so silently when it is clear that it is safe. In cases where there is some question, a dialog box is presented to the user saving changes that potentially conflict. The dialog box summarizes the issues, provides some explanatory Help text, and provides links that open separate browser tabs in which the user can review his or her potentially conflicting changes, and the changes of other users. The user can then make an informed decision about saving the pending changes. For information on how Polarion handles different merge scenarios, see the reference topic Merge concurrent changes.

During editing, an item Compare with Unsaved Changes appears on the Actions Menu of the Document Editor toolbar. This action opens a page in a new browser tab where you can compare the current state of the Document in the repository, and your unsaved changes.

## **Document Round-trip for Microsoft Word**

#### Share Documents with external stakeholders

# ALM REQ QA

The Round-trip for Microsoft Word ("Word Round-trip) feature enables you to export any LiveDoc Document to Microsoft Word format and share it with people who do not have access to your Polarion portal. It can also enable a Document author to work on the Document off-line. The exported document, along with all changes to the content, can then be imported back to Polarion, where the changes are incorporated into the Document in Polarion, and the Document history is updated.

When exporting the Document, you can control the level of change that people can make to the exported document.

- Export for Review: Content cannot be changed. Comments can be added to Work Items or text. On reimport, the comments are incorporated into the Document in the Polarion portal.
  - After reimport to Polarion, comments associated with text that is not part of a Work Item appear as usual in the Document. Comments associated with content marked as Work Items appear as comments in the Document only. That is, they do not appear in other views of Work Items — in the **Comments** section of the Work Item in the Table view of the integrated Tracker, for example.
- **Export for Prioritization**: Change is restricted to simple, single-value Work Item data fields such as Severity. Allowed fields are specified by the person exporting the Document. The user of the exported round-trip document can also insert comments to Work Item descriptions and to text that is not part of a Work Item. When the round-trip document is reimported to Polarion after changes, the Document's Work Items are updated with any changes made in the exported document, and the Work Items' history is updated.
- **Export for Collaboration**: Enables the user of the exported document to change the text in the document as well as the description and/or severity level of Work Items, and multiline text and rich text fields. The document user can also insert comments to Work Item descriptions and to text that is not part of a Work Item, or remove existing comments. This export option can also be used by authors who want to work on a Document offline. In the exported round-trip document, Work Items can only be modified, not created, copied, or deleted. When a changed round-trip document is re-imported into Polarion, the original LiveDoc Document is updated with all changes to text content, comments, and Work Items, and the relevant histories are updated.

## Tip

You can prevent pages breaks from occurring within Work Items by enabling the **No Page** Break option in the Work Item Presentation dialog box, accessible in the Document Editor when you click the Work Item type icon in the toolbar. When selected for any Work Item type, Polarion attempts, as much as possible, not to have page breaks occur in Work Items in the exported Microsoft Word document.

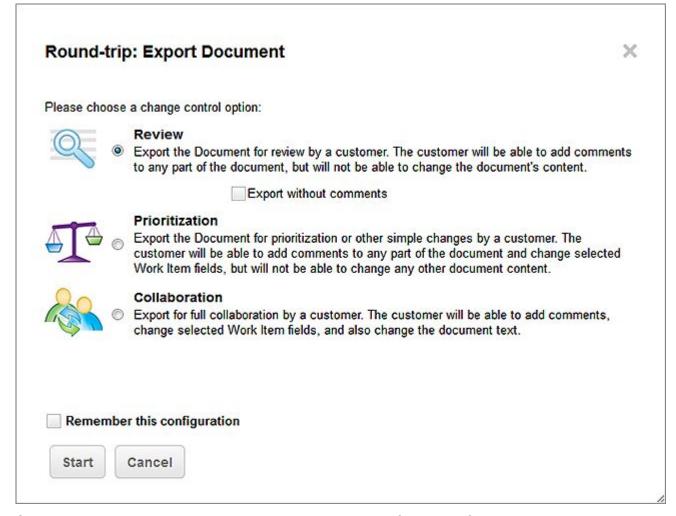
#### Warning

- If the Document contained comments when it was exported, and the user of the exported round-trip document changes the text of existing comments, these changes are not imported back to Polarion. Round-trip document users should add new comments.
- If the default paragraph style of a Microsoft Word document is customized, it will not affect its appearance when imported into Polarion, but will be applied when exported back into Word.
- HTML and CSS styles inserted into a Wiki content block may not export correctly. To ensure that styles are exported as expected, only use the styles that are created by Polarion's Document Editor.

 Polarion does not clean up HTML within a Wiki content block. If a block contains invalid HTML, even though it may look fine in a browser, it may look different when exported to PDF or Word formats.

**Recommendation:** Use something like HTML TIDY to clean up Wiki content block HTML before exporting.

 Cross-references created in Polarion will survive a Word Round-trip, but instead of scrolling in the Word document, they will open the referenced object in Polarion.
 Cross-references created in Word will not be recognized when imported into Polarion



See Macros and Wiki mark-up support in Word Round-trip for more information.

## **Note**

Organizations can optionally implement restrictions via the plugin API to influence the template used for export to Microsoft Word and Excel, with restrictions on individual exports by individual users, to include a watermark in the exported documents and/or to conform to other required policies. If such a plugin is in use, the Round-trip: Export Document dialog box shows a message: The selected template may be substituted with one that complies with company policy.

## Tip

Embedded image or attachment previews are preserved during the Word Round-trip process.

## **Export for review**

When you export a Document to Microsoft Word using the Export for Review option, changes to the exported document are limited to comments only — no other content can be changed.

- 1. Open the Document in the Document Editor. (Click it when browsing Documents and Pages in Navigation or in search results.)
- 2. In the Document Editor toolbar, click and choose Word Round-trip Export Document. The Round-trip: Export Document dialog box appears.
- 3. In the Round-trip: Export Document dialog box, select Review.
- 4. If you want to exclude comments currently in the Document from the export, select the Export without comments check box.
- 5. Click **Start**. When prompted, choose a location to store the exported document and click **OK**.

## **Export for prioritization**

When you export a Document to Microsoft Word using the Export for Prioritization option, changes to the exported document are limited to selected Work Item data fields and comments. The **Description** field of Work Items is always read-only in the exported file.

- 1. Open the Document in the Document Editor. (Click it when browsing **Documents and Pages** in Navigation or in search results.)
- 2. In the Document Editor toolbar, click and choose Word Round-trip Export Document. The Round-trip: Export Document dialog box appears.
- 3. In the Round-trip: Export Document dialog box, select Prioritization. A table of Work Item data fields appears in the dialog box listing the fields you may allow users of the exported document to change.
- 4. Select the check boxes next to the names of Work Item fields you wish to allow users of the exported document to change.

- 5. If you want to exclude comments currently in the Document from the export, select the **Export** without comments check box.
- 6. If you want your current selections to be the default for all subsequent exports, select the **Remember this configuration** check box.
- 7. Click **Start**. When prompted, choose a location to store the exported document and click **OK**.

#### **Export for collaboration**

When you export a Document to Word using the Export for Collaboration option, changing all document content is permitted, as well as changes to specified Work Item data fields, plus the addition of comments. The Description field of Work Items is always read-write.

To export a Document permitting full collaboration and content changes:

- 1. Open the Document in the Document Editor. (Click it when browsing **Documents and Pages** in Navigation or in search results.)
- 2. In the Document Editor toolbar, click and choose Word Round-trip Export Document.

  The Round-trip: Export Document dialog box appears.
- 3. In the **Round-trip: Export Document** dialog box, select **Collaboration**, then click **Start**. A table of Work Item data fields appears in the dialog box listing the fields you may allow users of the exported document to change.
- 4. Select the check boxes next to the names of Work Item fields you wish to allow users of the exported document to change.
- 5. If you want your current selections to be the default for all subsequent exports, select the **Remember this configuration** check box.
- 6. Click **Start**. When prompted, choose a location to store the exported document and click **OK**.

#### **Export table of contents**

If a Document was created using the Import from Microsoft Word feature, and the original document contained a table of contents (TOC), the properties of the original TOC are stored during the original import, and used when the Document is exported via the Word Round-trip feature. The TOC in the resulting Word document has the same properties as the TOC of the original source document, and they cannot be changed in the exported Word document.

If the Document was created as a new Document via the portal, and if it contains a TOC, the Word Round-trip exporter creates a standard TOC in the exported Word document.

## **Use Round-trip export templates**

When you create a new Document via import from Word, Polarion creates a Microsoft Word template file named **roundtrip.docx** based on the styles in the original Word document. If you later export the Document from Polarion using Word Round-trip Export, the content of the Document in Polarion will be formatted according to the template that was stored during the initial import.

Polarion comes with a global default export template that defines header, footer, and basic styles. The template is used to format Word documents created from Polarion LiveDoc Documents using Word Round-trip Export. You can optionally create a copy of this template to use with specific projects.

You can customize any export template to define the styling of the exported Round-trip document as described later in this topic. For information on accessing round-trip export templates, see Round-trip Export Template Locations.

A Word Round-trip template can define styles for all the heading levels using paragraphs with text **Heading N** where N is the heading level number — 7, for example. All the defined styles can be used in the round-trip Document by users to create headings. If the template does not include enough heading styles for the exported Polarion Document, the needed styles are created by Polarion with names **Heading N**, where N is the level number. Microsoft Word supports only nine heading levels, even in the numbering, so the headings above Heading 9 level cannot have an outline number.

To customize a round-trip export template:

- 1. Navigate to the **FAdministration** topic **FWork Items** → Round-trip Template in the repository or project scope, or to its location in the repository using the Repository Browser.
- 2. Download the **roundtrip.docx** to your local system.
- 3. Edit the **roundtrip.docx** file in Word to defines the styles you want, saving your changes locally.
- 4. Upload the edited **roundtrip.docx** file back to the repository in the same location you took it from, using the relevant Administration page or the Repository Browser.

If you downloaded a template file in order to create a project-scope round-trip export template for a project that did not already have one, upload to the project location described in Round-trip Export Template Locations.

#### Note

You need read and write permissions for the repository and the **templates** folder. For information on field IDs that you can include in a customized template, see Round-trip Export Template Fields.

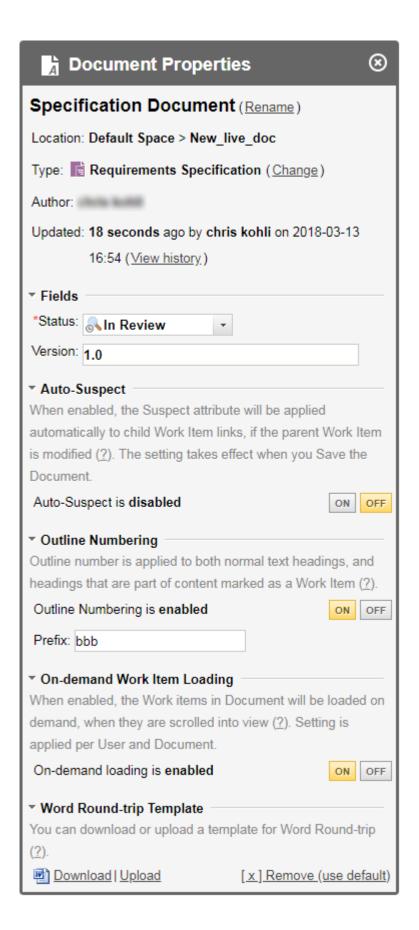
#### **Document-specific export templates**

When editing any Document, the Word Round-trip Template of the Document Properties panel of the Document Sidebar enables you to upload a Word document to be used as the export template for Word Round-trip exports of the current Document. Microsoft Office Word documents saved in .docx format can be used to create a template for Word Round-trip, or you can upload a customized Word Round-trip template previously downloaded from Polarion. When you click the Upload link in the Document Properties panel of the Document Sidebar, the Upload Round-trip Template dialog box presents options for each type of upload.

If a round-trip template exists:

The **Download** link is enabled in the sidebar pane which you can use to download the current round-trip template for customization. A mouse-over tool-tip indicates which template is available for downloading: default or custom.

• The **Remove (use default)** link is enabled. Clicking it removes the previously uploaded Document-specific round-trip template. The default round-trip template for the project will then be used for any Word Round-trip operations.



The **Document Properties** panel of the **Document Sidebar** provides access to a round-trip template file used for round-trip operations with the Document. It may be either the global or project default template, or a custom template uploaded by a user.

## **Customizing header and footer**

You define a header and footer as you would normally do in Word. However, you can include some predefined variables in the header/footer in content controls, for which values are substituted in the output during the export process:

- documentName (this is the Document's ID)
- documentTitle
- revision
- date (date picker content control in Word)
- generated
- generatedBy
- projectld
- projectName
- space
- spaceTitle
- updated
- updatedBy
- created
- createdBy
- productName
- productVersion

#### Tip

If you need more details about these, see the reference topic Round-trip Export Template Fields.

Polarion will fill these content controls, if present, when the Document is exported with Round-trip Export.

To add one of the above variables to a header/footer, you add a plain text content control to header/footer and set its tag to the value of the identifier. (For more information see the information about content controls on the web page Create a Form in Word 2010.)

#### Customize the table of contents

Using the Table of Contents feature in Word, insert a table of contents into the round-trip export template document if one is not already present, or if you want to change the style of an existing table. Set the depth of the table of contents in the usual way in the Word version you are using.

## **Customize styles**

The default templates contain several content elements — headings, paragraphs, and more — with some default styles. Use the Format Styles feature of Microsoft Word to change the properties of any existing style in the round-trip template you are customizing. Do not simply change the style of the individual element (heading or paragraph) using commands such as Format Paragraph or Format Font.

## Reimport round-trip documents and synchronize changes

After an exported round-trip document has been changed, it can be reimported to Polarion to synchronize the changes with the original LiveDoc Document. Before reimporting, make sure the changed round-trip document is in some location that is accessible when you are working in Polarion.

To reimport a previously exported round-trip document and synchronize external changes:

- 1. Open the original Document in the Document Editor. That is, the document that was previously exported using the Word Round-trip - Export Document command in the Document's ... menu. (Click the Document name when browsing **Documents and Pages** in Navigation or in search results.)
- 2. In the Document Editor toolbar, click and choose Word Round-trip Import Changes. The **Round-trip: Import Changes** dialog box appears.
- 3. In the dialog box, click **Browse**, navigate to and select the Word document you wish to reimport.
- 4. Select any of the options you want to apply to the reimport process:
  - Overwrite Conflicts: If selected, then changes in the re-imported document that conflict with the online Document will take precedence and overwrite the same part of the Document. If this option is not selected, and conflicts occur, the import will fail and no changes will be imported to Polarion, even on-conflicting changes.
  - Import Only Comments: If selected, all changes made to the exported Word document will be ignored by the reimport process except any comments which were added. Comments will be added to the Document in the same position as the appear in the Word document. No text is replaced.
- 5. If you want future round-trip reimport operations to use the same options, select the **Remember** this configuration check box.
- 6. Click Start to launch the round-trip re-import process. Assuming there are no conflicts that are overwritten via the Overwrite Conflicts option, the Live Document in Polarion is updated.

## **Notes on Reimporting**

When an exported round-trip document is reimported to Polarion, the following points apply.

 A conflict can occur if the same Document content is changed online after a round-trip export, and in the exported round-trip document. When the round-trip document is reimported, the conflict is reported. Reimport can succeed only if the importing user selects the **Overwrite Conflicts** option during the reimport operation.

## Warning

If the overwrite option is chosen, the content in the portal is overwritten with the content of the reimported document. The previous version of the content still exists in the Document history.

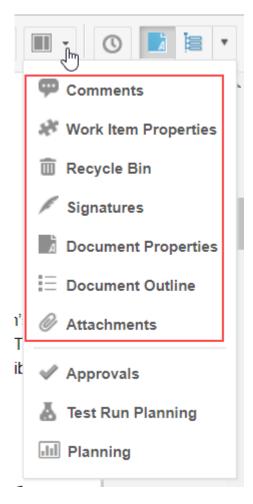
- If Document text is not changed in the round-trip document, then there is no new revision of the underlying system data file after import. However, it still shows in Document history.
- If Work Item description or any other field is unchanged in the round-trip document, there is no change in the field(s) in the history of the Work Item.

## **Document Properties, History, and Outline**

## **Document Properties**

Documents have a set of properties which can be read and/or set in the Document Editor. A complete history of every Document is automatically maintained. A new revision is created in the repository each time a Document is modified and saved. Revisions can be easily compared.

Document properties appear in the Properties panel of the Document Sidebar. Open it from the Document Editor toolbar by clicking Show Sidebar Document Properties.



Documents have the following properties:

- Name: Name of the document
- **Author**: User name of the person who created the Document.
- **Revision**: Current repository revision number of the Document.
- Auto-Suspect: Flag that determines if new Work Items created in the Document are automatically marked as Suspect.

## **Auto-Suspect Property**

If this Document property is enabled in the Document, the Suspect attribute is applied automatically to child Work Item links, if the parent Work Item is modified. (This setting takes effect when you 🖥 Save the Document.) The Auto-Suspect works for Document Work Items even if Auto-Suspect is not enabled in Administration. The Suspect flag set on Work Items can be useful for change impact analysis. For more information, see Suspect Default State and Suspect Links.

#### **Note**

The **Suspect** attribute is only applied to links to Child Work Items. (Not to Parent or Structural links within a document.)

See Suspect Links for details.

## **On-demand Work Item Loading**

The **On-demand Work Item Loading** feature can dramatically increase the response time, especially when editing Documents that contain many Work Items. When enabled, Work Items are not loaded until they are scrolled into view rather than waiting for all Work Items to be loaded. This makes editing large Documents more fluid and user friendly.

On-demand Work Item Loading is enabled by default for Documents containing 200 or more Work Items.

You can turn it **ON** or **OFF** for any Document in the Document Sidebar.

Administrators can set system-wide behavior in the system configuration file polarion.properties.

#### **Note**

## The **On-demand Work Item Loading** Property:

com.siemens.polarion.ui.document.lazyLoad.enabledByDefault=200

200 - The minimum number of Work Items a Document contains before on-demand loading is enabled. The default value is 200. Administrators can set any numerical value.

true - Enables the feature by default for all Documents.

false - Disables the feature by default for all Documents.

There are currently some limitations that occur when On-demand Work Item Loading is turned **ON**:

- If you wish to copy some items and paste them in the same Document, make sure that all the
  items have finished loading before pasting. The items are loaded if they have scrolled into view
  at least once. This limitation does not apply when pasting copied Work Items to a different
  Document.
- Caption numbers are sometimes replaced with the hash character # while viewing the Document in Polarion. The # hash characters are replaced with the correct numbers when the Document is exported. See Caption for details.

## Turn On-demand Work Item Loading on or off

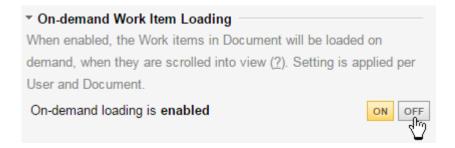
**On-demand Work Item Loading** can be set to **ON/ OFF** for a Document in the **Document Properties** Sidebar.

- 1. Click **Documents and Pages** in the **Navigation** bar, and select the target document.
- 2. Click on the top right.

3. Click Document Properties.

The Document Properties Sidebar appears on the right.

4. Click the **ON** or **OFF** button in the **On-demand Work Item Loading** section.



## **Document History**

Polarion automatically maintains a history of each Document. Every time you save the Document a new revision is created in the history. You can easily review any revision, and you can compare any two revisions to understand what changes took place between the older and the newer revisions.

#### **View Document History**

Each revision of the Document is automatically assigned a unique numerical ID, and is stamped with the date and time the revision was written to the repository. If you know the revision number you want to see, you can use your browser's Find feature to locate it on the Document history page. In addition to the date and time, the Document history page shows how long ago each revision took place.

To view a Document's history, click Oclock icon the Document Editor toolbar. You can also access the history via the View History link in the Document Properties panel of the Document Sidebar.

In the history page, you can browse the table of revisions until you find one you want to view. Then click **Show** in the **Actions** column. The Document loads in the editor in the state it was when the revision was saved, but the content is read-only and cannot be changed.

#### Note

If a Document revision is opened that contains a block of Wiki syntax, but a historical index of the database is currently in progress, then the block's content will not be displayed until the indexing is complete. While the historical indexing is in progress, the following warning message appears:

## Warning

Historical database indexing in progress. The Document's Wiki macros will not be rendered until it's complete.

You can use the **Current Revision** button to return to the current state of the Document in the editor.

## **Note**

Baseline revisions that do not match will have the revision number and name rendered in italics in all link states (normal, hover, selected).

## **Compare Document revisions**

# ALM REQ QA PRO REV

You can compare any two revisions in the Document history page. You cannot compare more than two revisions at once.

- 1. Open the Document history page.
- 2. In the **Revision #** column, check the boxes on the rows of the two revisions you want to compare and then click the **Compare** button.

You can compare either of two content states:

- **Document**: Compare the state of textual content in the two revisions.
- Work Item: Compare the state of Work Items defined in the document in the two revisions.

You can jump between changed elements or Work Items using the Changes buttons at the right side of the history page header. Return to the history page using the **History** button.

## Compare table changes when cells have been merged

When cells that contain text are merged and then compared with a previous version, the layout will appear a little strange in the comparison view.

Lorem	lpsum CaLC	
In ante turpis, lobortis vel neque quis, rutrum porttitor ipsum. Donec commodo condimentum justo, eu tempus velit lacinia sed.	In ante turpis, lobortis vel neque quis, rutrum portitor ipsum. Donec commodo condimentum justo, eu tempus velit lacinia sed. Sed luctus risus et dui malesuada euismod. Aenean in viverra metus, et volutpat ex. Vivamus congue volutpat lorem, eu consectetur nisi ultrices sed. Quisque id dictum justo, in blandit nunc. Quisque a ante sed quam malesuada malesuada non quis elit. Praesent ullamcorper volutpat enim, vel egestas ex ornare venenatis. Donec vel enim non velit euismod feugiat quis eu felis. Duis ornare est a velit varius imperdiet. Etiam bibendum ante euismod elit fermentum, at viverra mi rhoncus. Curabitur mattis magna sed dapibus semper. Donec porta faucibus ornare. Nunc mollis, felis facilisis faucibus mollis, tortor augue convallis libero, non ultricies velit velit ut metus. Fusce lacinia nunc quis mi congue dictum.	
Sed luctus risus et dui malesuada euismod. Aenean in viverra metus, et volutpat ex.	Quisque id dictum justo, in blandit nunc	Quisque id dictum justo, in blandit nunc

This is because it is more important to view the actual changes than to display the layout correctly in a comparison view.

(This only affects the comparison view. When exporting or viewing the most recent, merged cell revision, or the historical revision before the merge, both layouts will appear as expected.)

## Export comparison results to PDF

After comparing two revisions, you can export the comparison result to PDF. You can only export the Document (content) view.

- 1. Compare 2 revisions as described earlier.
- 2. If you are in the Work Items view, click the **Document** button on the toolbar to return to the Document view.
- 3. In the Document view, click on the toolbar, and choose **Export to PDF**.
- 4. In the **Export to PDF** dialog, set the export options as desired and click **Export**.

#### **Document outline**

The **Outline** panel of the **Document Sidebar** displays a hierarchical tree of the Document's headings, including titles of contained Work Items. The Document outline can help you browse the Document structure to some specific part of a Document. Clicking a node scrolls the Document content to the respective heading or Work Item. When you are editing content in the Document body, the nearest parent heading or title is selected in the **Outline** panel.

If the outline is large you can enter some text in the Filter control at the top of the panel to filter the outline for the input text. If you add or remove content in the Document, you can use the Supdate button to refresh the outline tree.

For more information see the reference topic Document Outline Sidebar.

## **Document actions**

## **Reuse a Document**

## Warning

If a reused Document contains Work Items that have been approved using an electronic signature, the Work Items in new Documents do not show any indication of the source items' signatures, and reused items in the new Document are not approved or electronically signed. These items must be reviewed, approved, and e-signed.

You can reuse an existing Document, or any revision of an existing Document. To determine the number of a revision of a Document to reuse, review the Document history. You can reuse Documents in the same project, in the same space or a different space, or in a different project in any space in that project. You can either create a new, stand-alone copy of a reused Document, or a derived copy in which the text/image content can only be modified from the base Document, but which allows changes to some Work Item fields in the copy - the **Status** field, for example.

You can reuse multiple Documents in a single operation. You can optionally keep relative links between the Work Items contained in the reused Documents.

For example, if you reuse two specification Documents where one contains functional requirements, and the other contains test cases linked to the functional requirements in the requirements specification, you can have the test cases in the reused test case specification linked to the same requirements in the reused requirements specification.

If your system or project is configured to support Document types, a reused Document is of the same type as the original. If your system is configured to use Document workflow, the workflow on new Documents created by reuse is set to the initial workflow status specified in the project workflow configuration.

## Reuse a Document:

- 1. Open the project containing the Document you want to reuse.
- 2. Open the Document you want to reuse in the Document Editor.
- 3. On the Document toolbar, click and choose **Reuse**. The **Reuse Document** dialog box appears.
- 4. If you want to reuse a revision of the Document other than the current head revision, click the button in next to the **Revision** label. In the **Select Document Revision** dialog box, select the **Revision** option and enter the number of the Document revision you want to reuse.
- 5. In the **Title** field, optionally specify a different title for the new Document to be created as a result of the reuse operation.
  - If you modify the Title field, and you want the Title styled heading in the Document updated to that value, select the check box labeled **Update Title (Heading) in the Document**. Clear the box if you want the original title preserved in the Document's title heading.
- 6. In the **Name (ID)** field, optionally specify a different name for the new Document to be created as a result of the reuse operation. If this Document is saved in the same space as the base Document, you *must* specify a new name to avoid a conflict with the name of the base Document.
- 7. By default, the new Document is saved in the same project as the base Document. If you want to save it in a different project, select the target project in the **Project** list.
- 8. By default, the new Document is saved in the **Default** space of the project selected in **Project**. If you want to save it to a different space, select the space name in the **Space** list.
- 9. (Optional) Select the Remove outgoing Work Item links check box. When selected, links from Work Items contained in the base Document to other Work Items in the project (or another project) are not created in the duplicated Work Items in the new Document. You might want to do this if the base Document is a production Document, as opposed to being some kind of library document, or a template for production Documents.
- 10. Select a reuse option.

To create a new, stand-alone, fully editable copy of the base Document, select **Create a new stand-alone copy of the Document**.

If you select this option, also specify whether or not to link the duplicate Document's Work Items to the base Document's Work Items. If you opt to link, select the link role in the drop-down list.

To create a copy of the base Document which cannot be edited (except for some Work Item field values), select Create a new derived Document. (For more information, see Derived Documents later in this section.)

If you select this option, specify which Work Item fields should be read-only in the derived duplicate. By default, the Title and Description fields are specified. You may optionally add other fields, delimiting the list with a comma. For example, if you want the Severity field to be read-only in the derived Document, add it to the **Derived Fields** field.

11. Click **OK** to create the new Document copy in the specified project and space. The new copy opens in the Document Editor and is selected in Navigation. If the new copy was saved to a different project from the base Document, that project is opened before opening the Document.

## **Derived Documents**

Reusing a Document with the option to create a new derived Document enables you to use the first Document as a standard. For example, if you have a standard set of requirements or specifications that must be implemented in every project, you can create one base Document — from which other Documents are derived, and reuse it in projects as a derived Document. A derived Document is an exact copy of the base Document, which cannot be modified other than to set values of some Work Item fields — Status or Severity, for example — if these were not configured as read-only when the base Document was reused. Work Items contained in the base Document are duplicated in the project containing a new derived Document, and assigned new IDs. These items can then be tracked and managed as Work Items in the project containing the derived Document, which may be different from that of the base Document.

The advantage of derived Documents is that if something changes in the base Document — a new requirement is added, for example — the change can be easily propagated to all derived Documents, resulting in the creation of new Work Items, and sending of notifications about them to the Document owner.

There are several important things to keep in mind about derived Documents:

- The derived Document is linked to the source Document.
- Work Items in the derived Document are linked to the same Work Items in the source Document with the is derived from link role.
- If the source Document is changed, you can update a derived Document to reflect the changes.
- If the source Document is moved or renamed, links to all derived Documents are updated automatically.
- If you try to remove the source Document, or the space containing it, you are warned about the presence of derived Documents and offered the choice of canceling the action, or converting the derived Documents to stand-alone Documents. If you choose to convert, the link to the source Document is lost. Links between Work Items contained in the derived Document are preserved.
- If the project is configured to support Document types and Document workflow, updating a derived Document from the base Document does not affect the type or workflow status of the derived Document.

## **Propagate base Document changes to derived Documents**

When the content of a base Document changes, the change may be propagated to any or all derived Documents.

#### Caution

Changes *are not* propagated automatically. This gives users the option *not* to propagate base Document changes to some derived Documents. The owner of the base Document should manually notify owners of derived Documents about changes that they may want to incorporate.

- 1. Open the derived Document in the Document Editor.
- 2. On the Document Editor tool bar, click and choose **Update Document**.

## Tip

You can also update the derived Document using the **Update** link in the Document Properties panel of the **Document Sidebar**.

## Referenced Work Items in a copy

If you copy a Document that contains referenced Work Items, then those Work Items are also referenced Work Items in the copy. If any of these items are frozen to a specific revision of the original Document, they remain frozen to that same revision in the copy. Referenced Work Items show the (Revision) icon. The icon tooltip displays the revision number.

## Mark links as Suspect

When updating a derived Document, there is an option to set the **Suspect** attribute on incoming links. When set, then links incoming to Work Items modified by the update operation are marked as suspect. The user updating the derived Document and setting the **Suspect** option must have permissions to update the linked items, otherwise the update cannot proceed and the user receives an error message.

For example, setting links as suspect would be desirable when updating a derived requirements specification Document that has verifying test cases linked to the contained requirements. Because the linked test cases could be invalidated or broken by the update, flagging the links as suspect helps ensure that owners of the test cases are made aware that changes have occurred in the requirements so that they can review the test cases.

## Reuse multiple Documents in a single operation

Reuse multiple Documents in a single operation can be especially useful where you have a set of standard or boilerplate Documents that need to be reused in multiple projects. The base Documents to be reused can reside in any project and space, and can be reused in any other project and space.

 In Navigation in the project containing the base Documents to be reused, navigate to the space containing the base Documents and select Index. The Index page opens listing all Documents in the space.

- 2. Select the check box on the row of each Document you want to reuse, and then on the toolbar, click **Reuse Documents**. The Reuse Documents wizard appears.
- 3. On the **Selection** page, select the options as desired.
- 4. Next, select any additional Documents you want to reuse. Initially the table of selected Documents contains only the one(s) you selected on the Index page, which means they are from the current project and space. You can add Documents from any project, and from any space in a specified project. Use the 🛨 icon to add a new row for each Document you want to reuse. specifying the project, space in the selected project, and Document in the selected space. If you make a mistake, use the - minus icon to remove the incorrect Document's row in the table. By default, the Head revision of each selected Document in the table is the revision reused. If you want to reuse a different revision of any Document, click Head on the relevant table row and specify the number of the revision you want to reuse.
- 5. Click **Next** to move to the next page of the Wizard. The next steps depend on whether or not you selected the option Use same project and settings for all reused Documents (it is selected by default).
- 6. If you selected Use same project and settings for all reused Documents:
  - a. In the Project list, select the project in which the selected Document(s) should be reused. The current project is the default. A new copy of each Document selected for reuse will be created in the specified project when you finish the operation.
  - b. Provide a name and title for each Document, and specify the space in the target project where each reused Document should be created. Note that each Document's name is the identifier (ID) for the system, and therefore must be unique within the target space.
  - c. Select the desired reuse option (stand-alone copy or derived Document) and specify the options for your selection. Click Next when finished.
- 7. If you did not select Use same project and settings for all reused Documents:

In the Following Steps pane of the Wizard, you see a settings item for each of the base Documents you selected for reuse. So for example, if you are reusing 3 Documents, you see Settings 1/3 Settings 2/3 Settings 3/3. These are settings pages for each of the Base documents on which you can specify different reuse options for each of the Documents.

On the Settings page for each base Document:

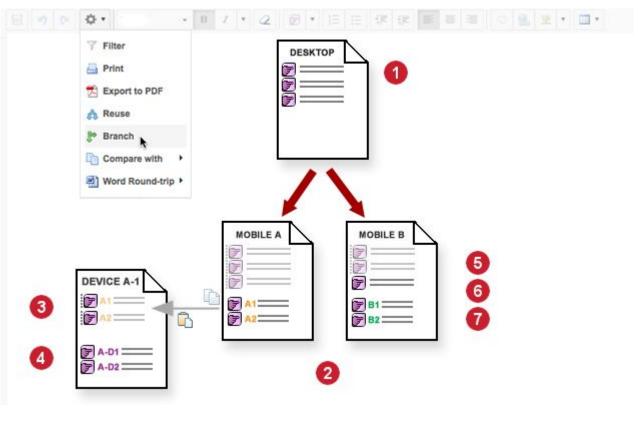
- a. Specify in which project the Document should be reused. The current project is the default. A new copy of the Document will be created in the specified project when you finish the operation.
- b. Provide a name and title for the Document, and specify the space in the target project where the reused Document should be created. Note that each Document's name is the identifier (ID) for the system, and therefore must be unique within the target space.
- c. Select the desired reuse option (stand-alone copy, or derived Document) and specify the options for your selection.
- 8. Click **Next** when finished with the base Document settings page.

After completing the reuse settings for all base Documents, the **Execution** page of the wizard appears, and copies of the base Documents are created according to the settings. When all copies have been created and stored, the page displays a button for each of the new Documents enabling you to open any or all in the Document Editor. Documents open in a new browser instance. When you have opened the Documents you want, click **Finish** to close the wizard window.

#### **Branch Documents**

Polarion supports branching of entire Documents or branching of existing Document-based Work Items into a new Document, in both cases creating a variant Document. As an illustrative use case, consider a scenario where there is an existing requirements specification for some application — a game, for example. The original specification is for the game running on desktop computers. Now there is a decision to create variants of the game for several popular mobile platforms and devices.

Much of the original specification is applicable to all the variants — rules of the game, for example. However, the user interactions are different for each mobile platform, so you need some way to specify platform-specific requirements while keeping those that a applicable to all platforms However, the scenario doesn't end there. Some of the user interactions are the same across all devices running Mobile Platform A, but there must be additional requirements for one or more specific supported devices running Platform A.



1 Master Document	6 Referenced in branch
2 Branched Documents	6 Added in branch

3 Referenced Work Items	Native in branch
4 Native Work Items	

In this scenario, the requirements specification Document for desktop platforms is branched to create variant specifications for two supported mobile platforms — Platform A and Platform B. In each of the branched variants, some requirements from the *master* — that is, the original Document — are referenced, which means that they appear in the variant, but are not contained in it, and may not be modified in it. In the Platform B specification, one requirement from the master is removed from the variant and replaced by a new platform-specific requirement. Then, new native Work Items — that is, items created in the branched Documents — are added to the branched specification Documents to address requirements specific to each platform. The native items are of the User Story type because the mobile development uses the Scrum methodology.

At this stage the specification Documents are Desktop Requirements Specification (master Document), Mobile Platform A Requirements Specification, and Mobile Platform B Requirements Specification. If Work Items in the master Document are modified, the changes propagate to the variant Documents. In cases where this is not wanted, or is not wanted until some future time, the variant Documents can be frozen to a specific Revision of the master Document, meaning that subsequent revisions of the master's Work Items do not propagate to variants until such time as the variants are unfrozen, which means the branch is then live, and all subsequent modifications of the master Document's Work Items will propagate to variants.

Now, the device variants on Platform A need to be addressed. The new native requirements from the Platform A specification are valid for Device 1, but there are some additional ones required to support the device. The approach here is to create a new Document, and then copy all applicable content from Mobile Platform A Requirements Specification and paste that content into it. Any headings and plain text are copied verbatim and changes in the Platform A are not propagated. Copied and pasted Work Items are referenced Work Items — they appear in the Device 1 specification Document, but are not contained in it. Referenced Work Items can only be modified in the Document in which they are native — in this case the Platform A specification.

If the project of the original Document(s) is configured to support Document Types and Document Workflow:

- The Document Type of any branched Document is the same as that of the original Document from which it was branched.
- The workflow is reset for every branched Document. Its status is set to the initial status specified in the project workflow configuration, and any initial workflow action specified in the workflow configuration runs.

#### Variants based on a revision

There may be situations when a variant needs to be based on a specific historical revision (not depicted in the previous figure). For example, suppose requirements for Device 2 need to be based on the Platform A specification as it was at the time of Release 4. Assuming the a Baseline was created at the time of this release, you could open the Baseline, browse to the Platform A specification, copy content from the baseline Document, and paste into the variant Document

(opened in another browser instance). Alternatively, you could open the baseline revision of the Document in History and use the Branch Document feature in the Document Editor.

In the variant Document (Device 1), any of the referenced Work Items can be frozen to a specific revision of the Work Item. If the items are subsequently updated in the source Document (Platform A specification), the changes do not propagate to the branched Work Items in the variant Document until such time as the Work Items in the variant are explicitly unfrozen by a user. Freezing referenced Work Items to a revision would generally not be needed when the source of the items is a historical revision of a Document, because the source Document revision cannot be modified.

#### Create a branched document

# ALM REQ

In order to branch a Document, you must have permissions for creating new Documents and new Work Items in the project where the branched Document is to be created. You can branch the current state of the Document, or the state from a **Baseline** or other historical revision of the Document.

#### Branch the current state

This section describes how to branch the current state of a Document, which is the Head revision in the repository.

- 1. Locate the Document you want to branch and open it for editing.
- 2. On the toolbar of the Document Editor, click and choose **Branch** in the drop-down menu. The **Branch Document** dialog box opens.
- 3. Set the dialog box fields as follows:
  - **Title**: Optionally specify a new title for the branched Document.
    - If you want to branch a revision other then the current one, click the button (labeled **HEAD** or with the number of the revision you are branching) and select the revision from which to branch a new Document.
  - If you have modified the Title field, and you want the heading styled as **Title** in the Document updated to that value, select the check box labeled **Update Title** (**Heading**) in the **Document**. Clear the box if you want the original title preserved in the Document's title heading.
  - **Name (ID)**: If you want to create the branched Document in the same space as the Document you are branching, you *must* enter a new name/ID. The new value must be unique within the space.
  - **Project**: If you want to create the branched Document in a different project, select the target project in the drop-down list. The default value is the name of the current project.
  - **Space**: If you want to create the branched Document in a different space than the one selected in this field, select the target space in the drop-down list. The list contains the names of spaced in the project selected in the **Project** field.
- 4. Click **OK** to create a new branched Document in the specified project and space.

#### Branch a historical revision

- 1. Open the Document to be branched, and click the \( \bigcup \) History icon to open the Document's history.
- 2. Locate the revision you want to branch, and in the **Actions** column click **Show** to open the revision.
- 3. On the Document Editor toolbar, click and choose **Branch**.
- 4. Set the dialog box fields as described in section Branch the current state.

#### Branch a Baseline

The procedure is similar to that described in the previous section on branching a historical revision.

- 1. Open the Baseline in which is captured the state of the Document from which you want to create a branched Document.
- 2. Click **OHistory** on the Document Editor toolbar to open the Document history.
- 3. Locate the revision of the Baseline you are now working. Baseline revisions have a light green background in revisions table, and the **Revision #** column contains the name of the Baseline as well as the revision number.
  - In the row of the Baseline revision, click **Show** in the **Actions** column to open the revision.
- 4. On the Document Editor toolbar, click , choose **Branch**, and proceed as described in Branch the current state.

#### **Branch a filtered Document**

You might want to create a branched Document that references only a subset of the Work Items in the master. For example, you might create a branched Document containing only items having severity Must Have and status Accepted.

To create such a branched Document, you must first filter the master document so that it shows only the Work Items you want referenced in the branched Document. With the master Document open in the editor, click the and choose **Filter**. Enter a query that filters the Document's Work Items. For example, status:accepted AND severity:must have.

When the Document is filtered as you want it, proceed to branch it the same way as an unfiltered Document.

#### **Work with branched Documents**

Once you have some branched Documents, there are various things you may need to do with them. This topic describes a number of commonly needed features and operations:

- Access and read a branched Document.
- Add comments to branched Work Items.
- Compare a branched Document to the master Document.
- Freeze a branched Document to a revision of the master Document.

- Unfreeze a branched Document to update it with changes in the master Document.
- Branch referenced Work Items to make them native to the branched Document.
- Freeze referenced Work Items in the branched Document to some revision.
- Unfreeze referenced Work Items to update them with changes in the master Document.
- Find all Documents that reference a Work Item

#### Access and read branched Documents

# ALM REQ QA PRO REV

You access and open branched Documents just like any other Document. You can browse spaces in Navigation, browse the Index page of any space, or search for the Document.

You can tell if the Document you are reading is a branched one, and from which Document it was branched in the Document Properties panel of the Document Sidebar. The name of the master Document is shown, and the name is a hyperlink that opens the master.

## **Compare branched Documents**

# ALM REQ

You can easily compare any two Documents. A common use case is comparing a branched Document with the master Document (that is, the original Document). You can compare the full content, Work Items, or Work Item fields in different compare views.

- 1. Open a Document that you want to compare with another in the Document Editor. If you want to compare a branched Document to the master, open the branched Document.
- 2. On the toolbar, click the icon. If comparing a branched Document to its master, choose Compare with Master Document. To compare any open Document to any other Document, choose Compare with Other Document.

In every case, you can compare the opened Document to either the current revision of the other Document (HEAD), or an earlier revision. The HEAD revision is the default. To select a different revision, click **HEAD** and use the Revision Picker dialog box to select the desired revision of the Document you want to compare *to*.

#### Select a Compare view

There are three compare views that compare different content of the two Documents. Change the view by clicking the respective buttons on the compare page toolbar.

- **Document**: Displays a comparison of the text content of the Documents, including the text of headings and Work Items. This view is opened by default when you compare Documents.
- Work Items: Displays a comparison of the Documents' Work Items and their structure. The
  Merge side bar is available enabling you to merge Work Items and/or fields between the
  compared Documents. (For more information, see Merge Document Work Items.)

Note that some fields are not displayed in the Work Items compare view even if there have been changes to those fields:

- Author 0
- Comments
- Created
- Module
- Fields: Displays a comparison of the Document's fields, for example, Title, Name, and Branched

## Compare to a filtered Document

The topic Branching a Filtered Document describes how a Document may be branched from a filtered state, so that the branched Document only references the Work Items shown by the filter. If you want to compare the branched Document to the master, you need to specify the same filter query in the **Filter** field of the **Compare Document** dialog box.

In the previous example of filtering the master Document before branching so that it shows only items with status Accepted and severity Must Have, the filter query was status:accepted AND severity: must have. In order to compare the branched Document with only the same Work Items in the master, you would use this same query string in the Filter field of the dialog box.

#### **Export comparison to PDF**

You can export the Document comparison to PDF using the **Export to PDF** item on the Operation menu in the comparison results page. Only the Document comparison view can be exported, not the Work Items comparison view.

An administrator can optionally configure a header/footer for exporting the comparison view. See PDF Export Header/Footer Properties.

#### Compare to the last saved version

On the Document Editor toolbar, the operation **Compare With Last Saved Version** is available. This compares the current state of the Document with the version that you opened in the Document Editor, or to the last saved version if the Document was saved after that.

## Find Documents referencing Work Items

# ALM REQ QA PRO REV

It is possible to have several (even many) Documents that reference a Work Item. For example, there might be a core set of requirements for a product that has 25 different models. There might be 25 branched Documents each containing the specifications for one model, and each Document referencing the core requirements. Suppose you are looking at one specific referenced requirement, and you want to know what other Documents reference it.

1. If you are looking at the referenced Work Item in a branched Document, open it in the Tracker. Click the type icon by the left border and choose **Open Item** on the menu.

On referenced Work Item's tracker page, the Open in Document button on the toolbar displays an arrow symbol. Click the arrow to see a menu listing all the Documents that reference the Work Item.

You can choose any item in the list to open the referencing Document.

#### **Work Items in branched Documents**

When you create a branched Document, the Work Items you see visually are not actually contained in the Document, but are referenced from the master Document. This is visually rendered in the Document Editor by means of a dotted gray left border on the Work Items. Referenced Work Items cannot be edited in the branched Document. If referenced Work Items are exported for Word or Excel round-trip from a branched Document, they cannot be edited in the exported Word or Excel file.

#### **Comment on referenced Work Items**

You can comment referenced Work Items in a branched Document in the same way you add comments to regular Documents (including users of Polarion Reviewer).

With an appropriate product license, referenced Work Items in a branched Document can be commented externally via round-trip for Microsoft Word or Microsoft Excel.

#### **Branch Work Items with Overwrite**

In a branched Document, you can branch referenced Work Items to create a Document-local copy of the items using **Overwrite**. Branched Work Items are contained in the branched Document and no longer referenced from the master Document. Rather, the local copy of each item is linked to the original item in the master Document so that traceability is maintained. If the original item subsequently changes in the master Document, the change does not propagate to the copy in the branched Document. Likewise, changes to the branched Work Item have no effect on the original.

When branching a referenced Work Item, you can create the branched, Document-local copy either from the current (Head) revision of the item, or from any revision of the item in its History.

When overwriting, the modified Work Item uses the original Work Item as a template, but does not copy the values of workflow fields like those listed below:

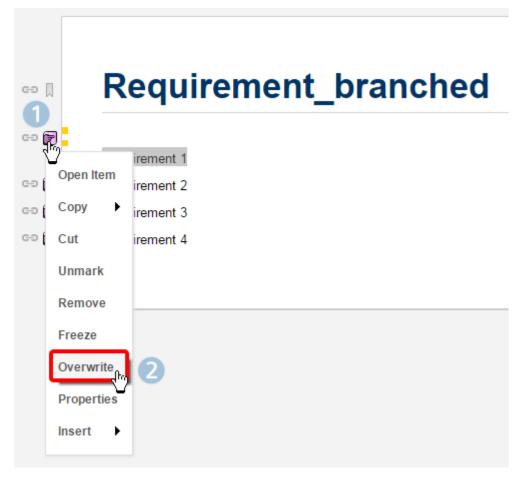
- Approvals
- Author
- Created
- Comments
- Work Records
- Time Spent
- Remaining Estimate
- Status
- Linked Revisions

The fields above can be modified before the newly created Work Item is saved.

(Except **Author** and **Created**. They are populated with the user that did the overwrite and the time and date it was done.)

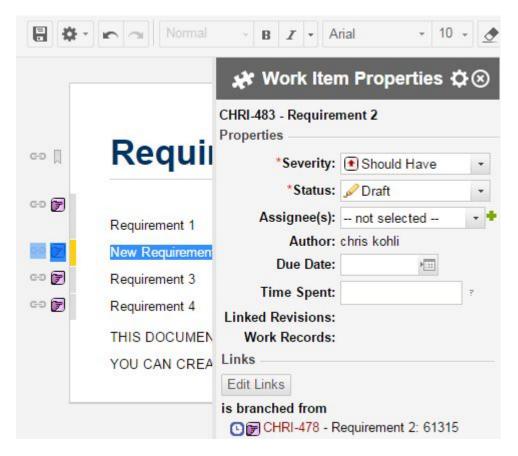
Overwrite a single referenced Work Item in a branched Document:

- 1. Open the branched document containing the referenced Work Item to overwrite.
- 2. Highlight a single referenced Work Item.
- 3. Click the Work Item icon on the left, and then **Overwrite** in the menu that appears.



The dotted line of the referenced Work Item becomes a solid line and available fields are editable in the **Work Item Properties** sidebar.

(Ithen Work Item Properties to launch the Work Items Properties sidebar.)



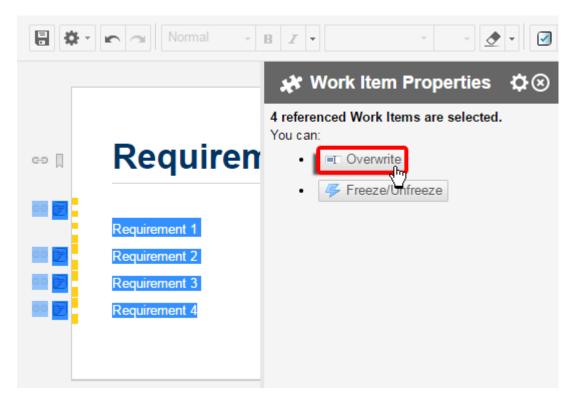
4. Edit the desired Work Item fields and click **Save**.

Bulk Overwrite multiple referenced Work Items in a branched Document:

#### **Note**

The maximum number of Work Items that can be overwritten at one time is 200.

- 1. Open the branched document containing the referenced Work Items to overwrite.
- 2. Highlight the multiple referenced Work Items to overwrite.
- 3. Open the Work Items Properties sidebar. (Click III then Work Item Properties)
- 4. Click on Overwrite in the sidebar.



5. Edit the desired Work Item fields and click **Save**.

#### Tip

If you want to branch a specific revision of a referenced Work Item, you must first freeze it at the revision you want to branch (see next section) and then perform the above steps on the item.

# Warning

#### Overwrite a Referenced Work Item with Restrictions:

When overwriting a Work Item in a generated Variant specification Document, the restriction editor loads the Feature Model from the master project. That is the master specification from which the Variant specification branch was generated. This allows you to merge the overwritten Work Item back into the Master Specification without losing the restriction definition.

#### Freeze Work Items at a revision

You may decide you want to prevent updating of one or more referenced Work Items in a branched Document if the items change in the master Document. In that case, you can freeze any referenced Work Item to a specific revision of the Work Item. The freeze might be only temporary — while development of the master Document is still in progress, for example. When a Work Item is frozen, a small clock icon appears with its title in the Work Item Properties pane of the Document Sidebar.

You can unfreeze any referenced Work Item previously frozen to a revision. Unfreezing causes any changes that occurred in the master Work Item to propagate to, and appear in the branched Document. During development of the master Document, you may freeze previously frozen Work Items at a different revisions throughout the process. Any changes in the new revision propagate to the referenced item in the branched Document, but any changes *subsequent* to that revision will *not* propagate until you either unfreeze the item, or freeze it at a different (and probably later) revision. You can perform the freeze, unfreeze, and revision change operations on multiple referenced Work Items at once.

Note that freeze, unfreeze, and revision change operations affect only the branched Document in which they are performed. Other branched Documents referencing the same Work Items are not affected. For example, suppose Work Item WI-5 is referenced in two branched Documents: BD1 and BD2. The owner of BD1 opens it and freezes WI-5 at Revision 152 (the current Head revision). Later, WI-5 is changed in the master Document. The change is not reflected in BD1 because WI-5 is frozen. It is reflected in BD2 because freezing the item in BD1 only affects that Document.

Freeze a single referenced Work Item at a revision:

- 1. With the branched Document open in the Document Editor, click anywhere on the content of the Work Item you want to freeze.
- 2. Click the type icon at the left border and click **Freeze** on the drop-down menu. The **Freeze Work Item** dialog box opens.
- 3. In the **Freeze at Revision** field, enter the revision number at which you want to freeze this Work Item. You can click the Select Revision icon and use the Revision Picker dialog box to select a revision number.
- 4. Click **OK** to complete the freeze operation.
- 5. After closing the dialog box, save the Document. You can confirm that the item is frozen at the revision you specified by opening the Document Sidebar (if hidden) and then opening the Work Item Properties pane. A small click icon papears by the title, and the icon tool-tip indicates the revision number.

Unfreeze a previously frozen Work Item, or change the revision at which it is frozen:

- 1. Select the Work Item in the branched Document.
- 2. Click the type icon at the left border, and click **Freeze/Unfreeze** on the drop-down menu. The **Freeze/Unfreeze Work Item** dialog box opens.
- 3. If you want to unfreeze the item, select the **Unfreeze** option and click **OK**.
  - If you want to freeze the item at a different revision, select the **Freeze at Revision** option and enter the desired revision number, or use the **Revision Picker** dialog box to select it. Click **OK** to finish.
- 4. After closing the dialog box, click 🖥 Save to apply the changes.

Freeze/unfreeze, or change the revision of multiple referenced Work Items at once:

1. In the branched Document, select the Work Items you want to freeze or unfreeze.

- 2. Open the Document Sidebar and its Work Item Properties panel.
- 3. Click Freeze/Unfreeze.
- 4. In the dialog box, select the option you want to perform on all the selected referenced Work Items. Note that if the selection contains Document-native, that is, not referenced Work Items, the operation ignore them and process only referenced items.
- 5. After closing the dialog box, save the Document to apply the changes.

## Insert referenced Work Items in bulk

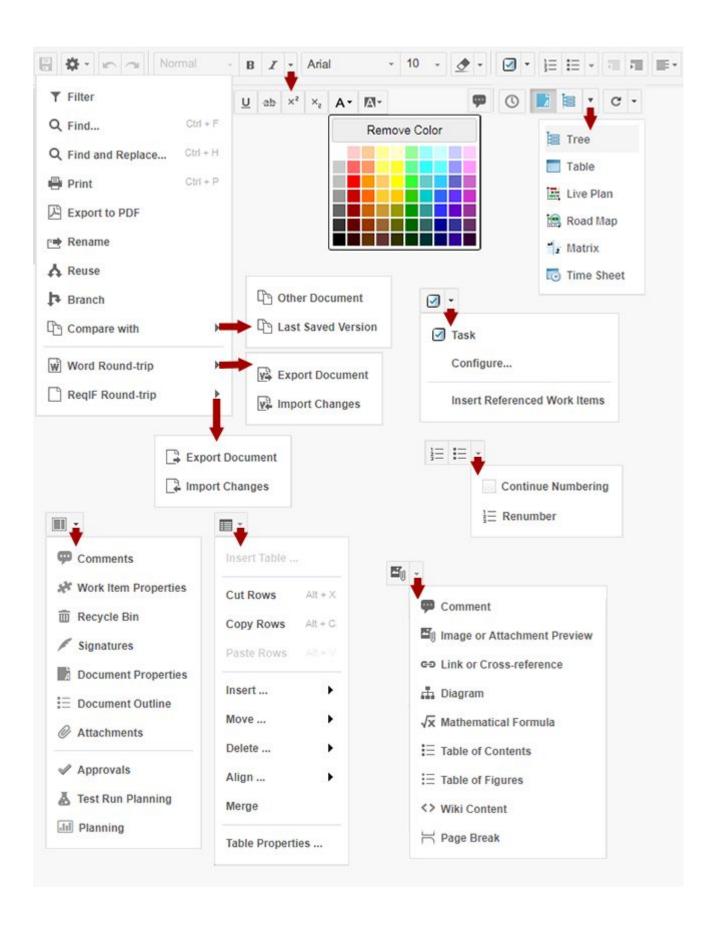
It is possible to insert multiple referenced Work Items into a Document in a single operation. Work Items can be selected in the Tracker in the same way as selection for Bulk Edit, and the references copied and inserted into a Document.

- 1. Open the target Document in the Document Editor, and open the Work Items topic in a separate browser tab.
- 2. In the Tracker, construct and run a query, if necessary, so that all the desired Work Items appear in the guery results in the top half of the page.
- 3. In the table in the top half of the page, select all the Work Items you want to have referenced in vour Document.
- 4. In the toolbar lower half of the page, click (Actions) and select Copy Link and in the dialog box, copy all selected links to your clipboard. Alternatively, you can select Copy IDs.
- 5. Go to your Document and place the insertion point on an empty line where you want to insert the referenced items.
- 6. On the Document Editor toolbar, click the Work Item type icon (tooltip: Mark/Unmark Text as), and on the menu select Insert Referenced Work Items.
- 7. In the Insert Referenced Work Items dialog box, paste the links (or IDs) from your clipboard into the first text field, then click **OK**.

## Trace to linked Work Items

When you have referenced Work Items in a branched Document, you may want to find what other Work Items are linked to them. These linked items are of course external to the branched Document. This is easily accomplished using the Tree view:

- 1. Open a branched Document in the Document Editor.
- 2. On the Document toolbar, select the Tree view (see figure below).



The Tree view shows a hierarchical view of the Document's structure including both referenced Work Items, and items contained in the Document. The toolbar provides a special set of controls enabling you to show linked or backlinked Work Items having a specified link role. You can control the depth of the search, optionally include linked commits (revisions), and optionally filter all levels of the Document tree using the current query. (If Filter Linked Items is not selected, the query is applied only to the first level of the tree.)

#### Referenced Work Item outline numbers

You can enable outline numbering in a branched Document just as you can with any other type. See Using Outline Numbering.

In a branched Document, the outline numbering you see pertains only to the structure of that Document regardless of any numbering that may be applied to referenced Work Items in the master Document. For example, suppose that right after you branch a Document, you have an outline sequence 1.1, 1.2, 1.3, and 1.4 in referenced items from the master Document, Now you insert a new Work Item in the branched Document between referenced item 1.2 and referenced item 1.3. After you save the changes, your new item will be numbered 1.3, the item that was numbered 1.4 will become numbered as 1.5, and the item that was 1.5 will be numbered as 1.6, and so on down the Document content.

#### Move or rename a Document

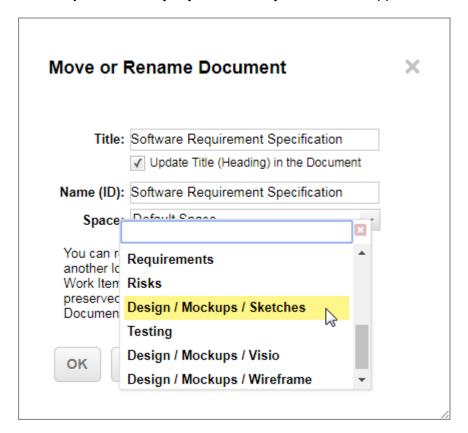
Your can move a Live Doc Document to a different Space or Subspace, either keeping the same name, or renaming it in the process, or you can rename a Document in the same space. You can only move Documents within the current project, and the target space must already exist in order to move a Document to it. You cannot create a new space during the process of moving a Document.

- 1. Open the project to which the Document belongs, if the project is not already open.
- 2. In Navigation, expand Documents and Pages and browse to the Space or Subspace containing the document and select **Index** in that space.
  - If you are not sure where the Document is, you can filter Navigation or use the Search feature to locate the Document and determine the space where it resides.
- 3. In the **Index** page, locate the Document you want to move or rename, and select the check box on its row in the table.
- 4. In the page toolbar, click Move/Rename Document. The Move or Rename Document dialog box appears.
- 5. To rename the selected Document in the current space, type a new name in the **Name (ID)** field, and optionally specify a new value in the **Title** field. The current space is selected by default in the **Space** field. The new name you choose must be unique within the current space. If you try to rename a Document with a name that already exists for a Document in the space. Polarion displays a message and you can try again.

Document name is case sensitive. For example, you could rename a Document My Document as My document.

6. To move the Document to a different Space or Subspace, select the space name in the Space list. If you also want the Document renamed, type a new name in the Name (ID) field (and optionally change Title). The name must be unique within the target space.

The **Space** hierarchy, if you've already created one, appears in the drop-down menu as follows:



7. Click **OK** to complete the operation.

#### Tip

Displaying of Document compare before and after moving may take a long time. The Document before renaming is not cached and the entire history is loaded from the repository.

The Rename action is also available in the Document Properties panel of the Document Sidebar.

# **Change Work Item type**

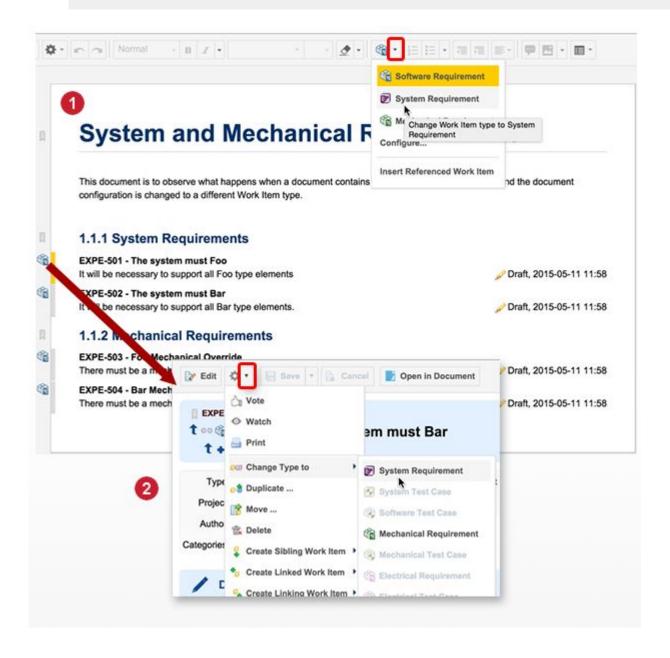
A Document can be configured to contain multiple Work Item types (see Multiple Work Item Types), and types may be added to a Document at any time. When multiple types are configured in the Work Item Presentation, then any Work Item type in the Document can be changed to any other configured type. For example, a Document might initially be configured to contain only Software Requirement type items. Later, it may be decided that the same Document should also contain System Requirement type items, and the Document configuration changed to include that type. Then,

some existing Software Requirement items might be identified as actually being system requirements, and those items changed to the System Requirement type.

The change can be made either in the Document, or in the Work Item Editor (see figure below). Note that when changed to another type, a Work Item displays the fields and other properties that are configured for that type in the Work Item Presentation dialog box.

#### Warning

The Work Item Link Rules in the project configuration are not checked when a Work Item's type is changed. If the changed Work Item has linked Work Items, the links should be checked to ensure that the link roles are still valid for the item's new type. Also, if the item has custom fields, it is possible that some custom field values might be converted during type change, depending on the configuration of the target type.



- **Document** editor
- Work Item editor

Work Item type can be changed via the **Document** editor or the **Work Item** editor.

# Change Work Item type using the Document Editor

- 1. Click anywhere in the Work Item in the Document body to select it.
- 2. On the Document Editor toolbar, click the drop-down control beside the **Mark/Unmark Text** button and select the desired type on the drop-down menu.
- 3. Save the Document to complete the change.

## Change Work Item type using the Work Item editor

- 1. In the Document Editor, click the icon to the left of the desired Work Item and choose **Open Item**. The Work Item opens in the Work Item Editor in a new browser tab.
- 2. On the toolbar, click Actions.
- 3. Hover over **Change Type to**, and on the submenu click the name of the type you want the current Work Item to become.
- 4. Click **Save** to finalize the change.

#### **Print Documents**

You can print a Document to any printing device accessible to the computer you are using. Document comments and comment markers in the Document text are hidden in printed output (except when printing the Compare view in the Document history).

- 1. Open the Document for editing in the Document Editor.
- 2. On the toolbar, click Actions, and on the menu choose Print.
- 3. Wait for the Document preview to load in the **Print** dialog box, and then click **Print**.
- 4. In the dialog box presented by your browser, select the target printer, set any desired print options, and launch the printing process.

#### **Export Documents to PDF**

You can export the content of Documents to Portable Document Format (PDF). During the export operation you can set options for page size, orientation, fitting output to page, bookmark generation, and inclusion of a header, footer and watermark. You can save the generated PDF to any accessible storage (local or network). Document comments, and comment markers in the Document text are hidden in the PDF output (except when exporting the Compare view in the Document history).

Administrators can configure the default content of the header and footer in generated PDF documents, either globally for the repository, or for a specific project. Headers/footers can include such information as the page name, history revision number, creator's user name, the export date, current page number and total number of pages, for example, 7/20. For information, see Configure PDF Export.

PDF header and footer content can also be configured for individual Documents during the Export action. This is an advanced configuration requiring knowledge of XML.

You can prevent pages breaks from occurring within Work Items by enabling the No Page Break in the Document Presentation dialog box, accessible in the Document Editor (click the Work Item type icon in the toolbar). When selected for any Work Item type, Polarion attempts, as much as possible, not to have page breaks occur in Work Items in the exported PDF document.

#### Note

Image or Attachment Previews inserted in a document are preserved when the document is exported to PDF.

#### Tip

Polarion does not clean up HTML within a Wiki content block. If a block contains invalid HTML, even though it may look fine in a browser, it may look different when exported to PDF or Word formats.

**Recommendation:** Use something like HTML TIDY to clean up Wiki block HTML before exporting.

## Export a Document to PDF:

- 1. Open the Document you want to export.
- 2. In the Document Editor toolbar, click and select **Export to PDF** on the drop-down menu. The Export to PDF dialog box appears.
- 3. In the dialog box, select the desired options for the PDF document.
- 4. When you have set all options the way you want them, click **Export**.

## Tip

You can change the settings and quickly export the same Document again until you close the dialog box.

5. Save or open the exported PDF according to options presented by your computer's operating system.

#### **Customize the PDF Header, Footer, and Watermark**

Advanced users who can work with XML can optionally customize the PDF header, footer, or watermark. In the **Export to PDF** dialog box, select the **Include header**, **footer and watermark** check box, and then click the **Configure** link beside it. The **Configure Headers**, **Footers and Watermarks** dialog box appears, containing an embedded XML editor, in which the XML code of the PDF export configuration for the current Document is preloaded. If there is no Document-specific configuration yet, the default XML from the project configuration is loaded, which you can modify and save as the Document-specific configuration.

To include a watermark image, you must add a <watermark> element to the configuration. For detailed information on including watermarks in exported PDF documents, see Configure PDF Export and Configure Export Watermarks.

# **Merge Work Items from different Documents**

## **Basic concepts of Work Item merging**

# ALM REQ QA

There is often a need to have more than one version of a Document in progress concurrently. For example, the requirements specification for version 1.0 of a software application might be completed, and development and/or testing begun, while work begins on the requirements for version 2.0. In such cases, the team can branch the Document containing the version 1.0 requirements specification to create a specification for version 2.0. The Requirements in the version 2.0 Document are referenced from the version 1.0 Document, which is then the master requirements specification Document for the application. In the branched Document, authors can create new Work Items of the Requirement type applicable to version 2.0. While that work is in progress, Requirements contained in the master Document, and appearing as referenced items in the version 2.0 specification, might need to be updated for version 2.0. Authors of the version 2.0 specification can overwrite Requirements referenced from the version 1.0 master with updated content applicable to the 2.0 version.

Then, at some point — a release, for example — it becomes desirable to merge the changed Work Items in the version 2.0 Document into the master Document in order to have the master in sync with the latest release. During the cycle, some items in the master Document may also have changed, and these changes need to be reflected in the version 2.0 document by updating the references and freezing them to the released 2.0 version. The master can then be branched again for a version 3.0 specification, and so on.

The process of branching Documents is covered in Branching Documents. This topic and subtopics describe:

- How to merge Work Items between two Documents, including but not limited to a branched Document and a master Document.
- How to update referenced master Work Items in a branched Document, freezing the reference to the current revision of the Master.

#### Merge actions and terms

Before getting started, it can be helpful to understand what you can merge under what conditions. In general, Polarion does not allow you to run a merge action that is not valid for the target. The user interface only offers valid actions for any merge direction. If you don't see some action you would

expect to find, it may have been disabled in the project configuration. For example, project leaders could decide not to allow Work Items to be sent to the Recycle Bin, and could have an administrator disable the Send Item to Recycle Bin action. In that case, you would never see that action offered in the user interface.

To familiarize yourself with the full set of possible merge actions, and when they are available, see Merge Actions for Compared Documents. It can also be helpful to review the topic on Merge Terminology.

To gain a good background for the merge process, it is recommended to read through the following topics in the order listed below.

- 1. Compare master and branched Documents
- 2. Merge Work Items of compared Documents
- 3. Handle merge conflicts

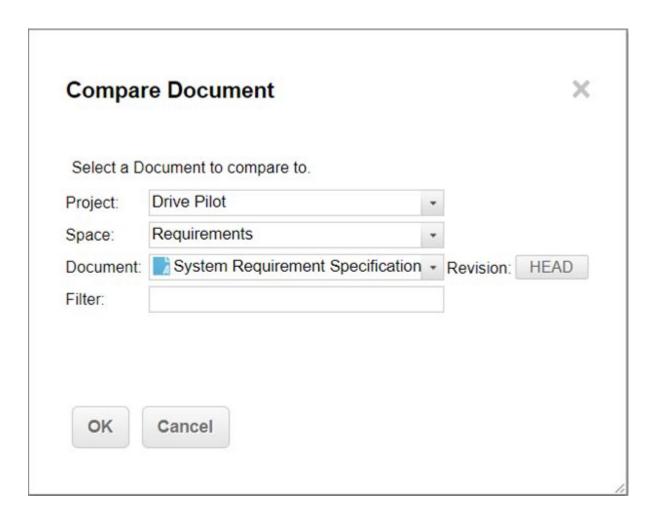
## **Compare master and branched Documents**

The starting point for Work Item merge operations is the **Work Items** page of the compared Documents view.

- 1. Open the a Document you want to compare with another one. This can be master Document or a branched Document.
- 2. If the open Document is a branched Document, click the 🔯 (Actions) icon, and on the menu choose Compare with - Master Document

If the open Document is a master Document, click the [22] (Actions) icon, and on the menu choose Compare with Other Document.

The **Compare Document** dialog box appears.



3. In the **Compare Document** dialog box, select the project, space, and Document to which you want to compare the currently open Document. The head revision of the selected Document is used for the comparison unless you click **HEAD** and select a different revision.

If the open Document is a master, select a branched Document. If the open Document is a branched Document, select the relevant master Document.

## Tip

Advanced users who are familiar with Lucene query syntax can optionally enter a Lucene query in the **Filter** field to limit the comparison to a subset of all the Work Items in the specified Document. Example: status:approved OR status:done

## 4. Click OK.

The Document's compare view now opens provided that the total number of Work Items in both Documents does not exceed the limit set in the system configuration. The default limit is 1000. If exceeded, there is an intermediate step in which you can choose specific things to compare, such as Document, Work Items, or Fields.

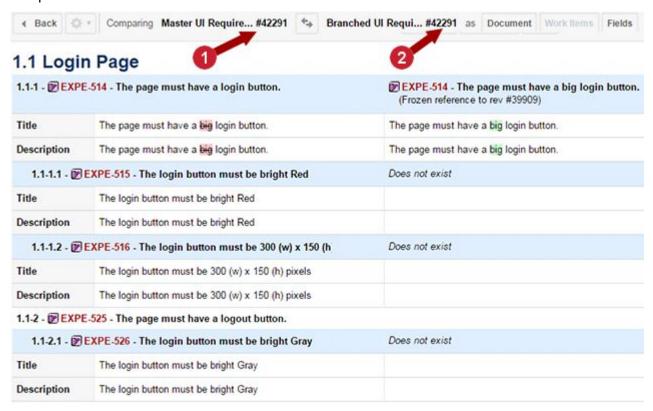
## **Note**

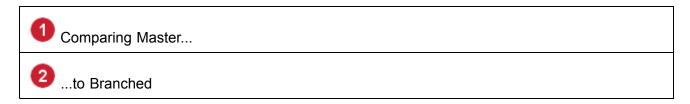
The default limit provides best system performance in most cases. An administrator can adjust the limit up or down in the system property com.polarion.ui.documentCompare.workItemsLimit.

#### **Merge Work Items of compared Documents**

## Review compared items and decide how to merge

When you first enter the Work Items comparison page, it presents a view of the Work Items in the compared Documents:





On the left is the Document from which you invoked the comparison. The Document to which you are comparing appears on the right. For example, if comparing a master Document to a branched Document, the master Document is on the left, and the branched Document is on the right. You can switch the positions of the compared Documents using the Switch Documents icon located on the page toolbar.

Work Items contained or referenced in each Document appear in respective columns. The order of Work Items is presented in the structure of the left Document. For Work Items that are changed, it is easy to see under which chapter they are located and whether there are parent Work Items. Headings and parent Work Items appear only to reveal the structure, and it is not possible to perform merge actions with them. You can easily see the structure of Work Items, and the differences in the two Documents.

Work Items that exist only in one of the Documents that are being compared are also presented in the actual or potential post-merge structure of the other Document. For example, items that exist only in the right Document are shown in the left Document at the place they can potentially be inserted via a merge action

A Work Item that has been overwritten in one Document is compared on single line with the original Work Item it was branched from, or another overwritten item that has same original Work Item.

To better understand why Work Items are compared on one line, as well as for better decision making on merge actions, some additional information is displayed on each Work Item's header:

- Frozen reference to rev # Appears if Work item is referenced from a specific revision.
- Live reference Appears if a Work Item is referenced from its Head revision.
- **Branched from [ID]** Appears if a Work Item was referenced from one Document and was overwritten in the other.
- **Does not exist** Appears if a Work Item does not exist in the Document, and is not in the Recycle Bin.
- Item is in Recycle Bin Appears if a Work Item is not shown in the Document, but exists in the Recycle Bin.

If a Work item has no additional information in the header it means that the Work Item is contained in the Document. That is, it is neither a reference nor a branched item.

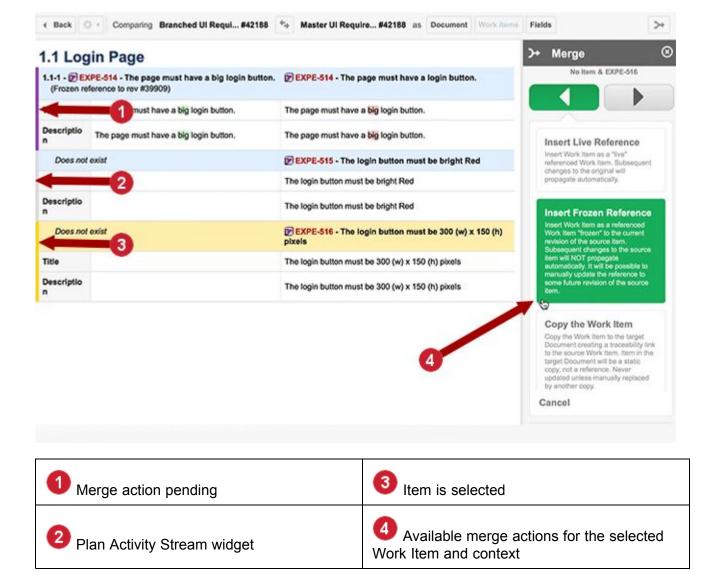
#### Note

Repair missing cross-references

You can repair missing cross-references by inserting the missing item using the **Insert Live/Frozen reference** merge action.

## Merge Work Items using the Merge sidebar

The **Merge** sidebar shows available merge actions for a selected Work Item in a given context, and enables you to perform merges. To access the sidebar, click (**Merge**) on the Work Items page of the Document comparison view. You must have permission to view both Documents, permission to modify any Document changed by a merge action, and permission to modify Work Items in a changed Document. If you do not have the necessary permissions, the sidebar is disabled.



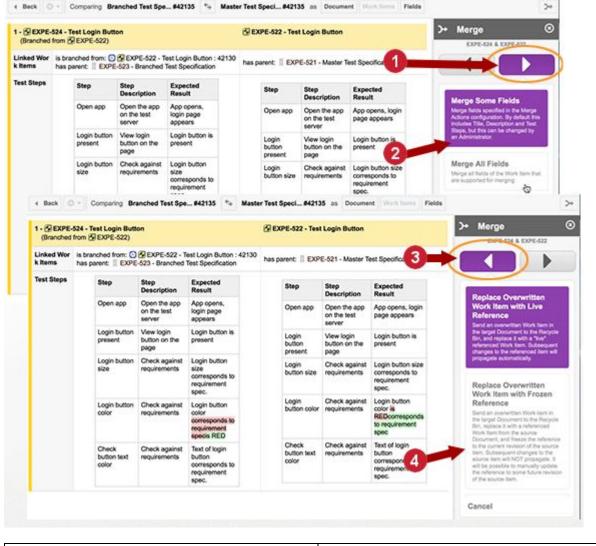
When **Merge** sidebar is open you can select a Work Item by clicking on its header. Work Item headers are light blue. Also, the left border appears highlighted, which helps you to easily see:

- Which Work Item is currently selected: the header of the Work Item and the left border are highlighted.
- Which Work Item has some pending merge action. The left border color changes to the color of the pending merge action in the sidebar. The border color remains if you select another Work Item, until the changes are saved.
- That a Work Item has no pending merge action and is not selected: the left border is gray.

When you select a Work Item, its header row highlights and the Merge sidebar shows the merge actions available for the current selection. (For a complete listing and description of merge actions, see Merge Actions for Compared Documents.)

Also keep in mind:

- The arrow buttons at the top show the merge direction, for example, Master to Branched, or Branched to Master.
- If an arrow button is disabled, it means there are no merge actions available in that direction for the selected item.
- Click an enabled arrow button to see the merge action or actions available in the respective direction for the selected Work Item:





In the Merge sidebar, click the merge action you want to run.

If you make a mistake and find you are looking at the wrong merge action direction or you don't want to do any merge action, click Cancel, or the selected arrow button, or the other arrow button if it is enabled.

You can optionally select merge actions for different Work Items without saving. After you have selected all the merge actions you want for all Work Items, click the Save button at the bottom of the Merge sidebar.

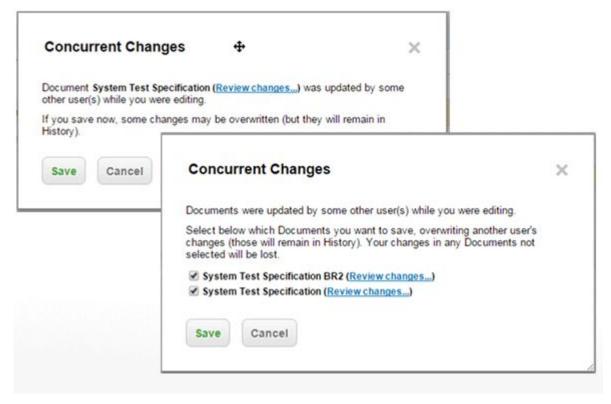
An administrator can configure which merge actions are available to users. For example, if project leaders decide that the Copy the Work Item action should never be available, the administrator can remove it from the configuration. For more information, see Configure Merge Actions.

## Merge Work Item fields

For some merge contexts, the Merge Some Fields and Merge All Fields actions are available for the selected merge direction. The most common scenario is Work Items of Test Case type that contain test steps, but it can be available for other types as well. By default, Merge Some Fields merges only the Title and Description fields. However, more fields can be specified in the Merge Actions settings in the global and/or project configuration. Merge All Fields merges all fields normally needed by users. For a list of fields not merged by this action, Fields Not Supported for Merging.

# **Handle Merge conflicts**

Concurrent changes in one or both Documents involved in a merge action are generally a rare occurrence, but a conflict could happen. Polarion always notifies you if another user has made concurrent changes to the same Document you are working with, and displays messages explaining your options. These include reviewing the changes made by the other user, so you can decide whether or not to keep them or overwrite them.



# **Sign Documents**

# **Background information**

Polarion supports a process of review and approval of individual Work Items, such as requirements. These may be contained in Documents, or created directly in the Work Item Tracker. For more information, see Reviewing and Approving Work Items). However, it can be desirable, for regulatory compliance and other reasons, to have such a process for entire Documents, such as requirements or design specifications.

Polarion can support your formal review and sign-off process for specifications and other types of LiveDoc Documents, using secure electronic signatures compliant with U.S. 21 CFR Part 11. A Polarion administrator can map your review and sign-off process to the Document workflow configuration. Each step in the process can optionally be set up to require one or more electronic signatures before the a Document can move to the next stage in the workflow.

Consider a simple example. Suppose a requirements specification Document goes through three stages before the requirements can be implemented in production: **Draft**, **In Review**, and **Approved**. An administrator can map these status values in the project workflow configuration for Documents of the type **Requirements Specification**. It is possible to configure as many Document types as you need. You might have more granular types like **System Requirements Specification**, **Functional Requirements Specification**, and types like **Risk Analysis**. The Document owner or manager can invoke actions to move the Document through the process from one status to the next. These status values can also be defined in the Document workflow configuration, and could be something like:

- Send to Review This action transitions the specification Document from the Draft status to the In Review status.
- Mark Approved This action transitions the specification Document from the In Review status
  to the Approved status.

#### Tip

Polarion includes project templates that have Document types and Document workflow already configured. The templates are based on input from actual users and are designed to meet the needs of a broad range of projects. Both project templates, and projects created from them can be customized to meet user-specific requirements.

## Sign Documents in an SSO environment

Under normal circumstances, with Single Sign-on (SSO), users do not have to enter their credentials anywhere in Polarion if they've logged on to an associated SSO account.

E-signatures are an exception. Because e-Signatures require that users provide two distinct tokens when signing their electronic signature, an alternative LDAP authentication via a direct LDAP connection needs to be configured. Your Polarion administrator must do the following:

- 1. ( ♣ Administration → ♣ Global Administration → ♣ User Management → ♣ LDAP Configuration) to go to the LDAP Configuration page.
- 2. Make sure that the **LDAP Server Connection Settings** are configured.

For details, see the Configure Single Sign-on topic, and the Polarion installation guides for Windows and Linux, available online as PDF documents on the Polarion support site.)

3. Select the Allow for e-Signatures check box in the Alternative Authentication via LDAP in a Single Sign-on Environment section.

# Alternative Authentication via LDAP in Single Sign-on Environment

Allow for Web Services:	
Allow for e-Signatures:	վետ
	~~

4. Click Bave.

# Regulatory compliance

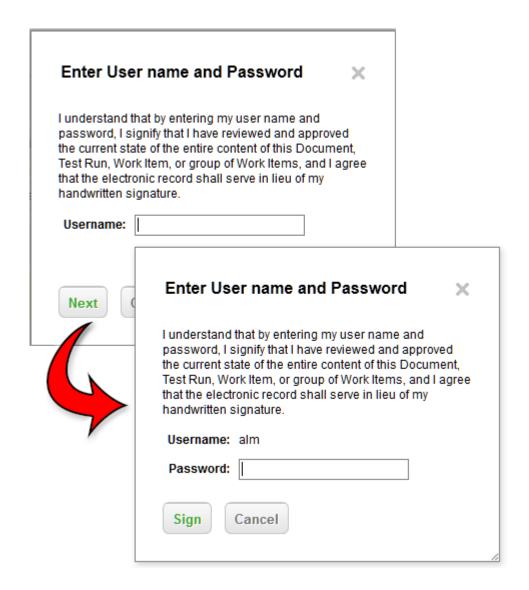
For regulatory compliance, it is necessary to record who reviewed the specification Document, who approved it, and when. The administrator can configure the Mark Approved action to require signatures, and a signature policy such as all invited reviewers must sign, or at least one invited reviewer must sign. When so configured, no one can set a requirements specification Document's status to **Approved** until all the necessary signatures have been registered.

#### Caution

# **Prevent Users from Saving Passwords in Chrome:**

Administrators should ensure that all their users have password saving turned off in Chrome to comply with U.S. 21 CFR Part 11 and other regulatory standards governing electronic signatures.

Electronic signatures for Documents are compliant with United States Code of Federal Regulations, Title 21 Part 11 (CFR 21 Part 11). Users who sign a Document workflow action are shown a dialog advising them of the significance of their signing action, and prompting them to enter their password to confirm their signature and identity. A record of each signature, including data that makes it CFR 21 Part 11 compliant, is recorded automatically in the Document



#### **Process and versions**

To collect signatures, the specification Document's owner or manager invites people to review and sign it by sending them an email containing a link to the Document in the Polarion portal. All invitees must have user accounts and the permissions required to access the Document, and to sign or decline to sign Documents. Each reviewer is presented the option to sign or decline to sign. They can also post comments. After all the necessary signatures have been registered, the Document owner or manager can invoke the **Mark Approved** action in the Document properties, and the Document transitions to the **Approved** status.

The signature process can support multiple versions. For example, the specification may be approved for production, but while production is going on, work resumes on the specification, adding new requirements for new features, or enhancements to existing features. The specification Document having the **Approved** status becomes read-only. However, is not necessary to create a different Document to continue development. The workflow can be configured with an action named something like **New Version**, which branches the approved Document, which can then be tagged with a version identifier and modified as needed for the next round of production. The previous version in

its **Approved** state, is automatically preserved in the Document history. The project manager might also create a Baseline of the project at the point which the specification was signed as approved.

The following topics provide information for different roles:

- Document authors or managers
- Document reviewers and approvers

## Information for authors and managers

This section contains information for authors and/or managers of specification or other Documents. It covers what you need to know in order to set up a review and sign-off process for your Documents.

Start by working with the Polarion administrator for your project to review the Document Types and Document Workflow configurations to make sure the types you need exist, and that each type has your process mapped in the workflow. Decide which transition actions must require signatures, and have each action configured accordingly in the Document workflow configuration. Information for administrators is provided in Configure Document Signatures. Depending on the project template used to create the project, very little if any modification may be needed.

With the Document signatures configuration in place, there is nothing more to do except develop your Document up to the point where it is ready for the workflow transition that requires signatures — or the first such, if there is more than one. For example, your Document starts with a **Draft** status, and at some point is ready for review and approval. Following the simple scenario described earlier, you would invoke the Send to Review action. The action is enabled in the Status field in the **Document Properties** because that action was not configured to require signatures in order for the transition to **In Review** status to take place.

However, you see that the *next* action, **Mark Approved**, is disabled in the Status field. This is because the action is configured to require signatures. You won't be able to invoke that action and transition the Document to the Approved status until signatures are logged. It is at this point that you need to invite people to review and sign your Document.

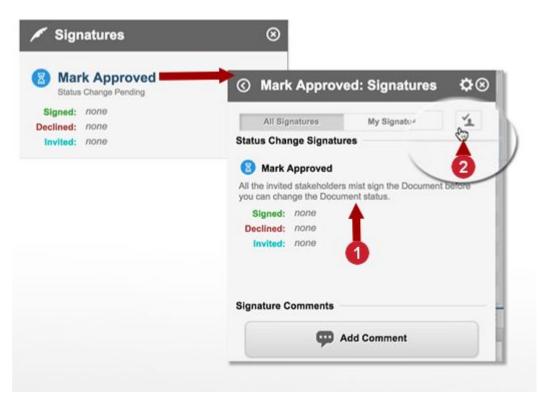
## Invite people to review and sign

Everyone you invite must have a Polarion user account, and have permission to read Documents in the project. Otherwise, they cannot to see your Document and receive an error message if they follow a link leading to it. All reviewers must also be assigned the project role that has been granted the SIGN/DECLINE permission for Documents. Otherwise, they do not appear in the list of people you can invite to review and sign your Document.

#### Tip

User roles, and permissions granted to user roles, are determined by organization or project managers and configured by the Polarion administrator.

- 1. Open the Document in the Document Editor, and open the Signatures sidebar.
  - The sidebar shows a panel labeled with the name of the workflow action that requires signatures. In the case of our running example, that would be **Mark Approved**.
- 2. Click the action name. The sidebar shows two page tabs and a button.



- 1 Signature policy
  2 Invite people to sign
- 3. Click **Invite for Signature**.

In the dialog, select the first user you want to invite from the user names in the drop-down list.

If you want to invite another user, click the ♣icon, select the user in the list, and click ♣ again.
 Repeat until you have added all the users you want to invite.

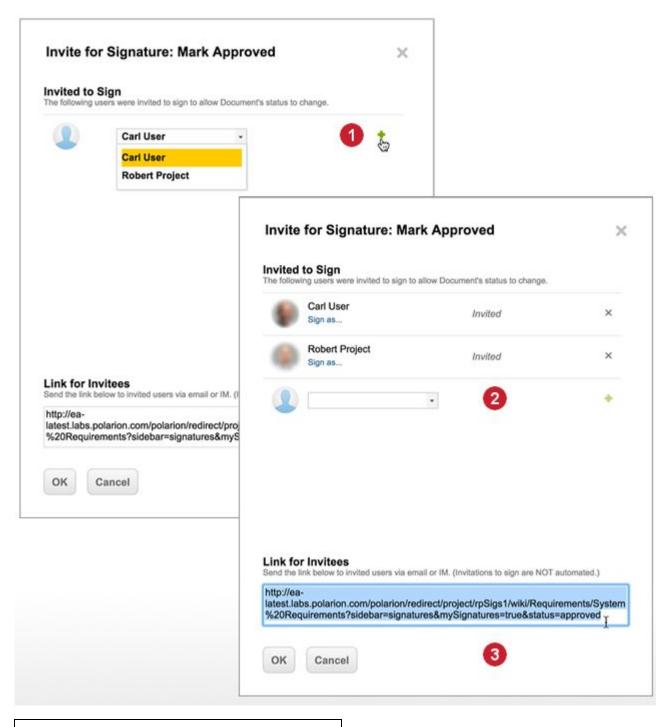
#### Tip

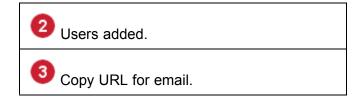
You cannot invoke the transition action until the configured signature policy is satisfied. For example, if the policy requires all invited users to sign, the action cannot be invoked until all have logged a signature. If it requires that at least one must sign and none must decline, then that condition must be satisfied in order to invoke the action.

You can view the signature policy in the **All Signatures** tab of the Signatures sidebar, shown after you click an action name. See the previous figure.

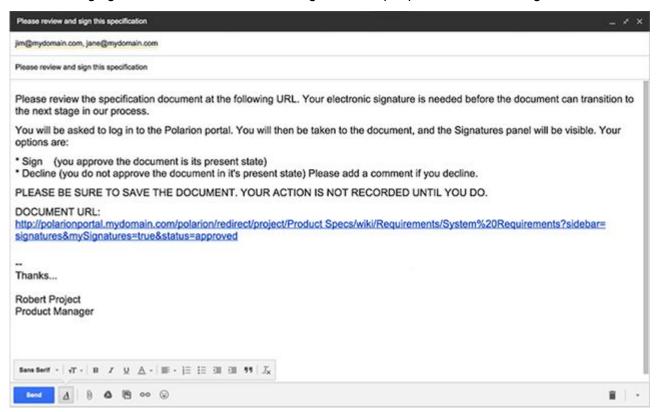
5. At the bottom of the dialog, select the URL in the **Link for Invitees** text box and copy it to your clipboard.

- 6. Click **OK** to close the dialog, and then click **Save**.
- Open your email client program, or internet chat, etc. and compose a message to all the people you invited, asking them to review and sign the document. Paste the URL from your clipboard into the message body.





The following figure illustrates an email message to invite people to review and sign a Document.



When sending a message to your reviewers, include the Document URL. Paste the Document URL from the **Invite for Signature** dialog box.

## **Automate the invitees List**

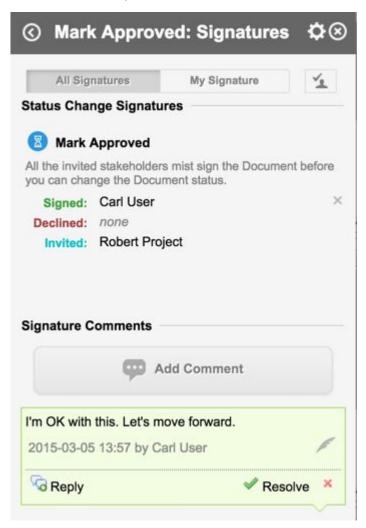
Your administrator can predefine the list of invitees for any Document workflow action, so that the Document owner/manager does not have to add invitees manually for every new Document. Invitees can be added according to their user role in the project, or explicitly by their user ID. You might want to consider having custom user roles defined for your project and have those roles assigned to the people who routinely review and approve specifications or other kinds of Documents.

# **Monitor signatures**

You can monitor the progress of collecting signatures in the **Signatures** sidebar of your Document. The sidebar shows which invitees have signed, which have declined, and the names of all invitees.

The All Signatures tab of the sidebar shows this same information, plus the signature policy.

If the signature policy is currently satisfied, a message displays indicating this. A link is provided that opens the **Document Properties** sidebar. You can then invoke the action to transition the Document in the **Status** field, where the action is enabled. Be sure to save the Document after changing the field.



# Restart the process

Sometimes feedback from invited signers reveals that additional work on the Document is necessary. In that case, another round of reviews needs to be started. The Document owner/manager can reset signature status of Signed or Rejected for the current workflow transition back to the Pending state by clicking on the X icon on the right part of the signature Statistics. The Reset action is available only in the All Signatures panel for a specific workflow transition — the is, not in My Signatures, and not in Entry Point). It is not available after the Document status has been transitioned. You can see recalculated signature statistics immediately without saving the Document.

In cases where a Document needs to be reworked, or a signature was declined, the author or owner may decide that signatures up to that point are obsolete and a new round of reviews and signatures is needed on a new, reworked Document version. To accommodate such scenarios, an administrator can configure the workflow action that sends the Document back to the authoring status — Back to **Draft**) to run a workflow function that marks the current signatures as obsolete.

## Complete the process

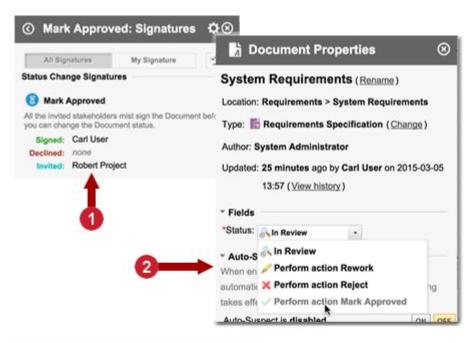
After all signatures needed to satisfy the signature policy have been logged, the Signatures sidebar displays a message indicating that the Document can now be transitioned to the next Status. You as the Document owner, author or manager must explicitly take this action.

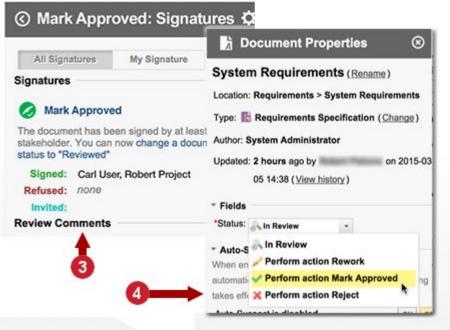
Transition the signed Document:

- 1. Open the **Document Properties** sidebar.
- 2. Click the drop-down control of the **Status** field.
- 3. Select the signed workflow action in the list. The list item should now be enabled.
- 4. Save the Document.

## **Note**

To be able to transition the workflow, you must have the **MANAGE SIGNATURES** permission for Documents, assuming your system has the default configuration of workflow signatures.





As you can see in the above figure, when signature policy is satisfied, the workflow transition action is enabled.

Pending signatures	3 Signatures done
2 Action disabled	4 Action enabled

# Tip

When the signature policy is satisfied, the **All Signatures** tab of the Signatures sidebar displays a link that, when clicked, opens the Document Properties sidebar, where you can change the Document Status.

In projects based on Polarion Project Templates, requirements specification type Documents are preconfigured with Document workflow and permissions settings that make such Documents read-only once they have reached any other than the **Draft** status in the workflow. The prevents modification of any Requirements Specification Document that was submitted for review or approved. Documents can be branched to store the approved version of Document for implementation, while the master specification can be worked on for a subsequent version or release. For technical details, see Document Signatures Custom Set.

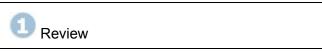
## Information for reviewers and approvers

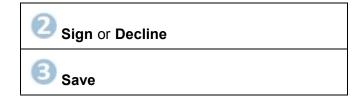
Review this topic of you have been invited to review and sign a Document in your organization's Polarion portal, but are unsure what you need to do.

#### **Access a Document**

Normally you receive an email or other electronic message from the owner of a specification or other Document requesting your review and signature. This message should contain a link to the Document in your organization's Polarion portal. Simply click the link to open your default web browser and navigate to the Document in the portal. If you are not already logged on, you must do so, after which the Document you need to review opens. The **Signatures** sidebar should be visible, and the **My Signatures** tab selected.







If you are already logged on to the portal, Documents awaiting your signature are listed, with links, in your My Polarion page (assuming the page has not been customized to remove this listing).

## Signature options

After you have reviewed the Document, you have three options:

- 1. **Sign** Click the **Sign** button in the **Signatures** sidebar. You can optionally add a comment. By this action you signify that you agree that the Document in its current state can transition to the next step in the workflow.
- 2. **Decline** Click the **Decline** button. You should also add a comment stating your reasons for declining.
  - By this action you signify that you do not agree that the Document in its current state can transition to the next step in the workflow.
- 3. **Add Comment** State questions or discussion points for the author or other stakeholders, without signing or declining.

By commenting only, with no other action, you may be blocking the Document from moving to the next step in the workflow, depending on the signature policy configured for the action you are asked to sign. You can see the signature policy in the All Signatures tab of the Signatures sidebar.

## **Warning**

After taking one of the above actions, be sure to click **ave** on the Document Editor toolbar, or use the Save keyboard shortcut to save the Document. If you leave the Document without saving, your signature action and any comments are not logged.

#### What your signature means

When you sign a Document, you are not necessarily signifying final approval of the Document. You are only signifying that you agree that it can move to the next stage of the development process, as configured in the project workflow. For example, you may have been asked to sign the workflow action **Send to Review**, that transitions the Document's status from **Draft** to **In Review**. By signing, you signify that you agree that the Document is ready to that transition.

Document-related signatures are U.S. CFR 21 Part 11 compliant secure electronic signatures. When you click Sign, a dialog box appears showing a message asking you to explicitly enter your user name, followed by one asking for your password. There are two steps to ensure that the electronic signature is compliant with most standards.



The automated Document history records the data required for regulatory compliance, including your user name and the date-time of your signature.

The action may be configured with a **signature policy** such as **all invited users must sign**. Even if your signature results in the signature policy being satisfied, the action of signing does not actually perform the workflow action and transition the Document to the next Status. The Document author or owner or manager must explicitly do the transition in the **Status** field of the **Document Properties** sidebar.

#### Additional information related to Signatures

## Signature comments

Throughout the process of obtaining signatures for various Document workflow actions, stakeholders can have discussions by posting comments that pertain specifically to the signature process. Such **Signature Comments** have several commonalities with **Document Comments**, which are not related to the signature process:

People can reply to a given comment.

- There can be multiple comment threads.
- Comments can be marked as Resolved.
- The comments appear in the Comments sidebar, and clicking on a comment marker in the text opens that sidebar, focusing the comment.
- The same options are available by clicking icon on the sidebar header.

However, Signature Comments are created in the Signatures sidebar, in the Signature Comments section, using the **Add Comment** button. The comments become part of the history maintained for the Document signature process. The Signatures sidebar shows only unresolved Signature Comments. allowing signature stakeholders to find them easily and not be distracted by Document Comments.

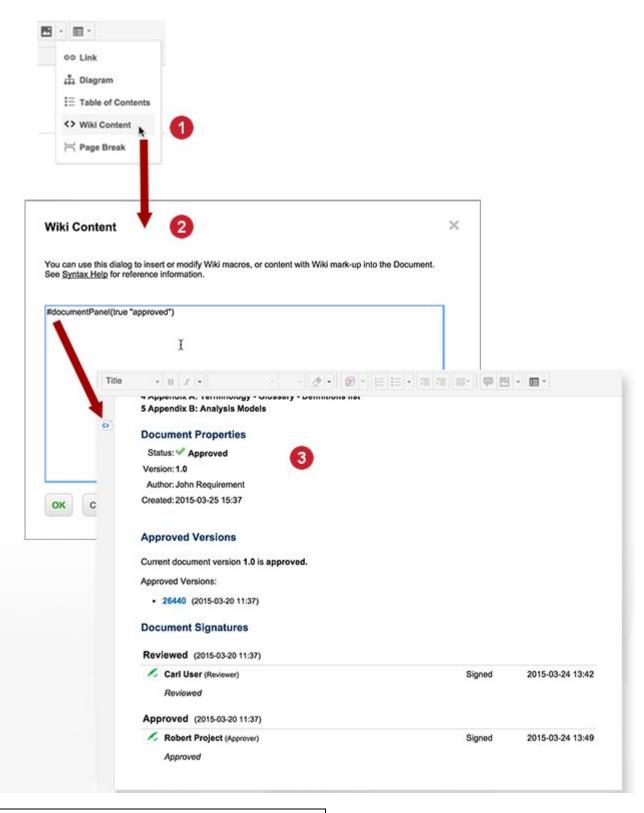
Users must be granted the permissions required to create, reply to, and resolve comments in order to perform these operations.

# Tip

If you have already signed a transition and logged the required signature comment, and the Document owner subsequently resets the signature process, your previous signature comment must be marked **Resolved** before you can sign the document again. If necessary, you are prompted to resolve your previous comment when you sign the Document again.

## Signature information in Documents

You can optionally display a panel of information about the current status signatures in the body of a Document. The #documentPanel macro, when used in a Classic Wiki content block in a Document, renders an overview of Document Properties as well as overview of Document Signatures for the current version of the Document.



Ocument editor toolbar.

- Classic Wiki macro.
- Panel rendered by the macro in the document body.
- 1. Place the insertion point on the line where you want the panel to appear, for example, after the Table of Contents.
- 2. On the Document Editor toolbar, click the Insert drop-down control, and choose Wiki Content.
- 3. In the Wiki Content dialog, enter the documentPanel macro with the desired parameters and click **OK**. For details on parameters, see **Syntax Help** in the Classic Wiki Editor.

Example: #documentPanel(true "approved") - shows Document signature that are not done yet, and specifies the ID of the workflow status representing the final approved status of the Document.

## Tip

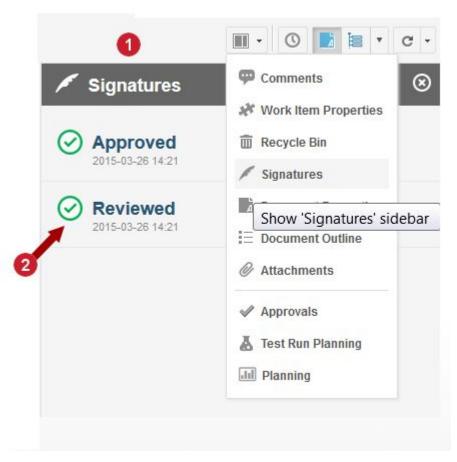
Classic Wiki Syntax Help is normally available on your Polarion portal at a URL like the following:

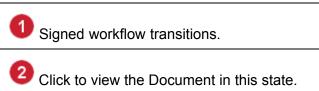
http://[POLARION SERVER NAME].[YOURDOMAIN].com/polarion/#/wiki/Doc/SyntaxHelp

#### View signed revisions in Document history

Changes in Document workflow signatures are shown in Document history when comparing two revisions. Information is shown in the **Fields** section.

When you open the **Signatures** sidebar, it shows all signed transitions — those for which the signature process is complete. Clicking on a name opens a historical view of the Document in the state it was when the transition was signed.





## **Query by Document signature**

This is and advanced topic for developers and power users who want to formulate and run queries on Document signatures. Two indexed fields are available for searching for Documents by signature.

- **signatureState** —The overall state of the active workflow signature; specifically, for the next workflow transition. The goal is to be able to easily query Documents that are in certain state of the signature. Possible values: **pending**, **declined**, or **ready**.
- signatures Search for Documents by individual signatures and their outcomes for the active
  workflow signature; specifically, for the next workflow transition. The goal is to be able to easily
  query for Documents that are waiting for my signature. Possible values: pending, signed, or
  declined.

Syntax: signatures:[VERDICT]=[USERID OR \*]

## **Examples:**

1. signatures:invited=\$me OR signatures:invited=\$[user.id]

The sme variable works only in Classic Wiki content. Dollar sign needs to be escaped in Classic Wiki as \s. Search all Documents where current user is invited to sign the active transition.

- 2. signatures: signed=\* Search all Documents for which the active transition has been signed by any user.
- 3. signatures:declined=bJones Search all Documents for which the active transition has been declined by the user bJones.

### Usage:

Pages macro in Classic Wiki

```
{pages:query=signatureState:pending|display=list} Or
{pages:query=signatures:invited=admin|display=list}
```

- 2. Table Widget in LiveReport Pages.
- 3. Polarion open API.

### **Signatures and Branched Documents**

If you Branch a Document that has signatures configured in its workflow, you can optionally include the signatures. The Branch Document dialog provides an option, Copy Status and Workflow **Signatures** that controls whether signatures are copied. If the option is selected, then on the branch, the action does the following:

- Document status is copied
- All completed (done) signatures and the transition revision are copied, and tagged to show that they are copied from a different Document.
- Any still open signatures are copied and all the signature verdicts are reset. Signature comments are not copied.

Branching with workflow and signatures is only available from a specific Document revision, not the head revision. If you try to branch the head revision, you are automatically switched to the last revision and asked to confirm the switch.

## **Working with Pages**

## Working with Pages overview

Pages are mostly what the name implies: web pages. However, they are only accessible to users who are logged in to your Polarion system. Pages differ from Documents in that they do not contain Work Items like requirements and test cases. However, Pages can show information about Work Items and other artifacts in your projects and repository.

When you create a new Page, you can choose one of the following:

**LiveReport Page**: Page with a layout for a report with charts that retrieve, aggregate, and/or format Work Item and other project data. You build these pages visually using a set of

user-configurable charting and other Widgets, and the rich-text formatting tools provided in the Page Designer.

• **Info Page**: Page with a layout for any kind of free-form textual information, such as guidelines, internal documentation, and guidelines. You build these pages using the rich-text editing tools provided in the Page Designer. Images and tables are supported.

The main difference in these types is in the page layout. You can use Widgets in both types. Some page features, such as the ability to extend Widgets with scripting or create reports, may be limited or unavailable with some product licenses.

When creating a new Page, the **Classic Wiki Page** option is also offered. These pages use the product's original Wiki technology. It is recommended that you use LiveReport and Info Pages for new reports and informational pages. For information on working with existing Classic Wiki pages, see **Using Classic Wiki**.

### **Page Basics**

The following main features and tasks are common to both types of Pages (LiveReport and Info). For information specific to LiveReport Pages, see Build Reports.

- · Creating new Pages
- Accessing existing Pages
- Deleting Pages
- Editing Pages
- Exporting and printing Pages
- Downloading and Reusing Pages

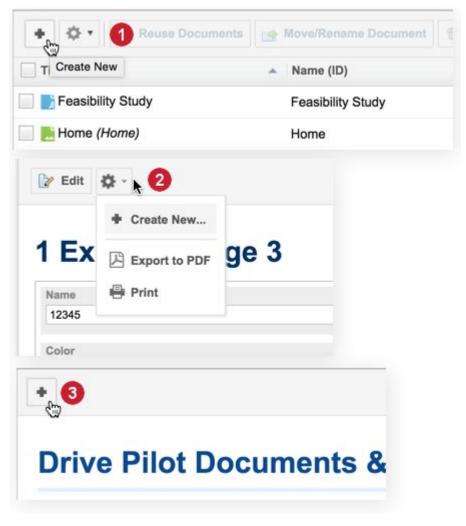
## **Create new Pages**

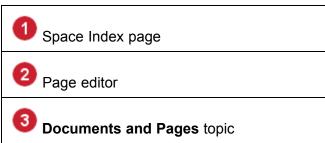
## ALM REQ QA PRO

The ability to create new Pages is controlled by user permissions settings in global and project Administration. You must be granted Permission to CREATE NEW and Permission to MODIFY for Pages in all projects in which you want to create new Pages.

Pages exist within spaces. You can create new pages in any space. There are several access points where you an begin creating a new Page:

- Index page of any space. Click on the icon on the toolbar. (Tooltip: Create New)
- Page toolbar. Actions Menu: Create New, when a page is in view mode. Click to reveal.
- Documents and Pages toolbar. Select this topic in Navigation, and then click on the icon on the page toolbar.

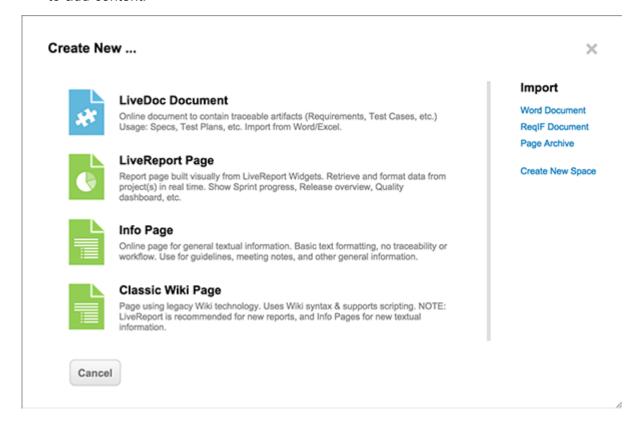




## Create a new Page

- 1. Launch the Create New dialog.
- 2. Click on **LiveReport** Page or **Info Page**, depending on the type of new page you want to create.
- 3. In the next dialog, enter the title of the new page (any character set), the system identifier in the Name (ID) field (ASCII only), and optionally select a different Space or Subspace to store the page. Note that the Title appears in Navigation, search results, links, and other user interface components that show the Page.

4. Click **OK** to finish and create the Page. The new Page opens in edit mode, ready for you to add content.



## **Access existing Pages**

# ALM REQ QA PRO REV

The ability to view Pages is controlled by user permissions settings in global and project Administration. You must be granted Permission to READ for Pages in all projects in which you want to view Pages.

You can access existing Pages in two ways: browse or search. These work the same way for Pages as for other types of content.

#### **Browse for Pages**

You can browse for Pages in Navigation. Expand the **Documents and Pages** node, then expand the node of any space. The Documents and Pages in the space are listed in alphabetical order and the content type is indicated by the icons (see the Create New dialog).

If there are too many items to show in Navigation, a message appears in the panel under the space name. You can either use Search or click on the message to load the space Index page, where you can browse the Pages and Documents in the selected space.

#### **Search for Pages**

The Search field of Navigation enables you to search your project or your entire portal for all types of content, including Pages. For more information, see Search the Portal.

## **Delete Pages**

# ALM REQ QA PRO

The ability to delete Pages is controlled by user permissions settings in global and project Administration. You must be granted Permission to DELETE for Pages in all projects from which you want to delete Pages.

- Locate the Page by browsing or searching to determine the space where it is stored.
- Open the space Index page, expand the space name in Navigation, and click Index. If you see a message that there are too many items to show, click to message to load the space Index page.
- 3. In the space Index page, check the box on the row of all Pages you want to delete, and click Delete.

## **Edit Pages**

### Modify a Page

The ability to edit Pages is controlled by user permissions settings in global and project Administration. You must be granted Permission to MODIFY for Pages in all projects in which you want to edit Pages. This section covers the editing features common to both LiveReport and Info type Pages. Information about editing features for developing reports are covered in the Build Reports topic.

- 1. Locate the page using browse or search, and open it for viewing.
- 2. At the top of the page, click **Expand Tools**.
- 3. On the Page toolbar, click **Edit** to open the Page Designer.

#### **Modify Page layout**

# ALM REQ QA PRO

A Page is composed of one or more *regions*. Regions are simply containers for Page content. You can modify the default layout of a Page to have whatever regions you need to show your content the way you want it. You can select a Layout as well as add, remove, split or resize regions.

#### Select a layout

Each Page type has a default layout:

- Info Page has one region occupying the full width of the Page.
- LiveReport Page has three regions: one occupying the full width of the Page, and two occupying half the width of the page.

When you first create a new page, you may want a layout different from the default. Several pre-defined Page layouts are available, which may provide the layout you want, or at least a good starting point for your custom layout. You can also apply one of the pre-defined layouts to an existing page. The Page content is preserved, but it may not appear where you want it in the changed layout. You can either copy content to a different region, or use the **Undo** button to revert the change. (Redo is available as well.)

You can access the pre-defined layout selectors in the **Layout** section of the **Page Properties** sidebar. The options are:

- Three regions: Top one at 100% of page width, two below at 75% and 25% of page width.
- Three regions: Top one at 100% of page width, two below at 50% of page width.
- Three regions: Top one at 100% of page width, three below at approximately 33% of page width.
- One region at 100% of page width.

You can also specify the minimum width of the Page in this sidebar. The width is in pixels, and the default value is 1000. In view mode in a browser, it sets the minimum width of the Page. If there is some wide content such as an image that cannot be scaled, this setting can increase minimum Page width so that the content is completely visible in the parent region.

In PDF export, the Minimal Width value is used to define page width of the PDF document, providing the possibility to preserve the layout. If wide content is included in an insufficiently wide page, it will be scaled (zoomed out).

### Add or remove regions

You can add a region to the page by placing a new one, or splitting an existing one, using the following commands on the header of an existing region:

- Add Region Above: Adds a new region above the current region.
- Add Region Below: Adds a new region below the current region.
- **Split Region**: Splits the current region into two, moving the current region, including its content, to the left and adding a new empty region to the right. Content of the original region may be scaled to fit the changed width.

You can remove a region from the page using the **Remove Region** command on the drop-down menu of the region's header.

#### Resize a region

If you have multiple regions next to each other in a column-like arrangement, you can resize them horizontally by dragged the separator that appears in the space between the regions. Vertical size of regions is determined by their content and cannot be changed manually.

#### Format text

# ALM REQ QA PRO

Pages support basic rich-text formatting. If you have used a desktop or online word processing application, the rich text tools of the Page Designer will seem quite familiar and work as you expect. If you have worked with Documents, the same text formatting tools are present on the toolbar when editing a Page. The following text formatting options are available:

- Paragraph, title, and heading level formats
- Bold, Italic, underline, strike-through, subscript, superscript, text color, and text background color
- Font family and font size
- Numbered list and bullet list
- Indent and outdent
- Paragraph alignment: left, center, right

# **Title**

## **Heading**

Bold Italic underline strike superscript: 1 subscript: 2 red text yellow background

Arial Courier New Georgia Times New Roman Verdana

10 pt. 12 pt. 14 pt. 16 pt.

Numbered List:

- 1. First item
- Second item
- 3. Third item

#### **Bullet List:**

- First item
- Second item
- · Third item

Indented paragraph

### Change the ordered list style

Administrators can change the ordered list style by adding or editing the com.siemens.polarion.document.listStyle= x property in the polarion.properties file.

(Where "X" can be a series of up to 9 values representing each tier of a Multilevel List.)

For example, com.siemens.polarion.document.listStyle= 1ai.

The property is enabled by default with the above setting.

The default ordered list appears as follows:

```
1. Level 1
2. Level 1
a. Level 2
b. Level 2
i. Level 3
ii. Level 3
1. Level 4
2. Level 4
```

If the property is commented out, as follows:

```
# ·Sets ·the ·numbered ·list ·style ·that ·will ·be ·applied ·throughout ·the ·entire ·application
# ·possible ·characters ·are:
# ·"1" ·for ·decimal ·numbers ·"1, 2, 3"
# ·"a" ·for ·lowercase ·letters ·"a, b, c"
# ·"A" ·for ·uppercase ·letters ·"A, B, C"
# ·"i" ·for ·lowercase ·Roman ·numbers ·"i, ii, iii"
# ·"I" ·for ·uppercase ·Roman ·numbers ·"I, II, III"
# com. siemens .polarion .document .listStyle=1ai
```

Then, the ordered lists only contain numeric values like the example below:

```
1. Level 1
2. Level 1
1. Level 2
2. Level 2
1. Level 3
2. Level 3
1. Level 4
2. Level 4
```

The com.siemens.polarion.document.listStyle property supports the following values:

- "1" for decimal numbers "1,2,3"
- "a" for lowercase letters "a,b,c"
- "A" for uppercase letters "A,B,C"
- "i" for lowercase Roman numbers "i,ii,iii"
- "I" for uppercase Roman numbers "I,II,III"

#### **Enable Outline Numbering**

Outline numbering of headings in Pages is disabled by default. To enable it:

- 1. Open the page for editing in the Page Designer.
- 2. Choose **Show Sidebar** Page Properties. The Page Properties sidebar appears.
- 3. In the **Outline Numbering** section, click **ON**.

#### 4. Save the Page.

Outline numbers for headings are not visible in the Page Designer while editing a Page. They appear only in view mode. Outline numbers are only applied to headings specified in the user interface. Headings created via a script (using, for example, <H2>) will not have outline numbering applied.

### **Copy and Paste**

Copying and pasting text within a Page is similar to other applications. There are several things applicable with Pages that may be worthwhile to note:

- Within the same page you can copy and paste the following:
  - Block of text to the same or a different region
  - Chart or table widget to the same or a different region
  - Links to a different region (relative links remain relative) 0
  - Region to another region 0
- Between different Pages you can copy and paste:
  - Block of text 0
  - Chart or table widgets
  - Links: Relative links remain relative.
  - Region 0
- From an external application to a Page you can copy and paste:
  - Block of text
  - Links 0
- From a Page to an external application you can copy and paste:
  - Block of text
  - Links O
  - Widgets: If text and widget is pasted to an external page, only text is pasted.

#### Note

#### **Copying Work Items within the Same Document**

When On-demand Work Item Loading is turned **ON**, make sure that all the Work Items you wish to copy have finished loading before pasting them somewhere else within the same document.

(This limitation does not apply when copying them to a different document.)

#### **Insert content**

# ALM REQ QA PRO

In addition to text, you can insert several other types of content into any region of a Page:

- Images
- Attachments
- Diagrams
- Formulas
- Tables
- Links
- Table of Contents
- Widgets

### Insert image or attachment preview

The button is available on the toolbars found on Pages, LiveDocs and Rich text Work Item fields. It launches the Insert Image or Attachment Preview dialog.

The dialog initially displays previews of images and attachments that are already attached.

It's also where new images or attachments with previews can be added.

#### Attachment previews for file types other than Images

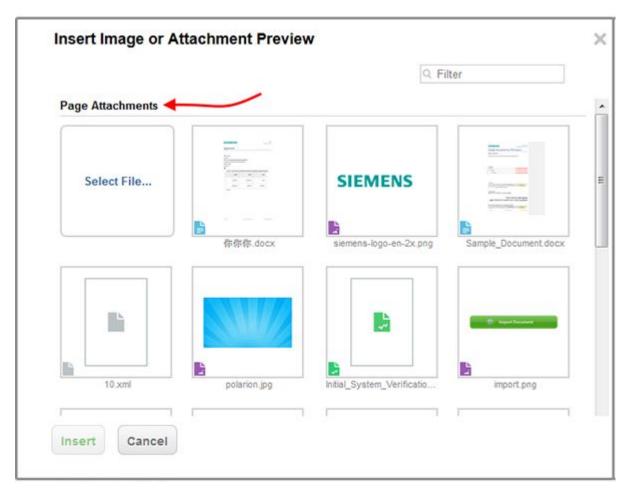
Polarion supports image and attachment previews for all common image formats. To view previews for Visio files and the popular document formats (.pdf, .docx, .xlsx) described below, Teamcenter Visualization Convert & Print Setup must be installed.

(See *Polarion Preview Generator* in the Windows installation document or *Configure attachment preview generator* in the Linux installation document for details.)

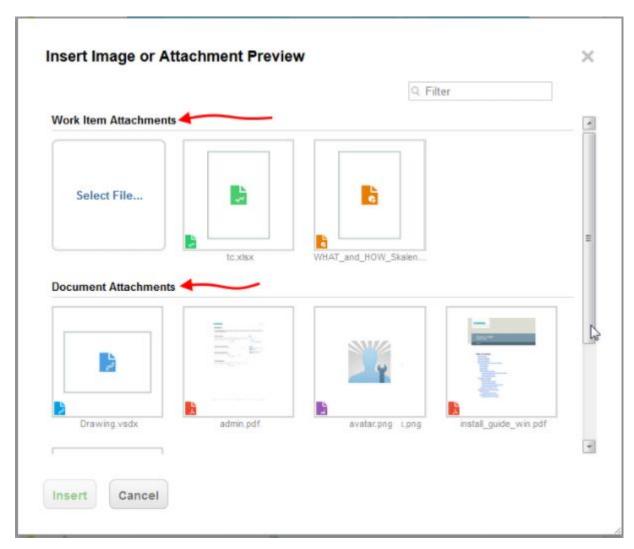
#### Add a new image or attachment

1. On the toolbar, click

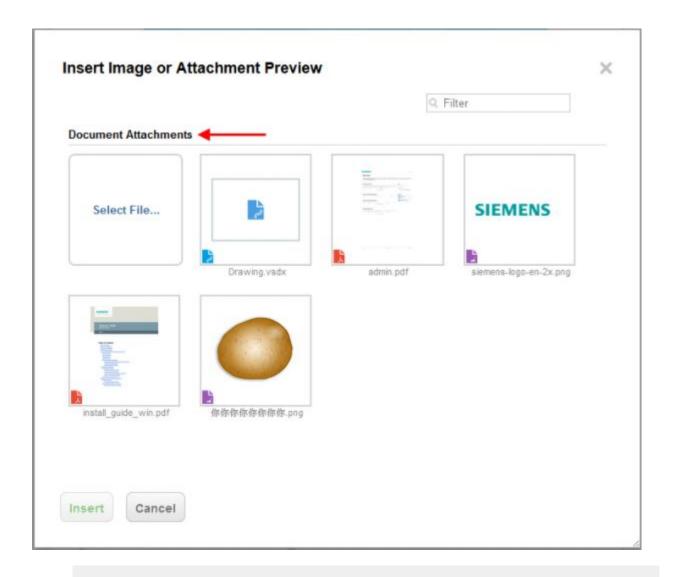
When you have images or attachments on a page (like a LiveReport page), only the images and attachments on the page itself appear. The images or attachments within the Work Items that appear in the page's widgets do not.



When you have images or attachments in both a LiveDoc and Work Item and the **Insert Image or Attachment Preview** dialog is opened inside a Work Item, both sections appear.



When you have images or attachments in both a LiveDoc and Work Item and the **Insert Image or Attachment Preview dialog** is opened from inside the LiveDoc, only the Document's images and attachments appear.

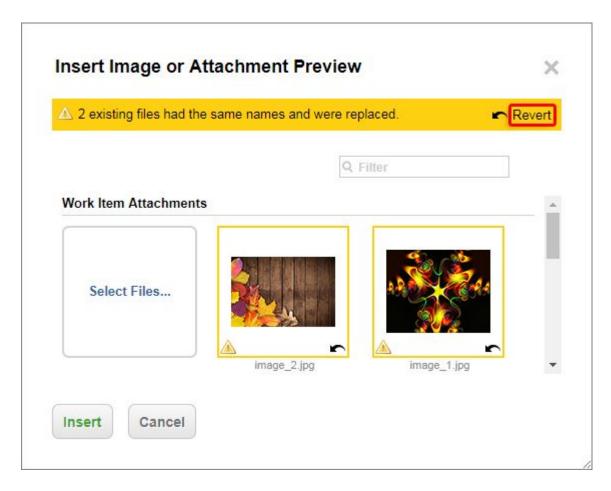


#### **Note**

Hover over an image or attachment preview in the **Insert Image or Attachment Preview** dialog to see its file name and title (if any).

- 2. Click Select File.
- Select one or more images or files to attach and click Insert.
   To select multiple images or files, press Shift or Ctrl, and then select the files to Insert files at once.
- 4. If items with the same file names already exist, you will be prompted with the following message:

  (When uploading a file with the same name as one already saved within a Work Item, the **Revert** option is not available and both files are saved with the same name.)



- 5. Click **Revert** to revert to the original files. Click **Insert** to add the ones you selected and confirm the overwrite.
- 6. Selected items are added and their previews appear in the Rich Page, LiveDoc, or Work Item.

Images and Visio diagrams with a width larger than 750px are resized, with their aspect ratio maintained, to a width of 750px.

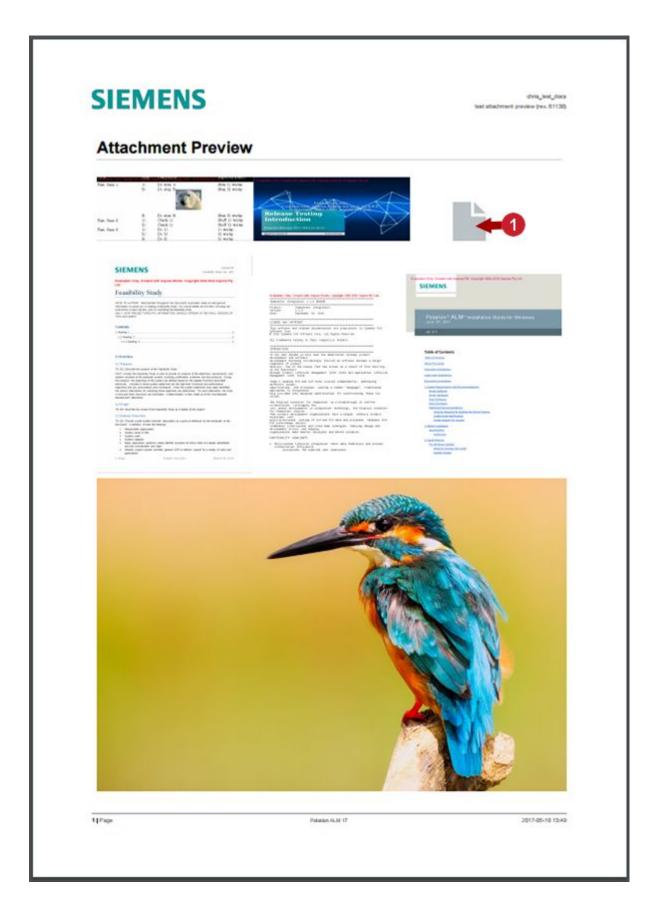
(The width of the page when inserted into a Live Document.)

The previews of common document formats (like .docx, .pdf, .xlsx, and .txt) are resized to a width of 250px and take up one third of a Live Document page's width.

For less common file types, where previews cannot be generated, an icon is added instead. (See the image below.)

The Attachments are added to the Attachments sidebar and their previews are also visible when the document is Exported to PDF.

PDF output:



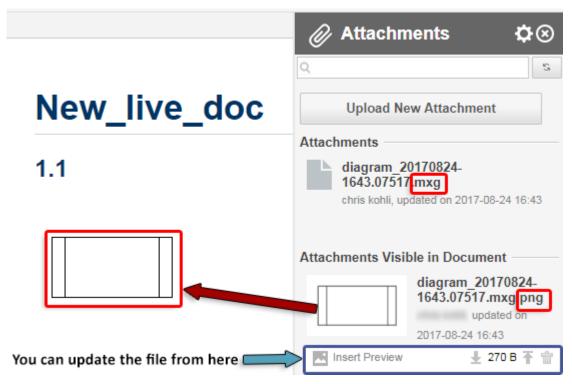
( lcon for less common file types.)

They appear even when the images or attachments themselves have not yet loaded, if On-demand Work Item Loading is **ON**.

7. (Optional) Filter the dialog with the **Filter** field.

## **Attaching Polarion Diagrams**

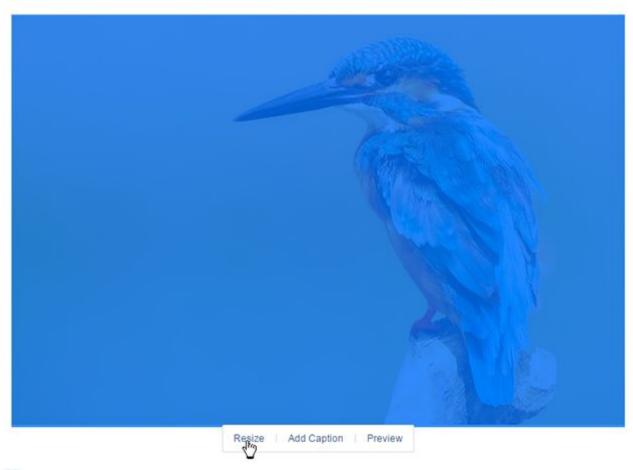
You cannot insert a Polarion diagram itself (the .mxg file) via the **Insert Image or Attachment Preview** dialog box. Instead a preview of the diagram will be added and the original .mxg file will be listed in the **Attachments sidebar**.



See the diagram section to create and edit diagrams directly into a Document, Page or Work Item text field.

#### Resize an image or attachment preview

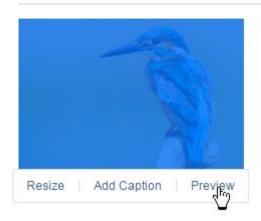
Select the image you wish to resize, and click Resize.



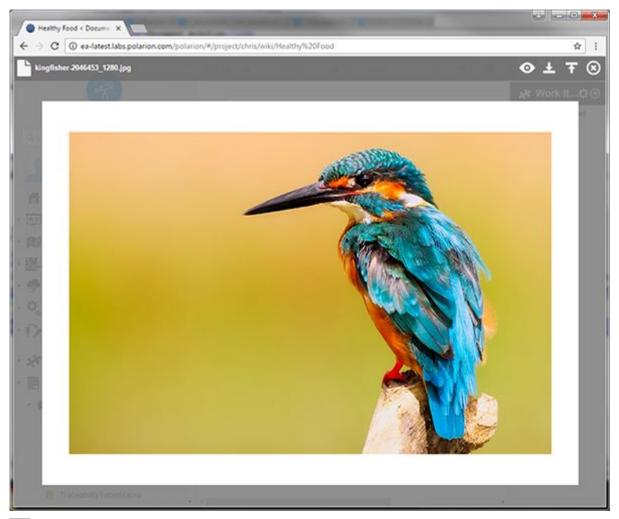
- Fit To Width: Resizes the image width to the document width. (Not available for Work Item Rich Text fields.)
- Reset Size: Resets the image size to its original dimensions.
- **Keep Aspect Ratio**: Active by default. Ensures that the height and width maintain the same ratio when resized.

## Preview an image or attachment

# Birds of a Feather



- For a full screen preview of resized images or attachment previews in **Edit** mode, click on the image or attachment, and then click **Preview**.
  - (Or double-click on the image or attachment preview.)
- When not in Executive Edit mode, a single click will launch a full screen preview.
- Click ESC to exit the full screen preview screen.



- View: Launches the object in its own browser window or the default program configured by the OS to open the selected file type.
- **Download**: Downloads the object to the browser's download folder.
- **Update**: Upload an updated version of the previewed object. (Disabled for read-only documents or for users with read-only permissions.)
- Olose: Closes the preview window. (ESC also closes the window.)

Inserted images, diagrams, or files become attachments of the Page, and remain attached even if visually removed from the Page content. To delete an image attachment completely, see Delete Attachments.

## **Note**

Adding URL links to images is not supported.

## **Pasting images**

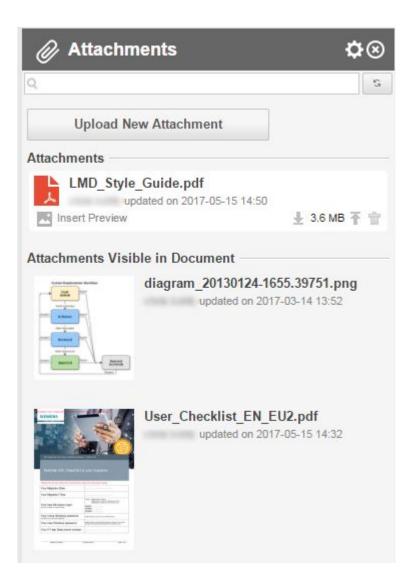
CTRL +V or Command +V can be used for images within the document, but if they are used for an image in a Work Item within the document, its thumbnail will not appear in the Attachments sidebar.

## Insert and manage attachments

Rich pages, Live documents and Work Item rich text fields can have almost any type of file as an attachment. Once an attachment has been uploaded, a link to it or a preview of it can be inserted directly into the document. Or it can simply be appended to the document by leaving it in the **Attachments** section of the **Attachments** sidebar.

#### Access attachments via the Attachments sidebar

- 1. Locate the desired Page by browsing or searching.
- 2. Click on at the top right and click Attachments.
- 3. The **Attachments sidebar** appears listing the **Attachments** currently attached to the document and the **Attachments Visible in Document**.



#### Add a new attachment

- 1. Open the Attachments Sidebar.
- 2. Click **Upload New Attachment**, select the file to attach, and click **Open**.

The selected attachment appears under the **Attachments** section of the **Attachments** sidebar.

- Save the document. 3.
- 4. If the attachment is an image, Visio diagram, or popular document format, a preview also appears within the **Attachments** sidebar. (An icon appears for less common formats.)
- 5. (Optional) To insert a preview of the attachment into the document, place the cursor at the desired spot within the document and click Insert Preview.



See Add a New Image or Attachment for details on how images and attachment previews appear.

### Update an existing attachment

Suppose you have a PDF file attached to a Page, and at some point you find there is an updated version of the same document, with changed content. You can upload the updated version to replace the existing attachment. The uploaded file must be of the same type and have exactly the same file name and extension.

- 1. Open the Attachments sidebar.
- 2. Locate the attachment you want to update. If there are many attachments or some attachments are hidden, use the sidebar's search field.
- 3. Hover your pointer over the attachment you want to update. It highlights and exposes several icons.
- 4. Click the Tupdate icon. Your operating system's file selection dialog appears.
- 5. Select the file in your local file system, and click the button that confirms the selection. Wait while the file uploads.
- 6. When the upload is complete, save the page to finish updating the attachment. It is now available to users viewing the page.

#### Delete an attachment

If you find some attachment is outdated or no longer needed, you can delete it from the Attachments sidebar. Keep in mind that if you have any links to the attachment file, you need to remove them from the Page content.

- 1. Locate the desired Page by browsing or searching, and open it for editing in the Page Designer.
- 2. On the Page Designer toolbar, click the **Show Sidebar** button and select **Attachments** to reveal the **Attachments** sidebar.
- 3. Locate the attachment you want to update. If there are many attachments or some attachments are hidden, use the search field of the sidebar.
- 4. Hover your pointer over the attachment you want to update. It highlights and exposes several icons.

#### Caution

Remember that any links to the deleted file will not work, so be sure to review the page content.

- 5. Click the (Delete) icon. The attachment no longer appears in the Attachments sidebar.
- 6. Save the page to finalize deletion of the attachment.

#### Insert tables

The Insert Table button on the Page Designer toolbar launches a dialog in which you can specify the basic properties of the new table such as number of rows and columns, width, alignments, border, and header. Tables can be inserted into any region of a Page. Locate the insertion cursor in the region at the point where you want to insert a table, and then invoke Insert Table.

After a table has been inserted, whenever your cursor is in a table cell, the Table menu is available on the Page Designer toolbar. It contains a number of items that enable you to modify the table. You can:

- Insert new rows and columns.
- Move columns and rows.
- Delete rows, columns, or the entire table.
- Set the text alignment for the table.
- View and modify the table properties.

If you split a region that contains tables, the tables will be automatically resized. You may want to review them and make adjustments such as resizing the columns. You can resize columns by dragging a vertical border to the left or right.

#### Insert links

Your can insert text hyperlinks into any region of a Page. The **Insert Link** button on the Page Designer toolbar launches a dialog in which you can specify the target URL, and optionally the link text of the hyperlink. If you don't change the Label field, the URL appears as the link text.

Three types of links are recognized:

- Links to Polarion items (represented by the item type icon and its ID/Title).
- Internal portal links, represented by internal URL link text.
- External links, represented as external URL link text.

Place the insertion cursor at the point in the region where you want to create a link, and click Insert Link.

#### Tip

You can also create a link by pasting a URL into the Page text. Be sure to leave a space before and after the URL, otherwise it is not recognized as a link. The link text will include all characters in the URL. After a link is created, the link itself can be copied and pasted. A pasted link is processed and recognized according whether it is a link to a Polarion item, an internal portal link, or an external link.

#### Insert table of contents

You can insert a automated table of contents into a Page. Although it is possible to insert this into any place in any region of a Page, the most common practice is to insert it at the beginning of the topmost region. A Page only accepts one table of contents. If you try to insert another one, you see an error message.

Table of contents entries do not appear in the Page Designer, even after you save changes. They are generated only after closing the designer and returning to view mode. If the Page does not contain any Table of contents elements, a message to that affect appears when the page is viewed. The TOC cannot be copied/pasted.

You can delete the Table of contents from a Page by clicking on it in the Page Designer, and pressing your keyboard's **Delete** key.

## Insert a Widget

#### **Note**

Users of a **Polarion Pro** license cannot insert Widgets.

Widgets retrieve and format project data from your Polarion portal for reporting purposes. Normally, you use them only in LiveReport type pages, but it is possible to use them in any page of either type. If you decide you want to show some chart on project Work Items in an Info Pages, you can.

The Widgets sidebar is shown by default when you are editing any Page, and you can optionally insert Widgets into any region. The distinction between Info and LiveReport pages is made purely to generalize different use cases. The page types use different layout templates but are otherwise the same.

See Insert a Widget for details.

#### **Export to PDF or Print a Page**

# ALM REQ QA PRO REV

- 1. Open the Page after locating it by browsing or searching.
- 2. At the top of the Page, click **Expand Tools**. The page toolbar appears.
- 3. Click on the Actions Menu and choose the desired operation on the drop-down menu.

You can export Pages to PDF or print them on a printer. When exporting or printing, you can optionally hide some types of Page content.

Embedded image or attachment previews are preserved when a document is exported to PDF or printed.

#### Tip

When exporting to PDF, you can optionally customize the header and footer of the exported document in the export dialog. PDF export headers and footers are also configurable by administrators, globally or for projects. See PDF Export Header/Footer Properties.

## **Control exported content**

You can optionally exclude any region in a Page from appearing in PDF or printed output. You do this in the Page Designer, so you must have permission to modify pages in the project.

- 1. Choose **Expand Tools Description** Edit and open the Page in the Page Designer.
- 2. On the header of any region, click Actions Menu to show the drop-down menu.
- 3. In the menu, select **Hide on Print** to hide the region in printed output, and/or select **Hide in PDF** to hide the region in PDF output. You can select one of these options or both options.
- 4. Save your changes. If you are finished editing, close the Page Designer.

### Tip

In LiveReport Pages, don't confuse region headers (light gray) with Widget headers (light blue, light yellow on hover). You can only hide the region that contains Widgets, not Widgets themselves.

## **Download and Reuse Pages**

You can download a Page as a ZIP archive. You can import a previously downloaded Page to create a new, duplicate Page. Reusing Pages this way can be useful if, for example, you have built a report and need a similar report in a different project, or in the same or a different space in the same project. You can download the completed Page, import the downloaded ZIP file to any project and space to which you have access, and then modify the page as needed.

## Download a Page

- 1. Open the Page after locating it by browsing or searching.
- 2. At the top of the Page, click Expand Tools. The page toolbar appears.
- Click (Actions), and choose **Download** on the drop-down menu.
- 4. Select the location where you want to store the Page archive. The suggested archive file name includes the revision number of the page you are downloading.

#### Import a previously downloaded Page archive

- 1. Open the project into which you want to import the Page archive to create a new Page.
- 2. Open the Index page of any space in the project. It will save a little effort if you open the Index page of the destination space.

- 3. On the Index page toolbar, click the **Create New** button to launch the **Create New** dialog.
- 4. In the **Import** section of the dialog, click **Page Archive**. The **Import Page** dialog appears.
- 5. In the dialog, specify the Title and Name of the new Page to be created from the imported Page archive. The Name (ID) must be unique to the space into which you are importing. (This is especially important if you are importing to the same project and space from which the page archive was downloaded.)
- 6. If you want to change the target space, select the desired one in the **Space** field. (The current space is selected by default.)
- 7. Click **Choose File**, and then browse your local system to select the Page archive file you want to import. (This file must be one previously downloaded from an existing Page).
- 8. Click **OK** to complete the import operation, and create a new Page from the imported Page archive.

You can open the page in the Page Designer and modify it as needed.

## **Page History**

# ALM REQ QA PRO REV

## Access a Page's history

As with other artifacts, Polarion automatically tracks every change made to Pages and maintains a complete change history. Pages are included in Baselines.

Access a Page's history:

- 1. Find the page by browsing or searching and open in for viewing.
- 2. At the top of the Page, click **Expand Tools**.
- 3. On the Page toolbar, click History.

The history page opens, listing all Page revisions and baselines, if any, in chronological order with the most recent at the top.

In the history page you can:

- See the number, author, age, and date of every revision.
- Click on the number of any revision to view the state of the Page at the time the revision was committed to the repository.
- Use icons in the page toolbar to select a revision to view by date or by Baseline.

When viewing a historical revision of a Page, you can:

- Export the Page revision to PDF or send it to a printer.
- Download the Page revision as a ZIP archive.

- Pick a different historical revision to view.
- Access the attachments that were present at the time of the revision, if any.
- Open the **Create New** dialog.
- Change page parameters, if any, to see different data according to the function of the parameters.

## Tip

You cannot modify the page when viewing a historical revision.

## **Export to PDF or Print a Page**

# ALM REQ QA PRO REV

- 1. Open the Page after locating it by browsing or searching.
- 2. At the top of the Page, click **Expand Tools**. The page toolbar appears.
- 3. Click on the Actions Menu and choose the desired operation on the drop-down menu.

You can export Pages to PDF or print them on a printer. When exporting or printing, you can optionally hide some types of Page content.

Embedded image or attachment previews are preserved when a document is exported to PDF or printed.

#### Tip

When exporting to PDF, you can optionally customize the header and footer of the exported document in the export dialog. PDF export headers and footers are also configurable by administrators, globally or for projects. See PDF Export Header/Footer Properties.

#### **Control exported content**

You can optionally exclude any region in a Page from appearing in PDF or printed output. You do this in the Page Designer, so you must have permission to modify pages in the project.

- 1. Choose **Expand Tools Description** Edit and open the Page in the Page Designer.
- 2. On the header of any region, click Actions Menu to show the drop-down menu.
- 3. In the menu, select **Hide on Print** to hide the region in printed output, and/or select **Hide in PDF** to hide the region in PDF output. You can select one of these options or both options.
- 4. Save your changes. If you are finished editing, close the Page Designer.

#### Tip

In LiveReport Pages, don't confuse region headers (light gray) with Widget headers (light blue, light yellow on hover). You can only hide the region that contains Widgets, not Widgets themselves.

## **Download and Reuse Pages**

You can download a Page as a ZIP archive. You can import a previously downloaded Page to create a new, duplicate Page. Reusing Pages this way can be useful if, for example, you have built a report and need a similar report in a different project, or in the same or a different space in the same project. You can download the completed Page, import the downloaded ZIP file to any project and space to which you have access, and then modify the page as needed.

## Download a Page

- 1. Open the Page after locating it by browsing or searching.
- 2. At the top of the Page, click Expand Tools. The page toolbar appears.
- 3. Click (Actions), and choose **Download** on the drop-down menu.
- 4. Select the location where you want to store the Page archive. The suggested archive file name includes the revision number of the page you are downloading.

#### Import a previously downloaded Page archive

- 1. Open the project into which you want to import the Page archive to create a new Page.
- 2. Open the Index page of any space in the project. It will save a little effort if you open the Index page of the destination space.
- 3. On the Index page toolbar, click the **Create New** button to launch the **Create New** dialog.
- 4. In the **Import** section of the dialog, click **Page Archive**. The **Import Page** dialog appears.
- 5. In the dialog, specify the Title and Name of the new Page to be created from the imported Page archive. The Name (ID) must be unique to the space into which you are importing. (This is especially important if you are importing to the same project and space from which the page archive was downloaded.)
- 6. If you want to change the target space, select the desired one in the **Space** field. (The current space is selected by default.)
- 7. Click **Choose File**, and then browse your local system to select the Page archive file you want to import. (This file must be one previously downloaded from an existing Page).
- 8. Click **OK** to complete the import operation, and create a new Page from the imported Page archive.

You can open the page in the Page Designer and modify it as needed.

## **Build reports**

### **About reports**

One of Polarion's major benefits is its ability to track actual progress of work on various artifacts of projects such as requirements, test cases, test runs, and change requests. Data is generated as people work day to day changing the status of different Work Items, executing tests, and creating plans.

Project stakeholders can browse various artifacts and gain a good understanding of the actual progress of the project.

Pages in Polarion include special tools such as Widgets and interactive reports to support tracking based on the actual progress of work and changing data. Polarion's LiveReport Pages make it easy to use the Widgets to build robust reports visually without scripting or programming skills. Developers and other advanced users can use custom scripting with **LiveReport Pages** to create highly custom reports.

### Introduction to Widgets

#### Tip

If you don't know how to Edit Pages, you should read up on that first before diving into Widgets.

Widgets are the building blocks of reports. You insert Widgets into Pages and set their parameters to make them show the information you want in the report. You can place Widgets into any region in a Page. Widgets, combined with the layout and rich text editing features of Pages, enable you to build useful and good-looking reports that users can print or export to PDF.

When you place a Widget into a page, you see some graphical rendering of it but no data. In order to make the Widget perform its intended function, you must edit its parameters in the Parameters sidebar. Each Widget has its own set of Parameters that work together to retrieve and render information. There are several commonly used Widgets that are best illustrated in an example of a simple table of project Work Items. Many of the parameters are applicable for charts as well.

There are three main categories of Widgets:

- Chart Widgets retrieve and display project data as some type of chart.
- Utility widgets provide data tables, object counts, and interactivity for report users.
- Scripting widgets provide programmers the means to implement custom functionality in reports.

The Widgets sidebar is organized into several categories containing groups of related Widgets scripting, for example. (The Categories can be customized by a developer.)

#### qiT

You can see a brief description of each of the available Widgets in the Widgets sidebar by hovering your pointer over each one.

## **Insert a Widget**

# ALM REQ QA

You can insert **Widgets** in any region of a Page. When developing a new report, it is usually best practice to construct the Page layout first (see Modify Page Layout).

- 1. Select your Report, click Expand Tools at the top and click Expand.
- 2. Click on the top right and click **Widgets**. The Widget sidebar appears.
- 3. Place the insertion cursor inside a region at the point where you want to insert the Widget.
- 4. In the sidebar, click on the Widget you want placed in the Page. The Widget is rendered visually in the Page and selected for editing. The **Parameters** sidebar opens, replacing the Widgets sidebar and displaying the parameters of the Widget you just added.
- 5. After placing a Widget into a Page, you can do one of the following:
  - Configure its parameters. See the Widgets section.
  - Place other Widgets to complete the layout of your report, and then configure Widget parameters later. See Editing Existing Widgets.

If you decide to complete the layout first, close the Parameters sidebar. Then reopen the Widgets sidebar from the Show Sidebar button on the Page toolbar.

### **Edit existing Widgets**

# ALM REQ QA

You can edit Widget parameters after completing the report layout, or you may find at some point that you need to modify the parameters of an existing Widget.

- 1. Select your Report, click Expand Tools at the top and click Expand Tools at the top and click Alexand.
- 2. In the Page Designer, click the Widget's blue header. The **Parameters** sidebar appears with the parameters for the selected Widget and the Widget's header turns yellow.
- 3. Edit the parameters and click **Apply**.

#### Widget layout and sizing

You can explicitly set the height and width of chart Widgets to achieve a desired page layout. The **Advanced** section of the Widget parameters contains **Height** and **Width** fields, which you can set to achieve the desired layout of a chart on a Page. Chart Widgets are block type Widgets (as opposed to in-line type), and the chart rendered inside them is always centered within the block.

- Height must be specified only in pixels.
- Width can be specified in pixels or as a percentage, for example, 350 or 75%.
- For the Plan Progress Widget, you can specify only the Width parameter.

#### **Note**

These parameters are also accessible to developers via the Polarion API.

### Insert content before or after a Widget

## ALM REQ QA PRO REV

After a Widget is placed into a Page region, you may want to insert another widget, or insert other content before or after that Widget. You need to use the special menu commands found on the header of each widget.

- 1. With the page open for editing in the Page Designer, click on the (Actions) icon on the Widget's header.
- 2. In the drop-down menu, select Insert Paragraph Above or Insert Paragraph Below.
- Insert another Widget or other content in the added paragraph space.

#### Note

Users of Polarion PRO license can insert paragraphs above or below Widgets, but cannot insert new Widgets.

### Copy a Widget

# ALM REQ QA PRO REV

You can optionally copy a Widget and paste it to a different region or Page. This can save time if you need several Widgets of the same type, but with minor differences in the parameters. You can just tweak the parameters in the pasted Widget.

You can optionally copy the source code of a Widget you have inserted and configured visually. This can be useful for developers who are writing extended or custom Widgets or developing custom scripts for use in one of the scripting Widgets.

#### Copy and paste a Widget

- 1. With the Page open for editing in the Page Designer, click the . (Actions) icon in the header of the Widget you want to copy.
- 2. On the drop-down menu, choose **Copy Widget**.
  - If your browser has security restrictions that block it from direct programmatic access to your clipboard, a dialog appears prompting you to use the Copy keyboard shortcut. If you see this dialog, you can always just select the Widget in the Page and press your Copy shortcut.
- 3. Place the insertion cursor in the region where you want to paste the copied Widget
- Press your Paste keyboard shortcut (Ctrl+V or Command+V).

### Copy a Widget's source code

- 1. With the Page open for editing in the Page Designer, click the (Actions) icon in the header of the Widget you want to copy.
- 2. On the drop-down menu, choose Copy Widget Source Code.

If your browser has security restrictions that block it from direct programmatic access to your clipboard, a dialog containing the source code appears, which prompts you to press your Copy shortcut to copy the code from the dialog. Otherwise, the source code is silently copied to your clipboard.

## **Delete a Widget**

# ALM REQ QA

- 1. With the Page open for editing in the Page Designer, click on the header of the Widget you want to delete.
- 2. Press the **Delete** key on your keyboard.

Tip

You can also use the **Delete** command on the Actions menu of the Widget's header.

#### Add a Widget

## ALM REQ QA

When you place a Widget into a page, you see some graphical rendering of it, but no data. In order to make the Widget perform its intended function, you must edit its parameters in the **Parameters** sidebar. Each Widget has its own set of Parameters that work together to retrieve and render information. There are several commonly used ones which are best illustrated in an example of a simple table of project Work Items. Many of the parameters are applicable for charts as well.

All widgets can be used on like LiveReports, like Info Pages, and like Plans.

#### Add a Widget to a LiveReport or Info Page

- 1. Choose a LiveReport or Info page from a Space under the Documents & Pages topic.
- 2. Click **Expand Tools** at the top of the page.
- 3. Click P Edit.
- 4. Click in the page box where you want to insert the widget.



## Add a Widget to a Plan

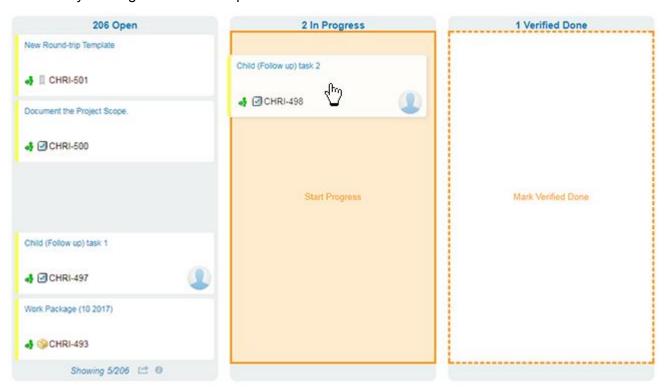
- 1. Choose a III Plan: from the III Plans topic.
- Click and then Customize Plan Report.
- Click in the page box where you want to insert the widget.



#### Kanban board

You can create a drag and drop Kanban style board for Work Items on LiveReport, Info Pages and III Plans. The number of columns (workflow states), the type of Work Items to include, and the board's appearance are all defined in the Kanban Board: Parameters sidebar.

The properties of a Swimlane (parent Work Items) and their children (Cards) can be viewed and modified by clicking on their whitespace.



#### Add a Kanban board

- 1. Select a LiveReport, Info Page, or a III Plan and launch the Widgets sidebar.
  - For a like LiveReport or Info Page:
    - a. Choose a LiveReport or Info page from a Space under the Documents & Pages topic.
    - b. Click Expand Tools at the top of the page.
    - c. Click Edit.
  - - a. Choose a Plan: from the Plans topic.
    - b. Click then Customize Plan Report.
- 2. Click in the page box where you want to insert the board.

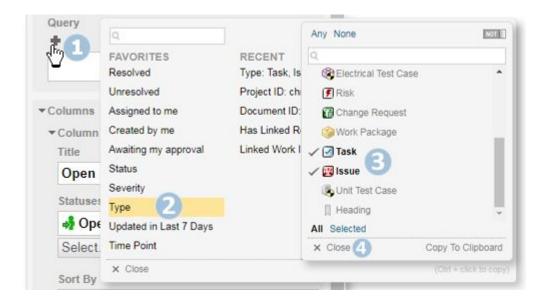


- 3. Select the Work Items or Planning tab under the Widgets sidebar.
- 4. Click on the Kanban Board icon. The Kanban Board: Parameters sidebar opens on the right.



- 5. In the Scope field, select a Project, Project Group, or the whole Repository.
- 6. Select a **Query Type**. (If you are unsure what kind of query type to use, just leave it on **Lucene**. You will be able to build a query graphically in the steps that follow.)
- 7. Select a Work Item Type.
  - To add a single Work Item type to the board, you can select it from the Type drop-down list.
  - To select more than one Work Item type, leave the default \* Work Item selected.
- 8. Use the **Query** field to define the Work Items you want added to your Kanban board.
  - a. If you're familiar with Polarion's advanced command-line query syntax, you can paste your query directly into the field.
  - b. Click 

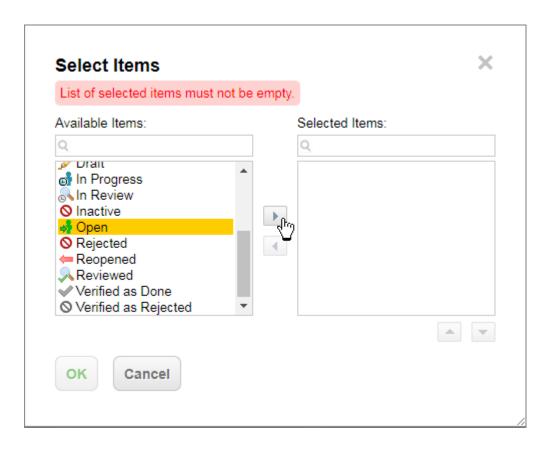
     to define what items to include graphically.



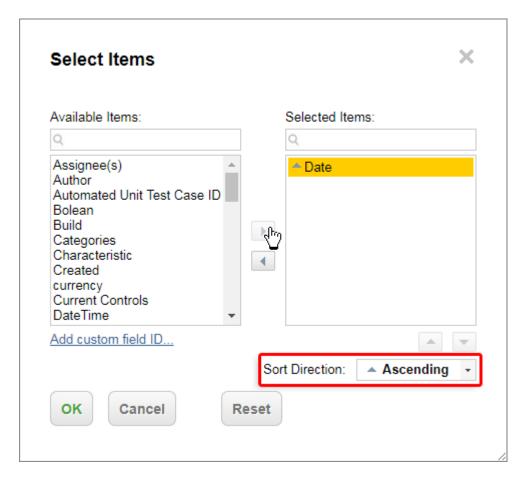
c. Click a set of options to include and click × Close . ( Click ♣ multiple times to add additional criteria.)

In large projects, you can combine the command-line and graphical query options to really fine-tune your selection.

- 9. Add a **Title** for the first status, for example **Open**, in the **Columns** section.



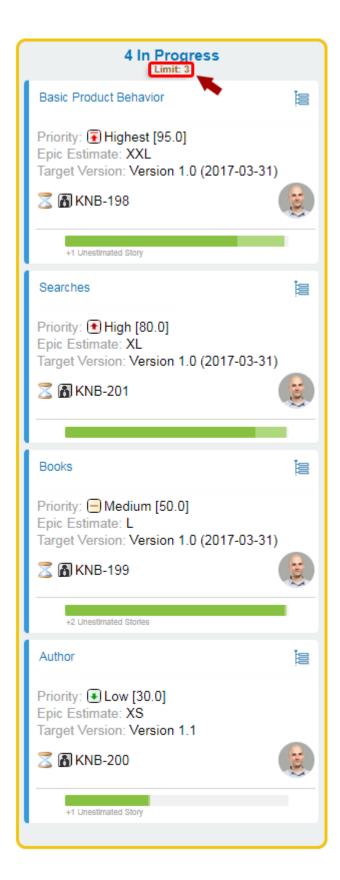
- 11. Click **OK**.
- 12. (Optional) Click Select... under **Sort By**, and select how the Work Items are sorted and the direction that they should be sorted in.



# 13. Click **OK**.

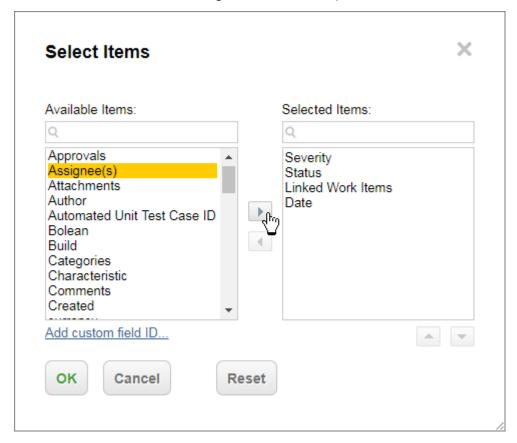
14. (Optional) You can set a **Limit** on the number of items that appear in the column. The **Limit** appears at the top of the column and if it is exceeded, the column is highlighted with a yellow border.

(The default setting of o disables this feature.)

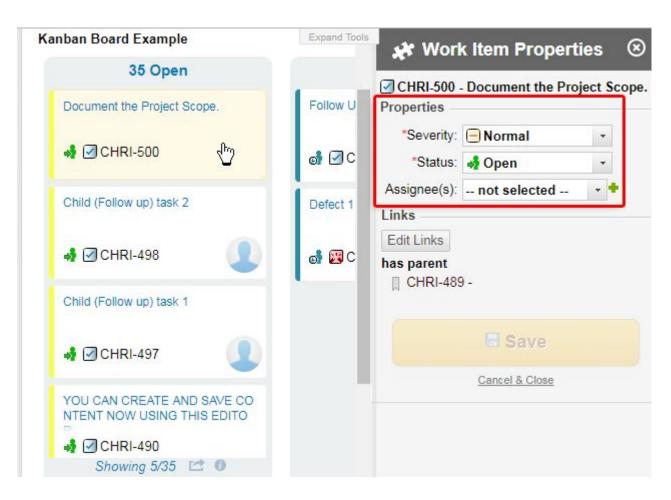


15. Click **Add Column +** and repeat steps 10 through 14 for each workflow status.

- 16. (Optional) Click on a card to launch Work Item Properties Sidebar and add fields that you want to appear.
  - As a report editor: (Edit mode):
    - a. Click Select... under Work Item Properties Sidebar Fields to add any fields that you want to appear under the Work Item Properties Sidebar. (Launched when a Kanban, Table or Work Items Board widget item is clicked.)



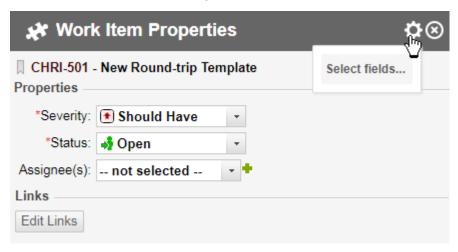
- b. Select a field on the left, click, and click **OK** when you've finished adding the fields you want to include.
- c. Click a card and the selected fields appear in the Work Item Properties Sidebar.



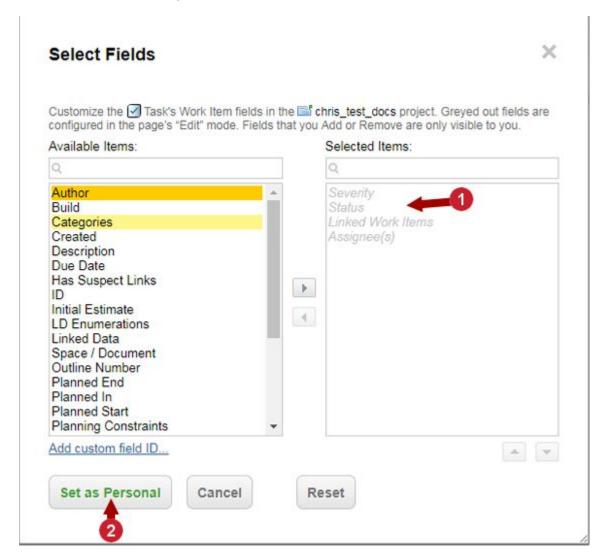
#### As a user:

Anyone who clicks on an item in a widget that supports the Work Item Properties Sidebar.

- a. Click on a **Card**, **Swimlane**, or table item.
  - The Work Item Properties Sidebar appears on the right.
- b. Click the icon on the top right and click **Select Fields**.



The **Select Fields** dialog appears.



- 1 Selected items can only be added or removed when the report is in **Edit** mode.
- The selected fields will be your personal, default fields for all widgets that support the **Work Items Properties** sidebar with the currently selected **Project**.
- c. Select a field on the left, click , and then click **Set as Personal** when you've finished adding the fields you want to include.

#### **Note**

Work Item Properties Sidebar fields:

- o Grayed out fields can only be removed when the report is in **Edit** mode
- The fields will appear in the Work Item Properties Sidebar for all Kanban, Table and Work Items Board widgets.
- o Fields added by a user (when not in **Edit** mode), will only be visible to them.
- o Add fields in **Edit** mode above if you'd like other users to also view them by default.
- o The added fields will only appear for the selected Work Item type ( ☑ ☑ ) and □ Project.
- d. The fields will be added to the Work Item Properties Sidebar of all the Kanban, Table and Work Items Board widgets in the selected project.
- 17. Click **Apply** or follow the instructions in the following **Customize the Kanban Board's Appearance** section to change the board's appearance.

#### Customize a Kanban board's appearance

You can customize the appearance of the cards by adding custom velocity snippets. You can change the color of the card, remove the status border color, show different Work Item fields, re-arrange the fields already rendered, and more. The custom Velocity scripts will be rendered instead of the default java scripts. Default velocity snippets are included with the widget under the Board Script and Card Script parameters and are a great starting point. The sample scripts mimic the java rendered output that's provided out of the box.

#### **Customization options:**

- Board Script: Insert at the beginning of the board. Always use to add custom CSS styles and/or JavaScript common for all your cards. This avoids needing to duplicate the scripts for each card.
- **Card Script**: Defines the content that appears within the cards.

The **Card Script** Velocity script allows for the following additional objects:

- sworkItem: The Work Item rendered in the card.
- \$columnindex: The column that the card is rendered on.

#### **Customization example:**

- 1. Click on the top right of the board in **Edit** mode to open the **Kanban Board: Parameters** sidebar.
- 2. Click Advanced.

3. Define **The maximum # of results per columns** that appear.

Click the **Showing** #/# link on the bottom of a column to see the next list of items.

4. Click **Yes** to Enable Customization.

.kanbanWorkItemTitleGradientDiv to,

- 5. Click on **Board Script**. For a full screen view of the script hit **F11**. Hit **Esc** to exit full screen view.
- 6. Change the background color of the card by adding background-color: #d0d2e0; to .kanbanWorkItemDiv.

(#d0d2e0 can be replaced with any color you'd like.)

7. Change the gradient of a long title by changing the background property in

```
background: linear-gradient(to bottom, rgba(255, 255, 255, 0), #ccffff) repeat scroll 0 0 rgba(0, 0, 0, 0);
```

8. Enable the **Work Item Properties Sidebar** by adding:

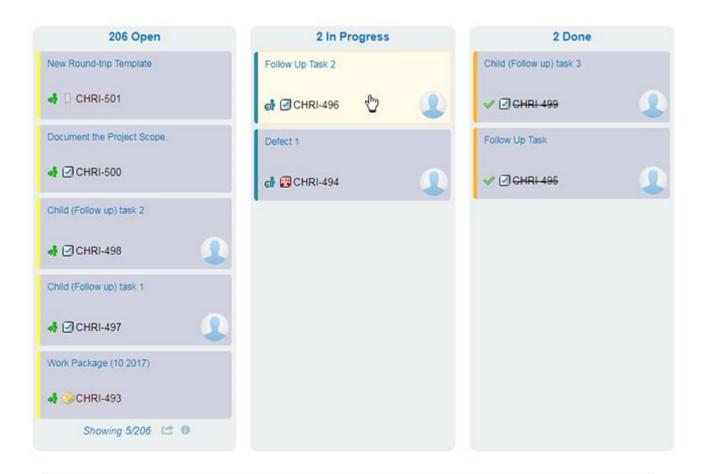
```
.polarion-sidebar-highlight .kanbanWorkItemDiv {
          background-color: #fff9e8;
    }
```

The entire **Board Script** would look like the following:

```
kanbanWorkItemDiv {
   padding: 7px 3px 3px 7px;
   box-sizing: border-box;
   border-left: 5px solid;
   border-right: 1px solid #EEEEEE;
   border-bottom: 1px solid #EEEEEE;
   border-bottom: 1px solid #EEEEEE;
   border-radius: 5px;
   position: relative;
   background-color: #d0d2e0;
}
...
kanbanWorkItemTitleGradientDiv {
   bottom: 0px;
   height: 12px;
   left: 0;
   position: absolute;
   width: 100%;
   display: inline-block;
   background: linear-gradient(to bottom, rgba(255, 255, 255, 0), #d0d2e0)
repeat scroll 0 0 rgba(0, 0, 0, 0);
   filter: progid:DXImageTransform.Microsoft.gradient
(startColorstr='#00FFFFFF', endColorstr='#FFFFFFF', GradientType=0);
}
...
polarion-sidebar-highlight .kanbanWorkItemDiv {
    background-color: #fff9e8;
}
...
```

9. Click Apply.

The example script above changes the Kanban board's appearance:



#### Note

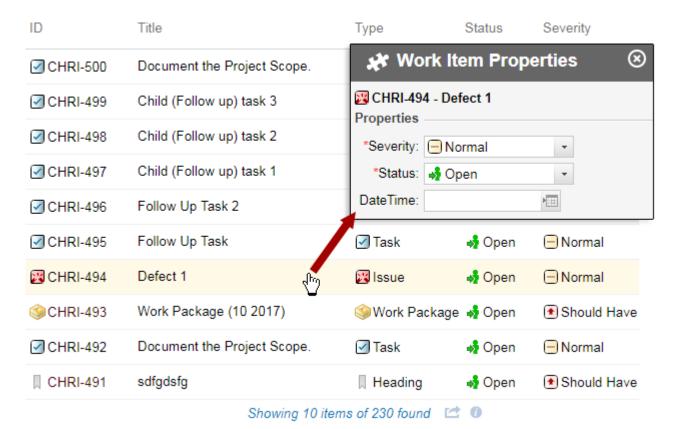
#### To revert customization changes:

The Work Item Properties sidebar and the ability to drag and drop are still available when customizations are enabled, and you can always revert to the internal Java rendering by setting **Enable Customization** to **No**.

# **Table of Work Items widget**

A simple but useful report element is a table of unresolved Work Items sorted by type, severity, and current status, and showing several key fields such as **ID**, **Title**, **Status**, and **Severity**.

If you have the appropriate permissions, you can modify a Work Item's fields by clicking on it.

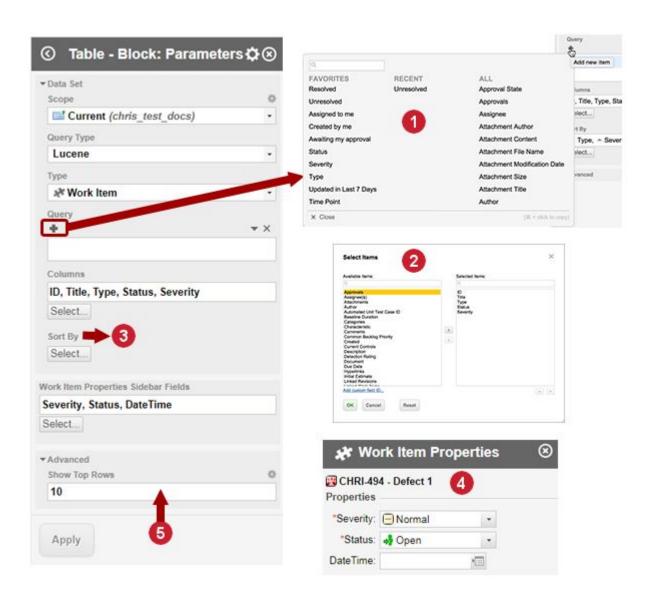


The fields that appear can be configured in the **Table - Block: Parameters** sidebar.

Launch the Table - Block: Parameters sidebar:

- For a III Plan: Click on III Plan: Cl
- For LiveReport and Info Pages: Click on Expand Tools → Edit → ✓ above the table widget.

Then follow the instructions defined in Parameters sidebar.



1 Visual query builder
2 Select fields to show as columns.
3 How the Work Items are sorted in the table.
4 Select Work Item Properties sidebar fields.



Number of items that appear in the table.

#### Note

The Work Item Properties sidebar Fields only work for WWork Items.

See the Kanban Board section to learn how to define what fields appear and what widgets are affected.

The table will show a subset of all available project data, so it's necessary to set some parameters to retrieve the desired data.

- Scope: Specify the scope of the data to be retrieved. The default is the current project, but you could retrieve data from a different project from the one in which you are currently working, or from the entire repository. In this example, the scope is the current project.
- Query Type: Select the guery language to retrieve data. The full arsenal of Polarion's data retrieval capabilities are available here, including Apache Lucene query language, SQL, or combinations of Lucene and Velocity, or SQL and Velocity. For the purposes of an easy example, Lucene is used, as many users are familiar with this type of query, and the Visual Query Builder is integrated in Widgets enabling users who don't know a query language to build queries visually.
- **Type**: Specify what main artifact type is the basis for the data. You can opt for the data to be Work Items, Documents, Plans, Test Runs, or Pages. You can refine this by selecting one of the subtypes defined for the different artifact types (if any).

If the artifact type is Work Item or Document, you can select the type in the list. For example, for the Work Item artifact type, there might be Requirements, Test Cases, Issues, Tasks, and so forth. The Document artifact type might have subtypes such as Requirements Specification or Risk Analysis. The subtypes list varies according to the Scope. For example, Repository scope lists all Work Item and Document types defined in the global configuration, whereas selecting a project limits the list to the types defined for the selected project.

For this example, Work Item is selected, because the table must show Work Items.

- Query: Refine data shown in the table through a query. In this simple example, Lucene is selected as the query type. For Lucene queries, the same Visual Query Builder tool you may already have used in the Work Item Tracker is available. You can use it to add multiple visual query elements to the Query to build queries ranging from ultra simple to very complex. For this simple example, only one element is added: a query on the Resolution field, for a value of "unresolved".
- **Columns**: Specify the columns to show in the table, and their order, using the **Select** button under the **Columns** field. The selection dialog is the same one you use to customize the Work Items table in the Work Item Tracker. Columns in the table display Work Item fields, because Work Item is selected in the **Type** parameter.

For this example, the ID, Title, Type, Status, and Severity fields are selected to appear as table columns.

#### Tip

What you can select to show in Columns depends on the artifact type selected in Type. Work Items have different items than Documents or Test Runs.

Sort By: Specify how the Work Items are sorted in the table. The specification of the example
called for sorting by type, severity, and current status. Sorting by Type first causes the different
Work Item types to be grouped. So all Issues appear first, followed by all Change Requests,
Requirements, and finally Test Cases. Within those groups, the rows are sorted by severity
in ascending order. Finally, the rows for a given severity are sorted by the current status of
the individual Work Items.

This field also has a Select button, which leads to the same dialog in which you can select the fields to sort on, the sort order (ascending/descending), and the order of the fields.

- Work Item Properties Sidebar Fields: Select the fields that appear in the Work Item Properties Sidebar.
- Show Top Rows: Expand the Advanced section of the sidebar to see this field. This can be an important parameter if the table is potentially very large. The default is 50, meaning the table only shows the first 50 items matching the query, and displays a link that opens the Work Item Tracker in a new browser tab and load all the marching Work Items.

# Warning

Click the **Apply** button in the sidebar to update the visual rendering of the **Widget** in the **Page Designer** after modifying parameters. This applies to all Widgets.

Users do not see any changes to the **Page** until you save changes you make in the **Page Designer**. After the **Page** is saved, the report is live in the sense that the table reflects the status of the items in the project as of the time the user loads the report page. Users can see any later changes in the data by refreshing the page in their browser.

#### Work Items board widget

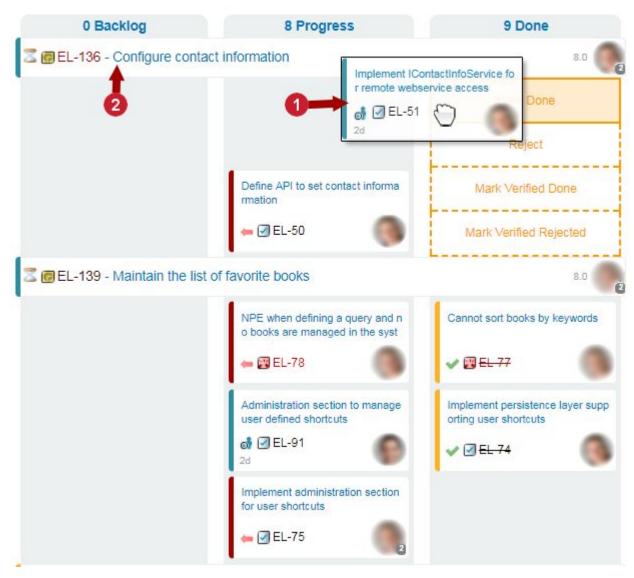
This widget creates a drag and drop Work Item Board on Mi LiveReport, Mi Info Pages, and Mi Plans.

Transitions and Statuses are based on the configured Work Item's workflow.

If there are no additional workflow conditions or required fields, a Work Item is automatically saved when dragged to a new state.

If a Work Item requires additional input, for example a **Resolution**, the Work Item Properties sidebar automatically appears when the item is dragged to a new state.

# **Planned Items**





The properties of a Swimlane (parent Work Items) and their children (Cards) can be viewed and modified by clicking on their whitespace.

Enter the required information and click **Save**.

See Parameters sidebar and Work Item Properties sidebar to customize its visible fields.

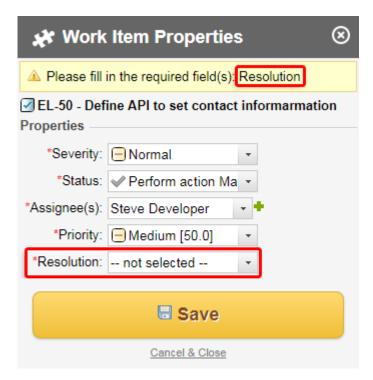
# **Work Items Properties sidebar**

The Work Item Properties sidebar fields only work for \* Work Items.

See the Kanban Board section to learn how to define what fields appear and what widgets are affected.

Enter the required information and click Save.

Work Item Properties sidebar fields:



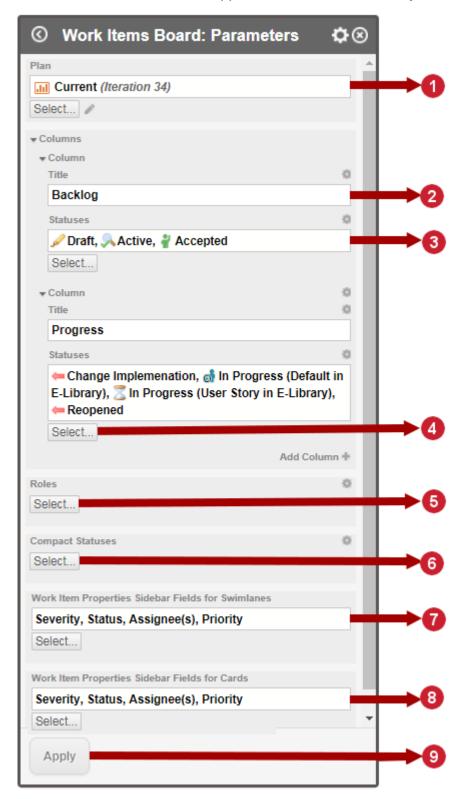
#### Work Item Properties sidebar fields:

- Grayed out fields can only be removed when the report is in **Edit** mode.
- The fields appear in the Work Item Properties sidebar for all Kanban, Table and Work Items Board widgets.
- Fields added by a user (when not in **Edit** mode), are only be visible to them.
- Add fields in **Edit** mode above if you'd like other users to also view them by default.
- The added fields only appear for the selected Work Item type ( and and and and are project.

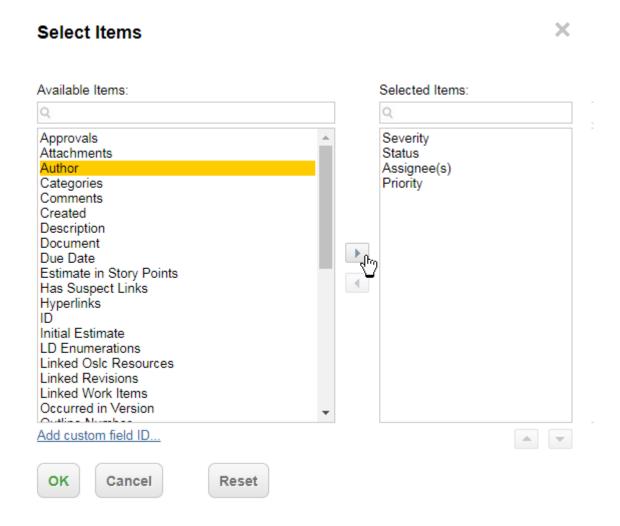


#### **Customize the Parameters sidebar**

You can customize what fields appear in the Work Items Properties: Parameters sidebar.



- Select a III Plan to render a Work Items Board for.
- (Click **Select** to select a **■Plan** from a list or **r** to enter its ID manually.)
- 2 Define a column's **title**.
- Opening the column's Status(es).
- Select the **Status** options that appear in the column. Available options depend on what is defined in the **Work Item** workflow.
- Define the sub task link roles: Is triggered by, Implements, Verifies, Has Parent and so on.
- **6** Work Items appear in compact form when in the selected Statuses.
- Fields visible in a Swim lane's Work Item Properties sidebar. (Status or Severity by default.)
- Fields visible in a Card's Work Item Properties sidebar. (Status / Severity by default.)
- Applies any changes made in the sidebar.
- 1. Open the **Parameters** sidebar.
  - For a I Plan: Click on I Click on Customize Plan Report → Ø above the table widget.
  - For LiveReport and Info Pages: Click on Expand Tools → Expand Tools → Medit → Medit
- 2. Click Select below Work Item Properties Sidebar Fields for Cards.
- 3. The **Select Items** dialog box appears.



- 4. Select an item from the **Available Items** box on the left and click **1**. The item appears in the **Selected Items** box on the right.
- Click Apply.
- 6. Repeat this process for all the fields you would like to add to the Work Item Properties Sidebar Fields for Cards and click OK.
- 7. Follow the same steps to add fields for Swim lanes by clicking **Select** below **Work Item** Properties Sidebar Fields for Swim lanes.

# **Scripted Chart**

A scripted chart renders a chart defined by a custom Velocity or JavaScript.

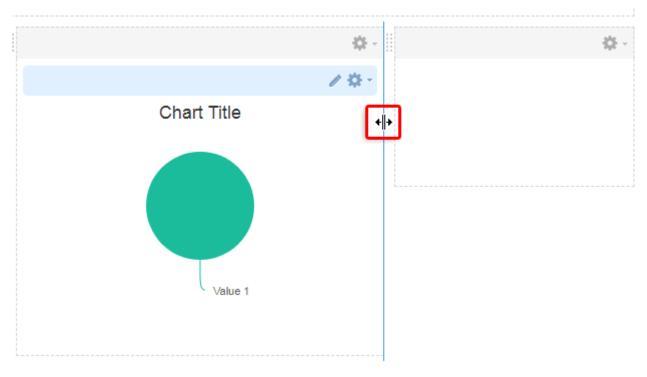
## Add a Scripted chart

1. Select a LiveReport, Info Page, or a III Plan and launch the Widget sidebar.

2. Click in the paragraph that you want to add the Widget to.



You can add a new paragraph box before or after the existing boxes or resize them.



- 3. Click on the **Charts** tab in the **Widgets** sidebar.
- 4. Click on the Scripted Charts icon:



5. Add a Velocity or JavaScript into the **Script** field. If you don't have a script of your own, you can paste the example below in and customize it.

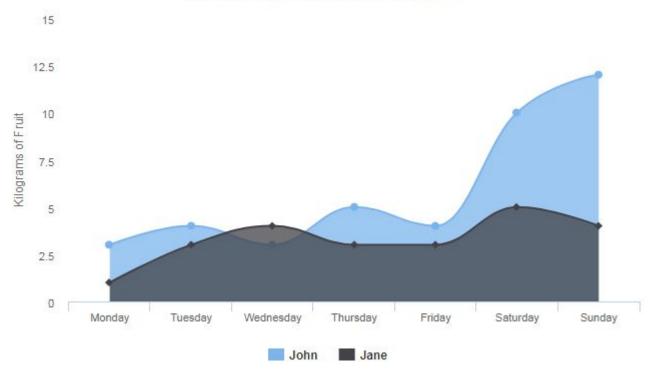
```
{
    chart: {
        type: 'areaspline'
},
    title: {
        text: 'Average Weekly Fruit Consumption'
},
    xAxis: {
```

```
categories: [
             'Monday',
             'Tuesday',
             'Wednesday',
             'Thursday',
             'Friday',
             'Saturday',
             'Sunday'
        ]
    },
    yAxis: {
        title: {
             text: 'Kilograms of Fruit'
    },
    series: [{
        name: 'John',
        data: [3, 4, 3, 5, 4, 10, 12]
    }, {
        name: 'Jane',
        data: [1, 3, 4, 3, 3, 5, 4]
    } ]
}
```

Click **F11** to adjust the script in full-screen mode and **ESC** to exit full-screen mode.

6. Click Apply. The charts are rendered and appear on the page.

# Average Weekly Fruit Consumption



# 7. Click **Save**.

## **Troubleshoot Scripted Chart Errors**

Here are some things you can try before sending your chart to a widget developer for inspection.

- 1. Debug the chart using the web browser's built in developer tools.
  - a. Click on the chart (so that it's highlighted in yellow), and then right click and select **Inspect** (Chrome), **Inspect Element** (Firefox), or **F12** (Edge).
  - b. Click on the **Console** tab that appears on the bottom half of the browser.
  - Click on the link beside the error.

```
● ▼Uncaught ReferenceError: chart is not defined VM1469:32
```

The browser jumps right to the line in the chart's code that is causing the error.

```
);

var options = chart ('container', {

chart: {

plotBackgroundColor: null,

plotBorderWidth: null,

plotShadow: false,

type: 'pie'

},

title: {

text: 'Browser market shares in January, 2018'
```

(In this example, "Chart" was not removed at the beginning of the example taken from highcharts.com/demo.)

d. Adjust the code and click **Apply**.

If the error is corrected, the chart appears in the LiveReport.

- 2. Modify and troubleshoot a chart using highcharts.com.
  - Go to highcharts.com/demo.
  - b. Select a chart type and personalize or troubleshoot it by clicking **EDIT IN JSFIDDLE** >.
  - c. Before you paste the chart into Polarion, remove *Highcharts.chart* before *('container'* at the beginning of the script.

#### Tip

A common Velocity script error is failure to correctly escape single-quote characters.

When scripting in Velocity, if a *value* contains a single quote (for example, Matt's) it must be escaped using the \$esc.javascript(...) method.

## Simple chart example

A simple but useful report is one that shows a chart of unresolved Work Items with high severity values currently in the repository, separated by Work Item type, reporting also the total number of Blocker and Critical defects. Such a report requires only two Widgets:

- Separated by Pie chart
- **Object Count Inline**

After placing the pie chart Widget in some region of the Page, set its parameters as follows:

- Title: Enter a title that describes what the chart shows, for example, something like "Unresolved Blocker & Critical Work Items by Type".
- **Scope**: Specify the scope, in this case, the chart is reporting across the entire repository, so **Repository** is selected in this parameter.
- Query Type: Specify the query type. This example uses Lucene because it provides the possibility to build queries visually. (Report authors who know Velocity or SQL can opt for to use these types of queries.)
- Type: Specify the artifact type. For this report, Work Items is used, so Work Item is selected for this parameter.
- Query: Specify the query; for this chart, two query elements are needed. The report is for unresolved Work Items, so the first query element is for that. Simply choose Unresolved in the Favorites section of the Query Builder. Then add another query element to select the severity level of the unresolved items. Select severity in the Query Builder panel. The report is to show items with Blocker or Critical severity, so select both of these in the value picker before closing it.
- Separate By: Select Type in this field to separate the matched items by type. Recall that the report is supposed to show Blocker and Critical items by Work Item Type. The chart Widget makes this quite easy.
- Show Data Labels: Specify whether or not the pie chart's wedges are labeled. The Yes option is a good choice in this example.
- Show Zero Values: Specify whether or not the chart should include a segment for zero matched items. For example, are no Change Requests with Blocker or Critical severity, selecting Yes causes the chart to render a small wedge reflecting this.

Remember to click **Apply** to reflect changes in the chart rendering on the Page. To complete this report as specified, add a paragraph above or below the chart Widget, and add an Object count -**Inline** Widget at the insertion point. Set this Widget's parameters as follows:

- **Scope**: Select **Repository**, because the requirement is to report a count of items in the repository.
- **Query Type**: Select **Lucene** to keep things simple.
- **Type**: Refine so that only **Defect** type Work Items are counted. If the repository contains this type, it appears in the types list under Work Items. This example assumes there is a type **Defect** and selects that type in this parameter. Now the Widget renders a count of only **Defect** type items.

- Query: Because Lucene is the query type, the Visual Query Builder is available. Again, two
  visual elements are needed: one to select only unresolved items, and one to select the severity
  levels to be included in the count. In this case, Blocker and Critical are selected.
  - If you click **Apply**, a number appears in the paragraph of the Page where you inserted the Widget. However, another user reading the Page will not know what the number means.
- **Text Before** and **Text After**: These fields provide a quick way to include some explanatory text before and after the object count.

In this example, the report author might enter There are currently in the Text Before parameter, and BLOCKER & CRITICAL Defects in the repository. in the Text After parameter. When Apply is clicked, the Widget renders There are currently 42 BLOCKER & CRITICAL Defects in the repository., assuming there are 42 items that match the query in the parameters.

## Trend chart example

Charts showing trends can be highly useful. For example, suppose you are interested in comparing the trend of high-severity vs. low-severity Issue type Work Items. This is easily done with the Multi Line Trend Chart. You can configure the chart with one data set to show high-severity items and one to show low-severity items. If you want to see more tends in the chart, you can add more data sets.

- 1. Place a **Multi Line Trend Chart** Widget in some region of the Page.
- 2. In the default **Set**, specify the scope of the Work Items (a project or the entire Repository), select Lucene as the query type. Then in the Query section of the Set, build a query with two elements:
  - Type: Issue
  - **Severity**: **Basic**, **Low** (or whatever the lowest severity values are for the Work Item type **Issue** in the selected Scope).

Set other fields such as:

Name: Low Severity

Color: Gray

- 3. Now click **Add Set** and configure the second Set to query for high-severity Issue type items, naming the Set *High Severity*, and using a different color, such as Red. For Widgets that have the color parameter, you can specify common color names supported in HTML (Red, Blue, Cyan, Black, etc.) or hexadecimal values (#ed1c24, #0000ff, etc.) For example, multi data set type chart Widgets use the color parameter to visually differentiate the data.
- 4. Enter a date range in the **Dates** parameter. This parameter is available in chart Widgets where the data retrieved need to be filtered by a date range. The **From** and **To** fields enable you to configure the time scope of the trends you want the chart to reflect. The **Horizontal Scale Axis** enables you to set the frequency of points plotted, and when plot points are calculated. For example if the scale is *Month*, you might specify a plot point value of 1, meaning data from the first day of each month is used for the plot point.
- 5. Click **Apply** to reflect the configuration in the chart on the Page.

If you want to configure additional Sets, use Add Set. Configure each one as desired, building a query in each Set to retrieve the data you want, which will appear as another line in the chart.

## Tip

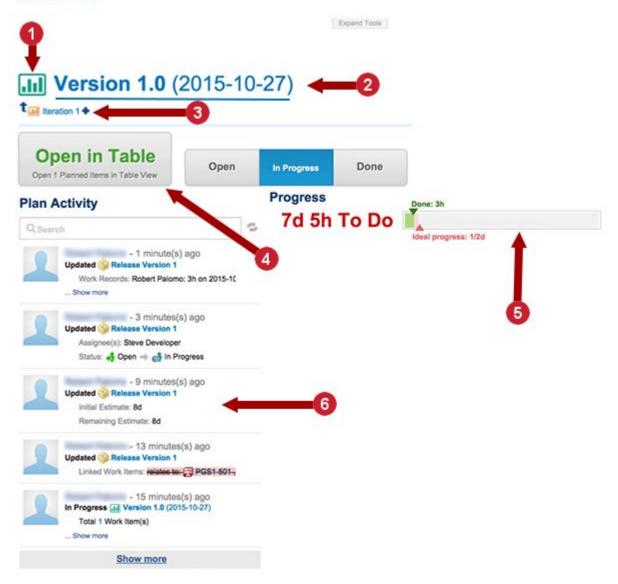
The Multi Data Set Pie Chart also provides the possibility to reflect multiple sets of data in a pie chart format.

# Simple Plan example

You can use the Widgets in the Plans group to report information about Plans such as activity and progress, and provide some interactivity such as creating a new child Plan, setting Plan status, and opening a Plan. The Widgets for this example are:

- Plan Label
- Plan Status Button
- Open Plan in Table
- Plan Activity Stream
- Plan Progress

#### Release Plan:



- 1 Plan Label widget.
- 2 Hyperlink to plan.
- Oreate new child Plan.
- Open Plan in Table widget.
- 6 Plan Progress widget.



# Plan Activity Stream widget.

#### Plan Label

Create a new region and insert an appropriate heading (Release Plan, for example). Below the heading insert a Plan Label Widget. This Widget shows the name of a Plan, which is a hyperlink that opens the Plan. It displays any existing child Plans, with the same hyperlinking to open them. It also provides the option of creating a new child Plan, based on some template Plan, directly from the Page. Set the Widget parameters as follows:

- 1. Use **Select** to select the Plan you want for which you want to report information. For purposes of this example, select a Release plan.
- 2. Assume you want to allow users to create a new child Plan directly from the Page. In the **Child Template** field, use the **Select** button and choose a template to be the basis for child Plans of the one specified in the Plan field. For purposes of this example, select **Iteration**, as an iteration Plan would typically be a child of a release Plan.
- 3. In **Show Add Button**, click Yes. This allows users who have permission to create new Plans to do so from the finished Page.
- 4. Click **Apply** to complete the configuration of this Widget.

#### Open Plan in Table and Plan Status

Beneath the Plan Label widget, insert these Widgets, leaving a few horizontal spaces between them:

- Open Plan in Table
- Plan Status Button

In the parameters for both these Widgets, select the same plan you selected in the **Plan Label** widget.

With these Widgets in place and configured, a Page user is able to open the Work Items of the specified Plan in the Work Items table, and change the status of the specified Plan directly from the Page.

# **Plan Activity Stream**

On your Page, you can easily show the **Activity Stream** of any **Plan**, enabling users to see the latest activities. To do this:

- 1. Create a new region below the one you created earlier, and enter some heading like "Plan Activity".
- 2. Place a Plan Activity Stream Widget into the region, and use the **Select** button in the **Parameters** sidebar to select a Plan. For the purposes of this example, select the same Plan you selected in the Plan Label widget parameters.
- 3. Expand the Advanced field and enter a number in Max. number activities to show. For example, if you enter 5, the stream in the finished page shows only the five most recent activities

on the selected Plan. The Widget running in the finished Page enables users to see more activities.

4. Click **Apply** to finish the configuration of this Widget.

# **Plan Progress**

The Plan Progress Widget renders a graph and statistics of a selected Plan that show the current progress of the Plan. The information is updated when a user loads the finished Page. Refreshing the Page refreshes the information.

- 1. On the region menu of the region you added above, choose **Split Region** or create a new region wherever you want on the Page, and add a heading like "Plan Progress".
- 2. Insert a Plan Progress widget in the new region beneath the heading.
- 3. In the **Plan** use the **Select Plan** button to select the same Plan you selected earlier in the Plan Label Widget.
- 4. Review the two options Show Ideal Progress and Show "To Do". This information is shown by default. If you want to exclude one or both, click No in the option field.
- 5. Optionally set the chart width. Expand **Advanced** and enter a value in pixels or percentage. For example: 600px or 85%.
- 6. Click **Apply** to finish the Widget configuration.
- 7. Click Save.
- 8. Click **Back**. The Page is now ready for users.

# Simple Test Run example

A test manager may not want to grant testers full access to Test Runs, where they could access templates, properties, and other features, but still wants to enable them to view information about a Test Run and even execute it. Using special Widgets in the Test Runs category of the Widgets sidebar, the manager can guickly construct a Page that...

- Displays the name of the Test Run.
- Displays the current status of the Test Run and its Test Cases.
- Enables a tester to execute the Test Run in Polarion.
- Enables a tester to export Test Cases to Excel, and after executing them externally, import the
  results back to the Test Run.
- Change the workflow status of the Test Run.
- View electronic signatures and test records.

# SST-EX1 (System Security Test) —— 1 **Execute Test Run** Status Overview **Execute Test** 1 waiting Test Case execution(s) 0 Blocked

# Set Test Run Status

Use this Widget to change the workflow status of this Test Run:



# **Test Records**

This section will show test records after the Test Run has been executed.

No test cases

# **Electronic Signatures**

No signatures yet.

Export Tests To Excel Import Results From Excel



1 Test Run label	6 Import manual test results
2 Execute Test Run button	6 Test Run status button
3 Test Run overview	7 Test run signatures
4 Export tests	8 Test Records

# **Building the Page**

1. Create a new LiveReport type Page and give it an appropriate name. This name appears formatted as Title in the new Page. Do not make the page name the same as the Test Run name because a Widget will display that. Consider if you might want to reuse the Page for different Test Runs over time and if so, make the page name more or less generic.

2. Under the title text, place a Test Run Label Widget. In the Widget's parameters, click the Select button, and in the dialog select the Test Run for which you are building the Page. You can optionally choose whether to have the Widget display the name of the Test Run Template from which the Test Run is derived, in addition to the Test Run's name. Always click Apply when finished with parameters.

## Enable the display of a Test Run's status

It is generally useful for the page to show the current status of the Test Run and the Test Cases selected for it.

- 1. Insert a new region below the first one and split it.
- 2. In the left region, type a heading like "Status Overview" and format it as e.g. Heading 2. Then, under the heading, insert a *Test Run Overview* Widget, and in its parameters, select the same Test Run as you selected in step 2 above, and click Apply.

#### **Note**

You will select the same Test Run for all Widgets in this Page, since this example deals with only one Test Run.

#### **Enable Test Run execution**

You can set up the Page to enable testers to execute the Test Run (they must have the necessary permission for this). The example page will enable both internal execution in Polarion and external execution of the Test Run via Excel round-trip. On a real page, you would most likely provide only one of these options.

1. In the new empty region on the right, type a heading **Execute Test Run** and format it as before. Under the heading, insert an **Execute Test Run Button** Widget, select the **Test Run** in its parameters. You can optionally specify sorting of the Test Cases. Click **Apply** when finished.

#### Tip

Center-align the Widget in the region.

2. Under the button Widget, insert two additional Widgets: **Export Tests** and **Import Manual Test Results**. In the parameters of each one, select the **Test Run**.

In the export Widget, you can optionally select a subset of the Test Run's Test Cases by means of a query. For example, it make sense to only export Test Cases with the **Open** status (status:open). This and the remaining parameters are optional:

- Specify the name of the Excel export template that should be used by the exporter only if you have some custom export templates and do not want the default template to be used.
- Specify a file name. A default file name for the exported Excel file is used if you don't.
- (Optional) Change the label displayed to users. For example, if you have a query to select only open Test Cases, you can change the label to, e.g. Export Open Test Cases to Excel.

- 3. Under the export Widget, place an Import Manual Test Results Widget. Select the Test Run as with the other Widgets, and optionally change the Widget label. Click Apply when finished.
- 4. Add a new region to the bottom of the Page and split it in preparation for the next steps. This is a good time to save your work in progress.

# **Enable Test Run status change**

In the new left-hand region, create a heading like "Set Test Run Status", and under it insert a Test Run Status Button Widget and select the Test Run in its parameters.

## **Display Electronic Signatures**

You should only implement this if your project is configured to use electronic signatures for Test Runs.

- 1. In the empty right-hand region, create a heading like "Electronic Signatures", and under it insert a **Test Run Signatures** Widget and select the Test Run in its parameters.
- 2. Click **Apply** when finished.
- 3. Add a new region to the bottom of the Page in preparation for the next steps. (Do not split it.)

#### **Display Test Records**

- 1. In the new region, insert a **Test Records** Widget and select the Test Run in its parameters. In the columns parameter, select which test record information you want to display in the Page.
- Click Apply when finished.
- The Page is now complete. Save it, click **Back**, and it is ready for use. Once you understand the Widgets, you can probably build a functional Page like this one in fifteen minutes of less.

There are some other Test Run related Widgets not used in this example:

- **Test Run Activity** renders the selected Test Run's activity stream in the page.
- Import Automated Test Results enables the Page user to select a file with xUnit test results and import it to the specified Test Run.

#### **Making Reports interactive**

# ALM REQ QA

You can provide a limited level of interactivity in reports using **Page Parameters**. This is considered an advanced feature. It enables you to provide report viewers with some options for selecting the data they want to see. For example, you might enable users to select a date range for a trend chart or the number of rows they want to see in a table.

Page parameters can only be used with many of the Widgets, and some of their parameters. You can tell if a parameter can be associated with a page parameter by clicking the small gear icon displayed in its section of the Parameters sidebar, and then clicking **Parameters** to show the Page Parameter(s) compatible with that Widget Parameter, if any have been defined.

You can only have one Page Parameters Widget in a Page. You need to decide what parameter values you want report viewers to be able to change in which widgets, define Page Parameters for them, and specify all of them in the single allowed Page Parameters Widget.

For example, suppose a Page contains a multi line trend chart, one or two pie charts, and two tables. You might decide to allow report viewers to interactively specify:

- A date range in the trend chart
- The number of rows to show in the tables
- Whether to show data labels in the pie charts.

At runtime, the Page Parameter includes an action **Save as Default**. A user can change the values and click this link to have the changed values become the defaults. Other users accessing the page see the values in the Page Parameter widget.

# **Define Page parameters**

Assuming the Page is otherwise complete, with the Widget parameters all configured, the first step in providing user interactivity is to define the Page Parameters. With the Page open for editing in the Page Designer, open the **Page Parameters** sidebar from the Show Sidebar icon on the Page Designer toolbar.

Three Page Parameters are needed for the example stated earlier in this topic. You create new Page Parameters using the **Parameter** button, which drops down a menu of data types:

- String
- Integer
- Boolean
- Enumeration
- Date

Each Page Parameter must correspond to the data type it will manipulate in the Widget(s). For example, to allow users to set a start date in a trend chart, the Page Parameter must be of the *Date* type. To allow users to specify the number of rows to show in a table Widget, the Page Parameter must be of *integer* type.

When creating a Page Parameter of type String, Integer, Boolean or Date you must specify two properties: Name and ID. Name is what report users see in the page. ID is the identifier used by the system. By default, the values are identical. The difference is that ID supports only ASCII characters. If you need the label to display a different character set, specify ID using only ASCII characters, and Label using your language's character set.

A Page Parameter of type Enumeration has several additional properties:

• **Scope**: Default is the current project. You can optionally select Repository, or another project. The selection here determines the types available in the **Work Item Types** list and the enumerations available in the **Enumeration** field's pick-list.

- Work Item Types: The types configured in the selected **Scope** appear in the list. The selection here affects the list of enumerations in the Enumeration field. Depending on the selection it may refine the list and filter out some of the items present as a result of the selection in **Scope**.
- **Enumeration**: The enumerations (default and custom) for the selected Work Item type in the selected scope appear in the list.
- Allow Multiple Values: Lets you select multiple values by the page developer, and later by the page user.
- Allow No Values: Allows you to omit a value for the Page Parameter. The not selected option appears to users in the drop-down list of enumeration values.

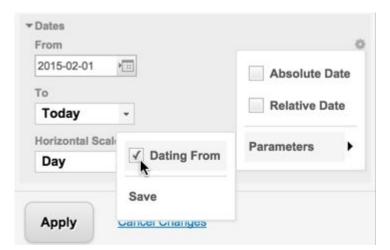
For example, you might want a Page Parameter to enable Page users to specify the specific types of Work Items planned in some Plan for some release... Requirements without Test Cases, for example.

The parameters needed for the example for this topic are:

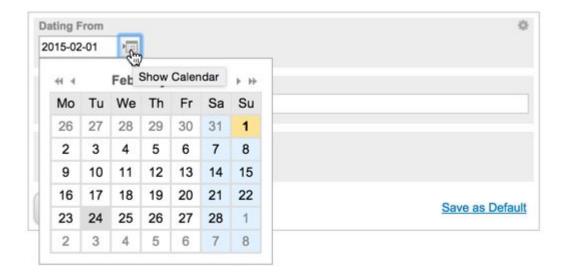
- Parameter labeled **Dating From** of type **Date**.
- Parameter labeled **Show Pie Chart Labels** of type Boolean (such as true/false and yes/no).
- Parameter labeled **Table Rows** of type Integer.

#### **Connect Page and Widget parameters**

After defining the Page Parameters, they must be "connected" to the parameters of the Widgets they manipulate. For example, the Page Parameter Dating From needs to be connected to the Dates -**From** parameter of the multi line trend chart widget:



You can tell if the Widget can be manipulated via an existing Page Parameter by clicking the gear icon in a Widget parameter, and clicking again on Parameters. This displays the Page Parameters that are compatible with this Widget parameter. Check the Page Parameter you want to use. In this example, you can check **Dating From** to connect that Page Parameter to this Widget parameter. The end result for your report users is that they can select the start date in the date range rendered by the chart.



### Tip

Several chart Widgets have Date type parameters. These can be either absolute or relative dates. Relative means the retrieved data is relative to some point in time, such as "Today", "2 weeks ago", or "6 months ago" rather than for some concrete data such as "January 22, 2014".

The report author can specify either or both for a Date type Widget parameter. A Page Parameter linked to such a Widget parameter renders the user's input options accordingly.

The process is the same for the other Page Parameters.

- The **Advanced** → **Show Top Rows** parameter of the Table Block Widget is set to 10 in the Widget parameters. By clicking the gear icon and **Parameters**, you can select the Table Rows parameter, connecting the Widget parameter to that Page Parameter. Report users can then change the number of rows displayed in the table.
- The Show Data Labels parameter of the pie chart Widgets can be similarly connected to the
  previously defined Page Parameter labeled "Show Pie Chart Labels". By connecting both pie
  chart Widgets this way, the Page Parameter will manipulate both. Report users can then show
  or hide the pie chart labels.

#### **Configure the Page Parameters widget**

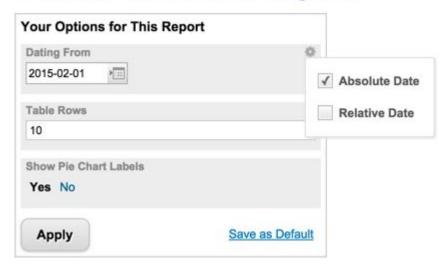
The final step is to add a **Page Parameters - Block** Widget to the Page. Recall that you can only have one of these Widgets on a Page. This is the element that provides visual controls for report users. Like other Widgets, this one has its own set of Widget parameters. For this example, they are set as follows:

- **Title**: This is some short text to provide a hint for report users. For the example, the Title might be "Your Options For This Report".
- Page Parameters: Select which of the defined Page Parameters you want report users to be
  able to see and use. Click the Select button, and in the Select Items dialog, select the Page
  Parameters the Widget should display for the list of defined Page Parameters.

**Width**: The width (in pixels) of the Widget in the report Page.

After saving the Page, users with permission to read Pages are able to view the Page and change the Page Parameters in the Widget to alter what they see in the connected Widgets in the Page.

# 1 Issue Statistics Report



# **Advanced Report topics**

# ALM REQ QA

The visual report building capabilities of Pages are designed to meet the needs of a broad spectrum of users, and to make it easy for nonprogrammers to build useful reports. Advanced users and developers may wish to implement custom scripting or programming in Pages. Two Widgets are provided for such users:

- **Script Block**: Inserts a scripted/coded Widget in the page as a block.
- **Script Inline**: Inserts a scripted/coded Widget in the page inline with text.

When one of these Widgets is placed into a Page, the **Parameters** sidebar opens and displays a simple source code editor where the developer can write or paste source code. Velocity is the recommended scripting language. Macros and variables are supported.

# Security: Best practices for Pages

#### Warning

Poorly written code on a Wiki or LiveDoc ™ (Rich Text) page can leave your system open to XSS (Cross-site scripting) attacks using malicious Work Item and API calls. (Anything, Java code included, can be added to a Work Item in the document and the API will return the actual value of the target fields.)

See Security: Best Practices for details.

## Reuse Widgets and their source code

# ALM REQ QA

A good starting point for custom-scripted widgets may be to reuse an existing widget that is already configured, and modify it with some additional code. You can copy the source code of any existing Widget and past it into a Script Widget, after which you can begin modifying the code.

- 1. Open the page containing the Widget you want to reuse.
- 2. On the header of the Widget, click the \*\* (Actions) icon and select Copy Widget Source Code.
- 3. Add a Script widget to the page and past the source code from your clipboard into the source editor in the Parameters sidebar.

You are now ready to modify the Widget source code as desired.

#### **Note**

Work Item Properties sidebar or drag and drop features:

To use the **Work Item Properties** sidebar or drag and drop features, a unique widget ID must be specified for each item within a selected **Script Block**.

# Script - Block: Parameters





See Making Widget's Work for details about these widget features.

## Use a Page script

Advanced report developers can use the **Page Script** sidebar of the Report Designer to write a Velocity script which can set values to the \$pageContext variable. The page script can use the same context variables as script widgets with following exceptions:

- Use \$renderingContext instead of \$widgetContext. It is class RichPageRenderingContext, which has most methods of RichPageWidgetRenderingContext, but not those related to Widgets.
- \$parametersis not available.

This \$pageContext variable is then available for all Widgets on the Page, but its value cannot be modified by the Widgets or the UI. Typical usage scenarios include:

- Prepare data in a Velocity variable and use that variable in Velocity content of different Widgets on the Page (such as Script block or any Widget with a data set) so that they use the same input.
- Prepare data in a Page parameter and bind that parameter to different Widgets on the Page, so that they use the same input.
- Define Page parameters via this script which cannot be edited via the user interface.

The following actions are available in the **Page Script** sidebar:

- Show Result displays the result of executing the page script, which can be useful for script debugging.
- **Show Log** displays errors and warning logged by Velocity during page script execution.
- Full screen at the bottom of the sidebar displays the script editor in full-screen mode. Press Esc to return to sidebar mode.

### **Note**

The script does not affect any Widgets in the Page until you click the **Apply** button.

Because rendering of the Widgets may be running concurrently, the objects placed in the \$pageContextvariable must be thread-safe if accessed by more than one Widget in the Page.

For more detailed usage information and examples, please refer to the Widgets SDK documentation, included with the Polarion SDK.

### **Organize Widgets with tags**

Widgets can be tagged by categories, and users can easily find Widgets by category. There is set of predefined categories for Polarion Widgets, but widget developers can also set their own categories for Velocity widgets.

Tags for Velocity Widgets are defined in widget.properties via property tags, which can have a comma-separated list of values, for example, tags=Work Items, Planning. Categories are

alphabetically sorted, with the exception of **All**, which is always first, and **Untagged**, which is always last.

### Use the SQL Query type

When you select **SQL** or **SQL + Velocity** as the **Query Type**, an editor displays in the sidebar, and a default SQL query is loaded. Query syntax rendered in gray is not editable. The parts of the statement that you can edit are highlighted in light blue when you click on them. They are just comment lines containing a hint as to what is expected in that specific part of the SQL query.

SQL is generally used only for complex join-type queries that, if done with Lucene, tend to impact overall system performance.

#### Note

If you find that SQL type queries that were working result in errors after you have updated to a new Polarion version, it is probably due to Polarion's transition to a new integrated database. For information about the transition and how to map incompatible queries to the new database, please see <a href="Database Compatibility">Database Compatibility (H2, PostgreSQL)</a>.

### Implement a custom action button

You may want to provide an easy and visually prominent way for Page users to execute some action that is not provided by any Widget. You can use the **Generic Button** Widget, found on the Generic tab of the Widgets sidebar. This Widget can be configured to execute an action or open a URL. The latter is a simple matter of entering a valid URL in the Widget's URL parameter. Invoking execution of any other action requires programming in the Click Handler parameter.

Developer Notes for Generic Button:

- If the Label parameter is empty, the button label defaults to the word Execute.
- Developers can enter Velocity script and use the Polarion Rendering API in Sub-label, Click Handler and Disable Logic parameters.
- The Sub-label parameter is optional and requires script programming. HTML and CSS mark-up are not allowed.
- The click Handler parameter is required if the URL parameter is empty.
  - You can create builder JSActionsBuilder and call methods to create several objects. Basic syntax:
  - \$renderingContext.createJsActionsBuilder(). For more details including actions and parameters
    see SDK documentation at
  - sdk/doc/javadoc-rendering/com/polarion/alm/shared/api/utils/js/JsActionsBuilder.html.
- The Disable Logic parameter is a script field like Click Handler, in which a developer can write code to conditionally disable the button. For example, you might want to disable it in historical views. The script returns reason why the button is disabled, which appears in the button tooltip. If the returned value is an empty string, the button is enabled.
- The Label color accepts any CSS color, which is rendered in the button label.

If the URL parameter is used, the button is clickable in exported PDF. Otherwise, it is disabled in exported PDF.

### Render Portal activities

For some Pages or reports, showing a stream of different activities on the page can be useful. The Activity Stream Widget (Generic tab of the Widgets sidebar) makes this very easy to do. You can opt to show one or more types of portal activities on the page by selecting from a list of sources in the Activity sources parameter. Activity sources include things like Work Items, Builds, Test Runs, etc.

You can show activities from the current project (the default), another project, or entire repository via the scope parameter.

The Query parameter enables you to enter query to retrieve the activities to display. Activity Source-specific fields are indexed with prefixes. These are the fields to be used in the Query parameter: workitem.type, workitem.id, workitem.revision, job.rootJob, job.name, job.worker, job.info, revision.name, build.tag, build.stamp, build.artifact.groupId, build.artifact.id, build.project.id, testrun.revision, testrun.groupId, testrun.type, testrun.status.

Both user.name and user.id are indexed for each activity. This can be used to limit the activity stream to the activity of a specific user. For example, the Query parameter with value etestman will return only activities generated by user "eTestman".

To filter activities according to whether or not the indexed field is empty, use HAS VALUE or NOT HAS VALUE.

### Display the contents of Spaces

You can easily display the contents of any space in any project or in the Repository scope in your Pages. For example, your project might have a space containing some specification Documents, and you want to provide your Page users quick access to them from your Page.

The Space Index Widget renders a table listing the Documents, Pages, and Classic Wiki pages currently existing in a space that you select in the Widget parameters. The name of each listed item is a link that opens the item. Keep in mind that users must have permission to read the items. Any items for which a user does not have read permission are skipped and not shown to that user.

The **Space Index** Widget accesses the same content that is shown on the space's Index page, but renders it somewhat differently. By default, the Widget limits the display to 20 items (configurable), and the items are sorted only according to the date they were last updated, with the most recent listed first.

- Select the desired scope in the scope parameter (required). Items for the current project and the Repository are present in the list. To select a different project, click **Select** and choose the desired project in the dialog.
- Specify the name of space in the Space Name (ID) parameter (required). This is usually the name that appears in Navigation.
- In the Show Top Pages parameter (under Advanced) you can optionally change the number of items to show. The default number is 20. Items are sorted by the date they were last updated. To show all items in the space, leave this parameter empty.

# **Polarion for Requirements engineers**

## Polarion and requirements management

Defining and managing requirements with Polarion provides significant advantages over legacy approaches.

- Best of both worlds support. People who are accustomed to a document-centric approach can
  continue to work with documents, while those who need data and tools to manage their work can
  take that approach. No one has to give up much in order to work effectively and efficiently in
  their role.
- Integration of requirements into the overall process. Too often, requirements have been isolated
  in office documents that are decoupled from the processes of implementation and testing. With
  Polarion ALM, requirements are an integral component of the overall development process
  from start to finish.
- More efficient and timely collaboration. All stakeholders can have access to the same version of
  requirements at all times. Changes are reflected in real time no delays waiting for emailed
  copies. The process is integrated and automated so that no steps are missed or skipped due to
  miscommunication. Stakeholders can see the current status, and they are notified automatically
  as changes take place and requirements move forward in the process.
- Easier and more robust traceability. Rigorous and thorough traceability has been tough to do
  with legacy document-based approaches. Even dedicated point solutions tended to make
  traceability an afterthought, and not very visible. Polarion makes deep and broad traceability
  easy to implement, and totally transparent.

#### Note

All Polarion products support basic requirements management. You can have a Work Item type named **Requirement** which can be linked to other Work Items representing things such as tasks. However, advanced requirements management features are present only in **Polarion ALM** and **Polarion Requirements**.

# Approaches to requirements

Traditionally there have been two main approaches to requirements management that have been more or less mutually exclusive:

- Document-centric approach Requirements are defined in office applications like Microsoft
  Office Word or Excel, and engineering and QA work from copies of the requirements, which they
  hope are the right and most up to date copies.
- Tool-centric approach Requirements are defined using data-driven software tools such
  as DOORS, Requisite Pro, and others. Engineering and QA can manage their processes
  more efficiently, but the business and domain experts who author requirements often resist the
  adoption. Extensive training becomes necessary for these users when the tools approach is
  mandated.

Polarion supports both approaches, and even enables a hybrid approach where some requirements and other artifacts are document-based while others are tool-based. Document-centric workers can stay within the familiar document paradigm using LiveDoc Documents to author specifications, and marking parts of the content as requirement artifacts that they and others can track, manage, and trace throughout the life cycle.

### Tip

Artifacts that represent requirements, test cases, tasks, and others are collectively called Work Items in Polarion.

LiveDoc Documents are often referred to as **Documents** in help text and the Polrion user interface.

People such as developers and QA testers who are accustomed to using data-driven tools can keep within that paradigm, using integrated tracker, workflow, and project planning and management features to work efficiently — even when artifacts like requirements live in Documents.

Polarion also makes possible a unique hybrid approach. For example, requirements, test cases, or other artifacts written up in Documents automatically create data in the background, which is accessible in tools and reports. Stakeholders can access their preferred presentation — Document or tools. It's quite easy to switch between these presentations.

### Features to explore

Definitely have a look at Documents. Within that topic, learn about the Word Import feature that lets you leverage existing assets authored in Microsoft Word, and the Word Round-trip feature that enables you to collaborate with external people who use Word, or work on your Documents offline and synchronize your changes with the portal.

## Requirements projects

Your requirements exist in the context of a Polarion project. Polarion provides a Project Template specifically for requirements projects. The template contains preconfigured Work Item types, link roles, and workflow that are suitable for many requirements projects. We recommend that you create a test project in a sandbox area of your system, and explore the requirements project template to determine whether it meets your needs, or if you will need to customize it.

When setting up any project, there are a number of things to consider and plan for. Before starting a new project for requirements, we recommend the topic Planning for a Project.

### Integrating with development and QA

Whether written up in Documents, or authored with integrated tools, Requirements in Polarion are a type of Work Item, that is, an artifact that is tracked and managed through a process. You can customize your project configuration to support whatever semantics you use. For example, if your organization uses the Scrum methodology, you might name the Work Item type that represents requirements as User Story. Regardless of semantics, requirements are tracked and managed the same as any other artifact — development tasks and change requests, or QA test cases and defect reports. Artifacts created by other teams can be linked to requirements to create traceability. For example, QA engineers can link Work Items representing test cases to Work Items of the

Requirement type that the test cases verify. Engineering can link Work Items representing tasks and change requests to the requirement they implement. Work Items representing defects found by QA can be linked to an implementation Work Item that fixes them, and easily traced back to that item's requirement or requirements.

When preparing to use Polarion for requirements, you need to think about how the requirement Work Items you create will be linked to others like tasks and test cases. For example, you will need to decide what processes will be managed in a given project. Will requirements have a separate project, or will they be part of the same project used to manage implementation and testing? Maybe Requirements and QA will have one project, and Engineering a separate project. All approaches are possible. Linking for traceability across projects is a little more involved than when all artifacts are managed in the same project, but it is still quite feasible. In any case, your requirements team will need to coordinate with these other teams to decide the best approach to projects.

## **Elicitation phase**

During the requirements elicitation phase there is typically a need for communication and collaboration, and for tracking the results of these, to finally come up with defined requirements. The following features can be used during the elicitation phase while you are still gathering input to process into actual requirements:

- Documents Create Documents in the portal, or import Microsoft Word documents. All stakeholders with access to your project can access Documents and collaborate on the content, with changes showing up in real time. For information, see Documents and Pages, and the topic on importing from Microsoft Word.
- Word Round-trip If you have stakeholders who do not have access to Documents on your Polarion portal, you can use the Word Round-trip feature to share Documents with them. They can comment and/or modify content, depending on the permissions granted when exporting. Changes can then be merged into the source Document in Polarion.
- **Pages** You may find **Info** type **Pages** useful for collecting and collaboration on information and content that will eventually become requirements that you copy to LiveDoc Documents, or that you create in the integrated **tracker**. For more information, see Working With Pages.

## **Definition phase**

Polarion has several features that you may find useful during the phase when requirements engineers are actually defining requirements, prior to their review and approval/acceptance.

- **Documents** If you prefer a document-centric approach to Requirements, then use Documents to create specification documents and requirement artifacts for tracking and management.
- Integrated tracker If you prefer a tool-centric approach you can create requirements artifacts directly in the integrated tracker: Navigation Work Items. Your project should be configured with the Requirement Work Item type, or whatever type corresponds to a requirement in your process and semantics. If you take this approach, we recommend you read the topic Manage Polarion Work Items.
- Auto-assignment This feature can save a great deal of work over the life of the project. You
  can configure Auto-assignment in your projects so that new requirements are automatically
  assigned to one or more owners. If your process involves parallel work on test cases for

verifying requirements, you can configure Auto-assignment for new requirements to include the responsible QA person as an assignee. For information, see Configure Auto-assignment.

**Linking** — You can begin creating traceability even as you define requirements. For example, if you or your QA team develop test cases in parallel with requirements, test cases can be linked to the requirements they verify. Likewise, if requirements you define involve change requests for your development team, you can create those artifacts and link to them to the appropriate requirements so that you have traceability right from the start.

You can do some linking of requirements within Documents. See Link a Document's Work Items. You can also link Work Items in the tracker (Navigation - Work Items), in the Table view or the Matrix view. For more information, see Linking Work Items.

## Approval phase

Polarion makes the process of approving requirements simple and efficient. The following features are the ones to take a closer look at to support this phase:

**Workflow** — Review and approval is just one or two steps in a larger process. That process is defined in the Project's workflow configuration. Polarion comes with several Project Templates that support requirements elicitation, authoring, and management. These have preconfigured Document types and Document workflows suitable for many requirements projects. If the defaults don't reflect your exact process, Project Templates, and/or individual projects can be customized by administrators to map your process into the workflows of both Documents and Work Items.

When your process is mapped into the project workflow, then people always know the current status of a requirement, and what the next step is. For example, when a requirement is created. workflow might assign it a status of **Draft**, and define the next possible action as **Send to review**. When a user invokes that action, the requirement transitions to a new status — **Under review**, for example. Then, the workflow might present possible next actions such as **Approve** and **Reject**.

Polarion automatically sends notifications to the relevant stakeholders when transitions occur. The workflow defines what happens when any action is invoked. For example, the **Reject** action might set the status back to **Draft**. The change can trigger a notification to the author, the assignee, and others. The **Approve** action could set a new status like **Approved**, also resulting in notifications.

The foregoing is a relatively simple scenario. It is also possible to set up Document-specific workflows that make Document content read-only when the Document has a given status — In **Review** or **Approved**, for example. It is also possible to configure the workflow to require an electronic signature by users transitioning a Document from one status to another, and/or to provide an electronic signature when approving an entire Document.

**Auto-assignment** — Again, Auto-assignment comes in handy. For example, you could configure Auto-assignment to reassign requirements to someone when the status changes — to **Approved**, for example. Possible assignees might include a developer in engineering, and/or a tester in QA. Again, appropriate notifications are sent when the status changes, so everyone who must do something with approved requirements automatically knows it's time to move forward.

### Tip

Be sure to have a project administrator look at the Notifications configuration to ensure that your scheme of notifications works the way you need it to work and notifications always go to all the right people for every system event, such as the status of a requirement changing.

## **Common operations**

## **Locate requirements**

Requirements are defined in the context of a project, so generally the first step in locating requirements is to open the project that contains the requirements you want to work with. For information, see Access projects.

In projects, requirements may be defined in the context of **LiveDoc** Documents, or they may be defined directly in the integrated **tracker** (**Navigation Work Items Table**), or a combination of the two approaches may be used. The simplest way to locate the requirements in a project is as follows.

- 1. In navigation, expand the **Work Items** topic. The Work Item types currently configured for the current project appear as child nodes.
- 2. Click **Requirement**. The Table views becomes current in the Work Items page and the table contains all the Requirement type Work Items in the project whether they are defined in a Document or directly in the tracker.

Work Item types are configurable to support the semantics of any process. In the above steps, if you don't see **Requirement** it is because your project has been customized with different types. You would click on whatever type corresponds to requirements in your organization or project. For example, if your process is some variant of the Scrum methodology, you might see **User Story** in Step 2 above rather than **Requirement**.

If a requirement you have located is defined in a LiveDoc Document, and you want to edit or view it there, you can easily open the containing Document. When selected in the top section of the page, the **Edit in Document** button appears in the Viewer/Editor toolbar in the lower part of the Work Items page. Click this button to open the associated Document and go directly to the selected requirement in it.

### **Access requirements**

Once you know how to locate requirements in a project, you will most likely want to access some or all of them. At a minimum, you need the permission to view Work Items for the project containing requirements. If you are unable to access requirements, or perform some operation with them, contact the project leader (usually listed on the project's Home page), or the project administrator to review your user permissions.

### Query for the type of Work Item

In a **LiveDoc** Document that contains requirements, you can use a query to filter the content to show only the requirements. See Filter Work Items in a Document.

In any context where the user interface provides the possibility to enter and run a query, you can isolate Work Items corresponding to requirements using the following query:

type:requirement

Where requirement is the ID of the Work Item type corresponding to a requirement in your specific project.

### Tip

Most contexts where you can specify a query provide the Visual Query Builder tool, which enables you to construct simple and complex queries without having to know the underlying query language and syntax.

### Legacy tool: Module

Special containers for requirements called **Modules** were provided in Polarion versions prior to version 2011. Modules appear in that and subsequent versions if they existed prior to the upgrade. It is recommended that users convert any existing Modules to the current LiveDoc Document format. Contact Polarion Technical Support if you need to convert legacy Modules to LiveDoc Documents.

### Legacy tool: 2010 Documents

Polarion version prior to version 2011 provided templates for creating Microsoft Office Word and Excel documents that could store Polarion Work Item data in them. That technology had limitations imposed by the document format, and it is now deprecated in favor of LiveDoc Documents.

If you have legacy live documents created with Polarion versions 2010 and earlier, in Word or Excel format, these documents still exist in the repository but they are no longer indexed, and consequently do not appear in query results. If you still need the legacy documents in that format, and you want them indexed, please consult Polarion technical support.

### **Create requirements**

### Note

To create new requirements, you must be granted the permissions to view and to create new Work Items in the project.

Your approach to creating a new requirement depends on the approach to requirements you have decided to use with Polarion.

If you have opted to define requirements using **LiveDoc** Documents, see Work Items in Documents. You can automatically create requirement artifacts in Polarion when you import a Microsoft Word document. See Importing Word Documents.

You can optionally create requirement artifacts directly in the project's integrated tracker. Remember that the Work Item type **Requirement** is a default type for some project templates, and that some other type corresponding to requirements may be define in your project configuration.

1. Open the project (see Access Projects).

- 2. In Navigation, click Work Items and select the Table view.
- 3. In the Table view's toolbar, click **Create** and choose **Requirement** (or the custom type that corresponds to a requirement in your project). A new Work Item is created.
- 4. In the Work Item Editor (lower half of the Table view), edit the data fields as needed and click the **Create** button when finished.

## **Modify requirements**

How you go about modifying existing requirements depends on how they are stored: in a **LiveDoc Document** or in the integrated **tracker**. Modifying requirements defined in a Document is very straightforward; simply open the **LiveDoc** and edit it as you would any office document.

Requirements created directly in the integrated tracker must be edited using that tool. Generally the Table view is best for changing text content or data fields. The tracker is especially useful if you want to edit the same field for multiple requirements, for example, to set the same status or assignee value for multiple requirements. You can use the **Bulk Edit** feature for that.

- 1. Locate the item as described in Locate Requirements. Select the desired requirement in the **Table** view of the **Work Items** topic.
- 2. Edit the requirement in the Work Item Editor, located in the bottom portion of the Table view. Some data fields can be edited in place, that is, without invoking Edit mode on the entire Work Item. To edit the entire requirement, click the Edit button in the Work Item Editor toolbar in the Table view.

### **Link requirements**

Traceability — from requirements to implementation tasks and source code, to tests, to defects and fixes — is an important issue for many organizations, especially those in industries where rigorous regulatory mandates must be met, and compliance must be verifiable. Polarion makes this kind of deep and broad traceability both easy to achieve, and highly visible. For many projects, the process starts with the requirements.

The key to success is timely linking of requirements to others, and to other artifact types representing such things as QA test cases and defect reports, and engineering tasks and change requests. As a requirements engineer, you will probably be most concerned with linking requirements to other requirements, and possibly to test cases. Other teams may link their artifacts to your requirements as well, depending on your organization's traceability needs. As with other common operations, you approach to linking depends on your basic approach to requirements: Document or tool-based. This topic points you to the features you need to link requirements for both approaches.

## **Linking in Documents**

You can link the Work Items defined in a Document to other items in the same Document, in a different Document, or stored directly in the Tracker. See Linking a Document's Work Items.

### Linking in the Work Item Editor

The table view of the **Work Items** topic, and the same view of a Document, provide an editor for Work Item data that you can use to link a requirement to any Work Item of any type. The link can be to Work Items in your project, or in another project. Linking to Work Items in a different repository is also possible, but that is an advanced issue, and is not needed for most requirements engineering projects.

The **Linked Work Items** section of the Work Item Editor provides a graphical interface for linking a selected requirement with another Work Item. A picker dialog box is available with integrated querying, including graphical query builder, that you can use to locate the target Work Item for the link.

To familiarize yourself with linking this way, see Linking Work Items.

See also: Linking via the Matrix View.

### **Export requirements**

In Polarion, requirements are a type of Work Item. Consequently, you can use any of the Work Item. export features to export requirements to other formats. For information, see these topics:

- Export Work Items (covers all export features).
- Share Documents using Word Round-trip (covers Word Round-trip import/export for Documents).
- Export Documents to RegIF.

### **Exchange requirements**

Polarion supports the exchange or interchange of requirements specifications with external customers or suppliers who use RegIF and/or RIF technology. You can import RegIF or RIF files to new or existing Polarion LiveDoc Documents. Conversely, you can export LiveDocs to new or existing RegIF or RIF files.

For information, see Use RegIF Import/Export.

## Review and approve requirements

If your job is to review and approve requirements developed and managed in Polarion, you need to know how to locate the requirements you need to review, and how to mark them as approved or rejected. There is no universally applicable procedure. What you must do depends on your organization's development process, and how that process is mapped into Polarion's workflow. Assuming that has been done, then you should receive email notifications about requirements that need your attention. This will most likely happen in response to a status change in a requirement type Work Item: from **Draft** to **Awaiting Approval**, for example.

Polarion fully supports a formal review and approval process. Workflow can be customized with the necessary statuses and transition actions, as well as automated assignment of items to those responsible for review and approval. Any type of Work Item, including requirements, can have an approval process built into its type-specific workflow.

For information on how to review and approve/disapprove Work Items, see Approve or Disapprove Work Items.

## **Advanced topics**

## The Traceability Matrix

# ALM REQ QA

The **Matrix** view of Work Items enables you to link multiple Work Items of two different types in a single operation. This approach can be useful when traceability linking needs to be managed

centrally rather than relying on individual users to link Work Items as they process them day to day. It can also be useful for correcting incorrect or inappropriate links. For a brief introduction to the Matrix view, see Matrix View. Linking with the Matrix view is a topic for Advanced users. To learn how to use this feature, see Linking Work Items in the Matrix View.

Most people will only need to link individual requirements as they create or process them. To learn about this approach, see the topic Link Work Items.

## Traceability reporting

# ALM REQ QA

You can use the **Matrix View** of the Work Items to generate traceability data which can be exported to spreadsheets or printed as hard copy. For example, you can easily create a report that shows which requirements have links to test cases, or the analogous artifact type configured for your project.

The following procedure illustrates how to create a report according to this example.

- 1. In Navigation, select **Work Items**, then in the view selector on the Work Items page, select **Matrix**
- 2. In the **Rows** list, select **Work Items**, then use the Visual Query Builder to construct a query that returns the requirements you want to export. For example, the following query returns all resolved requirements:

```
type:requirement AND HAS VALUE:resolution
```

3. In the **Columns** list, select **Work Items**, then use the Visual Query Builder to construct a query that returns the test cases you want to export. For example, the following query returns all resolved test cases:

```
type:testcase AND HAS VALUE:resolution
```

- 4. Click **Search**. The matrix shows which requirements are linked to test cases, and just as importantly, which ones are not linked.
- 5. Click at the top and choose **Export** if you want to save the report to a Microsoft Office Excel sheet, or **Print** to send the result to a printer.

### **Note**

If requirements and test cases are maintained in different projects, the queries would need to be run in the scope of a project group, provided that the requirements and test case projects are in the same project group. If they are in different project groups, then the queries need to be run in the repository scope. In the latter case, you would need to create the necessary queries to retrieve just the items from the projects you want to report.

If there too many items in the result set, system performance would degrade for all users if Polarion tried to render than all in the matrix. In such cases, Polarion displays a message informing you that it cannot display the items, but that you can still export the results.

### Tip

You might explore the possibility of using LiveReport Pages to build traceability reports.

## Polarion for project managers

## Planning to start a Project

## Overview of Planning to start a Project

This section covers information about Polarion features and common tasks that are of interest mostly for project managers. Of course some features and topics may overlap with information for software developers or system administrators. In such cases, this may contain only a cross reference to information elsewhere in the documentation.

When you are planning to manage a project with Polarion, there are quite a number of issues to think about. The prefabricated project templates that come with the different products are a good way to get many projects up and running quickly. But other projects may require more thought, planning, and customization. Where do you begin, and what issues should you think about and explore? This section deals with exactly that. Not every issue or every customization mentioned may apply to your situation, but you can only decide that if you are aware of the possibilities. So let's begin with an overview of the Polarion features and configurations to consider when gearing up for a new Polarion project.

### **Project templates**

Polarion products come with one or more **project templates**. These are carefully designed to provide artifact types, workflows, system configurations, and content that support different types of projects. For a list of the default project templates distributed with different Polarion products, see Default Project Templates.

In addition to the default project templates included in product distributions, the Polarion Extensions portal has more project templates, which you can download and add to your Polarion installation. You can find these at http://extensions.polarion.com.

Looking into project templates should be a project manager's first step in planning for a new project. There is a good chance that one of the available templates provides everything needed for your project. A good way to check this is to create sandbox projects on your own Polarion server. Create test projects using the templates that seem closest to what you need, and then look at them in terms of the issues discussed in the following sections to see if the template provides what you need. If the template doesn't provide what you need, determine what you might need to customize to get you there.

To learn how to create a project using a project template, see Create a New Project From a Template. To learn about customizing project templates, see Customize Project Templates.

### **Work Item Types**

The default project templates provide predefined Work Item types that work well for many software development projects. You see such types as "Requirement", "Defect", "Change Request", and "Test Case". But keep in mind that Work Item types can represent literally anything you need them to represent.

In your planning, think about what kinds of artifacts you need represented and processed, and consider creating custom Work Item types for them. Work Item types are defined in the enumeration configuration file workitem-type-enum.xml ( Administration Work Items Enumerations). For more information see Configure Enumerations.

#### **Work Item Link Roles**

Hand in hand with thinking about custom Work Item types, you need to think about what relationships can exist between items of your custom types. The standard types provided for software development have relationships like **depends on**, **relates to**, **implements**, **verifies**, and so forth. These relationships are defined in **link roles** and can be customized to define whatever relationships you need between your types of Work Items. Then in production, links with one or more link roles can be created between Work Items of your types, thereby creating exactly the kind of traceability you need. Link roles are defined in another enumeration configuration file: workitem-link-role-enum.xml ( Administration Work Items Enumerations). For more information see Configure Enumerations.

### **Link Role Rules**

Let's assume you defined several custom Work Item types and the link roles that cover all the relationships that can exist between these types. Some of those relationships should actually exist only between certain types, and never between other types. For example, you have a link role "fixes" for a relationship that can exist between Work Item types "Repair" and "Breakdown", but should never exist between Work Item types "Breakdown" and "Sales Call". That's where **Link Role Rules** come into play. For each link role you define, you can also define one or more rules that control between which Work Item types users can create links with that link role. You can configure the rules on the same page where you define the link roles.

For more information, see Link Roles and Rules.

### Workflow

Regardless of whether or not you need custom Work Item types and link roles (but especially if you do), review the workflow customization capabilities for Work Items, Documents and Test Runs. The Project Templates provided with the different products provide workflow configurations that are suitable for many projects and processes. If the predefined workflows in the project templates do not meet your needs, your administrator can configure the workflows for your project to support your process.

Workflow enables you to embed process knowledge in the project to ensure that everyone on the team complies with the process, even though they may not know all its details. Workflow makes process compliance seamless and painless. You can set up workflows to ensure that no steps are missed, that the steps progress in the correct order, and that the right things happen after steps in the process are completed. A good example is the workflow configured for the standard Requirements Project template provided with some products. Work Items of Requirement type go through a process or review and approval before they go into the implementation phases. After that phase, they automatically enter the testing phase. Workflow can be circular. For example, if a Requirement is not approved, it reverts back to an in progress state; or if a test discovers a defect, it reverts back to an unresolved state.

### **Electronic Signatures**

Polarion supports U.S. 21 CFR Part 11 compliant electronic signatures. Requiring signatures for specific actions or process steps can be configured in the global workflow configurations for Work Items, Documents, and Test Runs, (as default for all new projects) and individually in these workflows in individual projects, overriding the global configurations with project-specific statuses and transitions.

### **Custom Work Item fields**

Polarion provides a default set of data fields for Work Items which are adequate for many teams and projects. But if you need to track additional data with Work Items, you are not limited to the default fields. You can define custom fields of various data types. You can make them applicable to all Work Item types, or limit them to one or more specific types, and you can optionally define them as required fields. For example, if you have several products and Work Items are somehow applicable to a specific product, you might create a custom field "Product". You could also create an enumeration to store a list of products from which users can pick a value when filling in the custom field. Custom fields can be used as search parameters in queries. For example, if you have a custom "Product" field, users can guery for Work Items that contain a value in that field.

So in planning for a new project, give some thought to what additional data you need to track for each Work Item type and plan to create custom fields to track it. You define custom fields in Administration Work Items Custom Fields. For more information see Configure Custom Fields.

## **Automatic assignments**

You can configure your project to automatically assign newly created Work Items to team members based on conditions which you specify. This can save a great deal of time over the life of a project and more than pay back the time spent to set up the auto-assignment rules. Conditions can range from simple (for example, assign all new items to the project leader), to highly complex where assignment depends on multiple conditions and values set in several Work Item fields. For example, a complex condition might be an item assigned to some Category with a certain severity level, and it is assigned a specific Time Point. When the condition is satisfied, then the new Work Item is automatically assigned to one or more assignees specified in the auto-assignment rule. Auto-assignment rules might be created so that a new Work Item with Category value "User Interface" with a severity level "Major" or "Critical" is automatically assigned to user A, while an item with the same Category value but with severity level "Normal" or "Trivial" is automatically assigned to user B.

Automatic Work Item assignment is configured in Administration Work Items → Auto-assignment. For more information see Configure Auto-assignment.

### **Categories**

The standard Work Item field **Category** enables you to classify Work Items in one or more ways. Categorizing your project's Work Items provides an additional way to query them. This is useful for reporting certain subsets of a project's Work Items on wiki pages for stakeholders who do not use the tracker. For example, you might have a category "Usability" for Work Items that somehow relate to that issue. You can then easily construct a query to retrieve all items in the Usability category set for them and that were resolved for some time point. You can easily embed that query in a wiki macro that displays the query results on a wiki page. Categories can also be useful in automatically assigning newly created Work Items.

You can define Categories for your project in Administration Work Items Categories. For more information, see Configure Categories.

### **Users and permissions**

To make sure your project launches without issues, you should make sure that everyone on your project team has a user account and that they have the permissions they need to work on the project. The same goes for other stakeholders. You may want to look at the global user roles on your Polarion system and decide if these are right for your project, or whether you need to define project-local roles. For such roles, you need to consider what permissions are needed for users in each role. For example, suppose you have a project-local role **project\_customer** for some external user(s) at your customer's company. You may want that role restricted from accessing some kinds of data, for example, Work Item comments or time estimates, and also restricted from changing data.

To learn more about these issues, see User Management in Polarion.

## **Project Plan**

## **ALM**

Polarion's approach to creating a project plan is probably different from that of other project planning tools you may have used. With Polarion, you create Work Items for everything that must be done to complete the project. These are assigned to team members to process. You create Time Points corresponding to milestones like iterations and releases. As part of your planning, assign Work Items to these. On each Work Item, you and/or team members estimate the amount of time needed to resolve it. You also need to check your available working time as configured in the global Working Calendar (open Repository, then Administration Work Items Working Calendar) and have your team members configure their personal Working Calendar in their user account page.

With this data in place, Polarion takes over and automatically builds a project plan which is displayed in the form of a GANTT chart in the Dashboard topic and the Live Plan view of the Work Items topic. There you can immediately see if the project looks realistic to complete in the time points and who is over or undertasked in what time frame. You can then make adjustments to mile stones and time estimates before the project is implemented. As team members resolve Work Items during the course of the project, the Live Plan is updated automatically. You can always tell if the project is on track, lagging behind, or at risk of getting bogged down in the near future.

You should read the sections on Project planning. Some aspects of the Live Plan project planning engine are configurable, so you may also want to review Configure Planning. If you have stakeholders who use Microsoft Project, or you have a project defined with that tool that you want to import into Polarion, see Import from Microsoft Project and Export to Microsoft Project.

#### **Notifications**

The last consideration when getting ready to launch a new project is Notifications. These are configured globally, and the scheme is inherited by new projects. In most cases, your system's global default scheme for notifications works to begin your project. If you find that you need project-specific tweaks, you can modify the project configuration after the project is under way. For example, if you find that under the global notification scheme some notifications are not really necessary in the context of your project, you can reconfigure the notification scheme in the project scope. If some notifications are of vital importance, then you should review the global notification scheme

(open Repository, then \*Administration \*\*\* \*Notifications \*\*\* Events and Targets - global administrator permissions required for access). If you find that you need to make changes for your project, perform the configuration in the same Administration topic of your project.

For more information see Notifications, Events, and Targets.

## **Project template for Agile process**

If your team uses or wants to move to the Agile methodology, Polarion can provide significant benefits in terms of process support and embedded knowledge. Some Polarion products come with a project template preconfigured to support Agile software development. You can you can use this template as the basis for your own projects. Simply create a new project, and select the template during the process.

For a list of the default project templates distributed with different Polarion products, see Default Project Templates.)

### Agile software Project template

# ALM PRO

Based on the Agile methodology, this template provides standard Work Item types of the Scrum style (User Story, Task, Defect, Change Request), workflow that supports the process, key artifacts already defined, templates for backlog and sprints, plus some embedded knowledge in the project Wiki that can be helpful to new and experienced developers alike.

If your process is basically Agile but with some modifications to suit your specific needs, you can modify this project template or create a customized copy of it, changing workflows, reports, and so forth to map your exact process to Polarion ALM.

## **Project Administration**

Project managers can be granted administrator permissions for specific projects by the lead Polarion administrator. This gives a project manager the ability to manage users and roles as well as customize the system configuration specifically for the project. If you are granted project administrator permissions, you review the relevant topics in the Administrator's Guide.

### **Lock Work Records**

The Work Records feature enables team members to create records of blocks of time spent on specific assigned items. If your Polarion product supports Work Records and your process uses this feature, you may wish to freeze reporting periodically. For example, you may freeze reporting weekly, monthly, or quarterly in order to understand the costs for some period of time. You do this by configuring a locking date for Work Record in the project's Administration interface. Work records dated on or before the configured locking date cannot be modified or removed. For information, see Configure Work Records.

## **Manage Project planning**

### **Overview of Manage Project planning**

## **ALM**

Polarion implements a time-box approach to project planning. It is not necessary to spend time tracking down the progress on tasks and issues, and then update a project planning tool with the latest data. With Polarion, managers and teams concentrate on planning which Work Items will be completed for a specific time point and marking the Work Items accordingly. As Work Items are completed, Polarion keeps track of the project status automatically. Project managers and others can always check the status by querying the system for Work Items resolved for a given Time Point (or other time period), or Work Items still not resolved. Queries can be embedded in wiki pages, making it easy for anyone to check the status at any time. This approach works with all Polarion products.

The Live Plan view of the Work Items topic rolls up the latest project status information and presents it as a GANTT style chart. The Live Plan makes it easy to spot over and undertasked people, identify dependencies and potential bottlenecks, and understand if the project is on track or at risk.

## **Setting up for the Planning Engine**

# ALM PRO

Setting things up so the Live Plan engine can render your project plan actually starts when you create a Project in the Administration interface. At that time you specify the outer parameters of the project: the starting and ending dates. The start and end dates are the foundation on which the Live Plan engine builds the project plan. The project ending date is not a fixed, permanent date. You can change it any time as the project progresses, extending the project or moving the end date up.

Another important system configuration that affects project planning is the Working Calendar. It tracks how much overall working time is available. The Working Calendar defines both working and nonworking time such as weekends and holidays. A Polarion administrator should configure the global calendar. The global calendar generally specifies the organization's normal work policy such as the days and hours when people work, as well as days off for weekends and holidays. The global calendar becomes the default for all new projects. Users can configure a personal Working Calendar that defines an individual's work schedule. For example, some people may work only part time. At times, people may work overtime, take vacation time, or take other time off. Between the global calendar and the aggregate of all individual calendars, the Live Plan knows how much total working time is available, and it takes this into consideration when planning Work Items in a given time frame. For information on configuring the global calendar, see Configure the Working Calendar. For information on configuring users' individual calendars, see Configure Your Working Calendar.

If you have team members who regularly split their working time between two or more projects, an administrator can configure time splitting in their user accounts. This causes the Live Plan engine to account for the available time in the relevant projects. For more information, see Configuring User Time-splitting.

Time Points are a key feature for project planning and release management. These are essentially named milestones associated with a date. For example, you might have a Time Point named "Beta 1" that falls on some date prior to the actual end date of the project. If you practice iterative development, you can set up a Time Points for each one of your iterations. Once Time Points are configured for the project, Work Items can be assigned a Time Point. It is then easy to query and

view items that are unresolved for any Time Point. The Live Plan engine can plan items according to these milestones, and you can review them in the GANTT chart for any Time Point. For more information on working with Time Points, see the Time Point field. To learn how to configure Time Points, see Configure Time Points.

With the foregoing configurations in place, you can begin creating Work Items in the project and providing values in the Work Item fields related to planning. These include Priority, Severity, Initial Estimate, Remaining Estimate (estimate of time remaining to complete the item), Due Date, Time Point, Due Date, and Planning Constraint. For more information on these fields, see Set up Work Items for Live Plan.

With Polarion, you don't build a project plan in the traditional sense. Rather you concentrate on defining the Work Items that must be completed in the project's time frame, and for the Time Points within that time frame. You enter planning data for each Work Item either as it is created or later in planning meetings. Polarion's project planning engine takes care of building the project plan which you and others in the organization can see online as a GANTT style chart, and optionally export to Microsoft Project. The Live Plan is updated as developers resolve Work Items, or as new Work Items are added and/or estimated, or as time estimates change. The latest state is always available online. So if you choose the export route, keep in mind that you'll want to do it at regular intervals to keep abreast of the progress.

## Tip

Once you have the project's Work Items defined and planned, you can always run queries to see what items are open or resolved for any period of time, even if your Polarion product does not have the Live Plan chart.

## "Dummy" Project planning

Here is a possible approach you could take to set up Live Plan, especially with a new project. You can use Polarion with Live Plan to create different "what if" scenarios for the project before deciding on the actual scenario on which to start development.

In this approach, you create Requirements and linked Work Items for implementation, setting planning fields for the latter. But you don't initially assign the Work Items to developers. Rather, you (or your Polarion administrator) can create a set of dummy user accounts that parallels the people and roles of your actual development team. As you set up your planning data on Work Items, setting time estimates and planning constraints, you can assign each item to one of the dummy users. The end result is a Live Plan that exactly corresponds to your team. You can look for bottlenecks, under-resourced areas, dependencies, and so forth. If you identify problems, you can adjust planning data on relevant Work Items and review the effect of the changes on the Live Plan.

Once you have a Live Plan that you feel you can begin work on, you assign the Work Items to the appropriate real developers. You can do this fairly quickly using the Bulk Edit feature to assign multiple Work Items to one developer in a single operation.

### **Update the Live Plan**

# ALM PRO

The Live Plan is rebuilt on the server at scheduled days/times controlled in a default job named **Live** Plan Chart Refresh. The exact firing of this job is specified as a CRON expression. You can see

the currently defined CRON expression in the project's **Monitor** topic in the **Scheduled Jobs** section of the page. Your Polarion administrator can change the days and times when the Live Plan is updated. See Configure the Scheduler.

### On-demand update

You can invoke immediate rebuilding of the Live Plan by clicking **Update** in the Live Plan section of the **Dashboard topic** of your project.

Rebuilding does place processing loads on the server which could affect performance for all users for the duration of the update. If a project is relatively small, this may not be an issue. For very large projects, you may want to try to invoke the update during times of less demand for server resources.

### Conflict resolution in a Live Plan

Work Items that are forced to be planned with overlaps are reported in the Monitor. If such Work Items are found, the job fails (with status ERROR), but the Live Plan display is updated anyway.

If cyclic dependencies among Work Items are found, then the planning engine cannot continue, the update job fails, and the Live Plan display is *not* updated.

## **Understanding Live Plan scheduling**

# ALM PRO

Before setting planning data and constraints on Work Items, you should understand Live Plan's scheduling prioritization for Work Items.

- 1. Dependencies. Items depended upon by others are scheduled before dependent items.
- 2. Priority
- 3. Severity

All of the above priorities are calculated per assignee.

### Effects of Planning Constraint, Due Date, and Time Point fields

The setting of the **Due Date** and **Planning Constraint** fields also impact the Live Plan.

The impact of the **Due Date** field depends upon the setting of the **Due Date Planning** option in the global system configuration. The effect of the settings of this option are described in **Due Date Planning Option**.

The impact of other fields is as follows:

- **Planning Constraint**: Enables setting of an overriding start or end date. For more detailed information, see Setting up Work Items for the Live Plan in Planning Constraint.
- **Time Point**: If you set Time Point, that is in effect the end date for the item, you do not need to set the End Date constraint. If you do set it and it falls after the Time Point date, the item is highlighted in red in the Live Plan visualization, and its detail tool tip has a red background.

### Set up Work Items for a Live Plan

# ALM PRO

This section covers the fields you may want to set for the Live Plan visualization.

#### Note

The following discussion about setting fields for planning is applicable whether or not you use a Polarion product with Live Plan. You are always able to use a query based approach to monitoring project progress, taking advantage of the wiki to embed Work Item gueries in pages that reveal the current state of things.

## Severity

This is an important field for the Live Plan. It sets the severity level of the Work Item, which is the next thing calculated by Live Plan after Priority.

### **Priority**

This is an important field for the Live Plan. It sets the priority of the item in the plan. Items with higher priority are planned ahead of items with lower priority, all other factors being equal.

## Category

This field doesn't really affect the Live Plan, but it can be useful when adjusting plans as the project progresses, as it is used in the Project Dashboard's Unresolved items by Category. It's most convenient to set this field as you are setting planning fields.

### **Initial Estimate**

This is the amount of time you estimate to implement and close the Work Item. In versions 1.x and 2.x this field does not affect the Live Plan. Managers can set it as a guideline or target for developers. Later versions of Polarion will use this field for automated planning accuracy analysis features.

This field has a specific syntax for specifying time. Click the ? icon next to the field for examples of the syntax. For example, "1d" means one day and "1/2h" means one half hour.

### Remaining estimate

This is an important field for the Live Plan. The length of the item's bar in the visualization is based on the time specified here. Click the ? icon next to the field for examples of the syntax. For example, "1d" means one day and "1/2h" means one half hour.

### **Assignee**

It is not necessary to set this field in order for a Work Item to be planned and appear in the Live Plan visualization. Assigned items are planned by assignee, and each assignee has a separate row in the Live Plan visualization. Unassigned items appear in a separate row labeled with the Project name. So if, in the initial planning stage you don't know who will handle the item, you can still plan for when it should be handled and then assign it later.

For more information about assigning Work Items, see Assign Work Items.

### **Due Date**

This field specified the date when the Work Item is due to be finished. If you do not use a Time Point to define the item's completion date, you can set Due Date. The **End Date** planning constraint (if set) takes precedence over this date in the Live Plan.

#### **Time Point**

As mentioned in the Overview, Time Points are named milestones with a date attached. If you set this field, it means that the Work Item is planned to be finished by the date of the specified Time Point. Setting a time point does not preclude also setting an **End Date** planning constraint (see next section).

### **Planning Constraint**

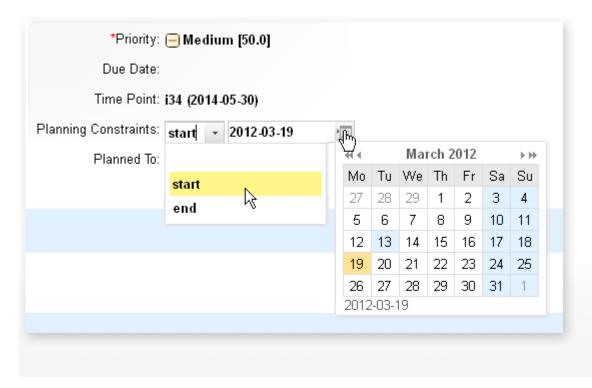
This field is essentially an override to ensure the position of a Work Item at a particular time in the project plan. One of two possible constraints can be set: **Start Date** or **End Date**. If the **Start Date** constraint is set, it overrides other factors such as dependency that would otherwise govern when Live Plan plans the item to start. If the **End Date** constraint is set, it specifies the actual end date of the item, overriding other fields such as Time Point and Due Date.

For example, consider the following two tasks:

- 1. TASK\_1 Implement Feature X
- 2. TASK\_2 Document Feature X

Suppose you have a Time Point named "Release". Both the above tasks need to be planned to finish by "Release". But TASK\_1 actually must be finished earlier to allow time for documentation before "Release". So you can set the End Date to ensure that TASK\_1 is planned to actually finish by the End Date even though it is associated with the "Release" Time Point.

For the field value, you can specify either a date only, or date and time. When time is missing, the task starts at the start of working hours on the specified day or ends at the end of working hours on the specified ending day. If the time is outside the working time of the assignee, the start time is shifted after the nonworking period and end time is shifted before the nonworking period. Working hours are determined by the Working Calendar configuration.



Planning constraint field and popup date picker

### Warning

You do not see relevant planning results for unassigned and un-estimated Work Items.

### Planned items in the Live Plan

When the Planning Constraint and Due Date fields are set, the Work Item spans the defined time period in the Live Plan chart. The popup detail (when you click on the Work Item in the Live Plan chart) includes Planned Start Date, Planned End Date, and Remaining Estimate.

### Planned To

This is a read only field generated by the Live Plan. It shows the date that Live Plan scheduled the Work Item to be finished.

### **Estimate Work Item duration**

### Note

You can set values in the following fields with all Polarion product licenses except Polarion Reviewer. The Live Plan is only accessible for users of Polarion ALM and Polarion PRO.

The process of estimating time duration of Work Items can vary according to how your organization prefers to handle it. Estimating the time duration of tasks could, for example, be the responsibility of a manager. Alternatively, each developer might estimate each Work Item as it is assigned to him or

her, or the project lead can make an initial estimate and individual developers review and adjust the estimate. The process is up to you.

There are two Work Item fields for estimating Work Item duration:

- Initial Estimate: Estimate the amount of time needed to resolve the item. It does not impact the
  Live Plan.
- **Remaining Estimate:** Specify how much time remains before the item is resolved. The Live Plan calculation relies on this field for the duration estimate.

These fields use a specific syntax for specifying an amount of time. Click the ? icon next to the field for examples of the syntax. For example, "1d" means one day and "1/2h" means one half hour.

### The Time Spent field

This field is not for planning *per se*. It does not affect the Live Plan. It is where developers can specify the time they actually spend on the Work Item. This can be useful for later review of planning accuracy (features for this are planned for future Polarion versions).

In the course of working on the item, a developer can update this field as appropriate. When updating, the developer can also reduce the **Remaining Estimate** field by the amount of time spent. This updates the Live Plan, where it is visible to management and thereby eliminates the need for the developer to write a status or progress report.

## **Dashboards and Reports**

## **Overview of Dashboards and Reports**

## ALM

If you're a project lead, project manager, or IT executive, Polarion's live Dashboards provide an information summary that enables you to view the progress of any project. Dashboards present near or actual, real-time aggregate information based on the actual progress of planned requirements, tasks, change requests, defect fixes, and so forth. Of course reviewing dashboards is no substitute for close contact with your team, but Project dashboards and the features that compile the information displayed can reduce the amount of time you and your team spend in meetings reporting on progress, so you can focus on problems and solutions instead. Dashboards can also eliminate the need for developers to write and managers to review and compile information from periodic status reports.

Polarion also provides managers with the ability to generate a variety of reports with output to both online and offline formats.

### **Access a Project Dashboard**

- 1. In Navigation, click on the current project or repository name (under the Polarion logo image), and choose **Open Project**. In the dialog box, in the **All Projects** tab, select the project whose dashboard you want to see. (If you would like to see dashboard information for a group of projects, select a project group instead).
- 2. In the **Navigation** pane, choose the **Dashboard** topic.

To learn more about dashboards what kinds of information are presented in the Dashboard topic, see:

- Dashboard
- Working with the Dashboard

### **Generate Reports**

The Reports topic provides access to a number of on-screen reports and also to Javadoc documentation for source code.

## **Time and Progress Reports**

You can generate and export a report on developer progress and time spent. In order to get useful data, your process should have developers estimating Work Item time. See Estimate Work Item Time and Report Time on Individual Work Items.

The process involves creating a query that isolates the Work Items you want to know about, and then exporting data to an Excel sheet using a specially-designed report template. The next section details the procedures.

## Build a query:

Which Work Items to include in the time and progress report is up to you. You should become familiar with querying for Work Items and using the Query Builder to get the subset of Work Items you want for the report. Run the query from the Table view.

An example of some query parameters might include unresolved items in Project having a Status of in-progress. You can optionally include some specific assignees.

### Tip

Don't forget that you can save any query as a Shortcut for quick reuse later on. See Using Queries and Shortcuts.

### Generate and export:

Once you have a query that isolates Work Items for the report, you can generate and export the report.

- 1. Make sure you are viewing Work Items in the **Table** view.
- 2. Click the **Actions** in the toolbar, and choose **Export** in the drop-down list. The Export Work Items dialog box appears.
- 3. In the Format list, choose xlsx: Microsoft Excel.
- 4. In the **Template** list, choose **Time Report**.
- (Optional) Add columns to the list of Selected Columns to include in the generated report document.
- 6. (Optional) Show advanced options and change any settings as desired.
- 7. Click **OK** to generate and download the report. You are prompted for a storage location.

To review the generated report, open the downloaded report file in Microsoft Excel or other compatible application.

## **Review Baselines and Baseline Reports**

# ALM REQ

It is possible to create Baselines in Polarion which mark a state of Work Items in a tracker (usually a requirements tracker) at some point in time. Later, managers and/or other team members can check the differences between baselines, or a baseline and the current state of Work Items, by generating and reviewing Baseline Comparison Reports. Baselines and reports are covered in Baselines.

### **Generate and review Work Reports**

# ALM PRO

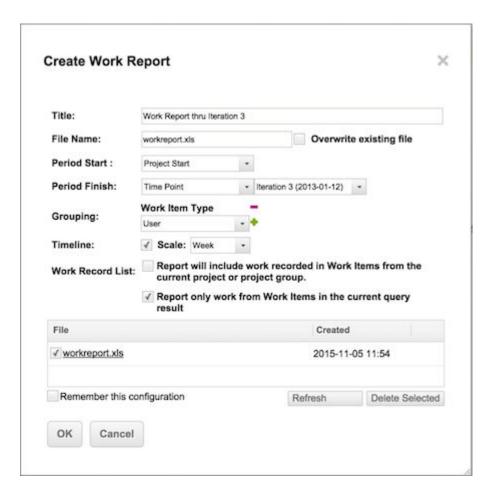
When your process involves project members using Polarion's time recording features when completing Work Items, it is then possible for managers and others to generate Work Reports which summarize all Work Records for a project or a user during a specified time period. For information about the time recording fields and Work records, see Reporting Time and Progress.

### **Note**

Work Reports are generated in Microsoft Excel format.

## Generate a Work Report:

- 1. Access the Project for which you want the report.
- 2. Select the Work Items topic if you want to report on all Work Items, or create a query to isolate some subset of project Work Items.
- 3. On the **Table** view's toolbar, click the **Operations** button, and choose **Create Work Report**. The **Create Work Report** dialog box appears.



- 4. Enter a title for the report in the **Title** field.
- 5. (Optional) Specify a file name for the report you are creating. If a report with the same file name already exists, it appears in the Work Reports section at the bottom of the dialog box. You must either give the report a different file name or check the **Overwrite existing file** option.
- 6. Specify the period you want the report to cover. Specify **Period Start** for the beginning of the period and **Period Finish** as the ending of the period.

You can designate the Start and End of the period in any of several ways:

- As a specific calendar date. Choose the **Date** option and specify a date using the date picker provided.
- As one of the dates configured for the Project, for example, Project Start. Choose the Project
   Date option, and select one of the listed project dates.
- As one of the Time Points configured for the Project. Choose the **Time Point** option and select a Time Point.
- 7. Specify the Scale you want shown in the report. For example, if you choose Week, Work Records are summarized by week within the specified report period.
- Specify the report grouping. You can group the report by any or all of the following:

- Project Group
- Project
- Category
- User
- Work Item Type
- Work Item
- If you want to limit the scope of the report to the Work Items currently shown in the Work Items
  table, check the Report only work from Work Items from current selection option. Clear the
  option to report on all Work Items in the project.
- 10. Click **Start** to send a job to the Scheduler. You can check the progress of the job in the **Monitor** topic. The time required to generate the report depends on the number of Work Items reported on and the availability of system resources.
  - When the report is generated, a link appears in the Work Reports section of the dialog box. The link text is the file name of the report. You can open or download the report file. The Work Reports section contains a **Refresh** button which updated the report from the most current Work Item data.
- 11. If you want to preserve the report settings for future reuse, check **Remember this configuration**.

### Nonworking time representation:

Nonworking time, as configured in Working Calendars, is represented in Work Reports.

- Nonworking time is indicated by gray background. It is displayed only on rows representing users.
- When the scale of the report is Day, cells representing nonworking days of a user have a gray background
- When scale is Week or Month, cells representing a week or month, all days of which are nonworking, have gray background.

## **Automating Work Reports:**

An administrator can configure Polarion to run work reports as a scheduled job. For more information, see Configure the Scheduler and Work Report Job Configuration.

## **CMMI Report**

## **ALM**

Polarion provides a simplified CMMI compliance report based on the CMMI 1.1 Staged Presentation Level 2 (http://www.sei.cmu.edu/cmmi). At this stage, it is not advisable to use this report as input for CMMI certifications. The CMMI report is a collection of pointers on how particular practices are implemented. It can be useful during preparation of a CMMI appraisal, as the report makes it easy to gather the data and provide references to established practices.

The report should be used as guideline rather than required rules. A CMMI report that says 0% is implemented means only that Polarion did not recognize artifacts in the way Polarion defines them. So, for example, a CMMI report might say there is zero Requirements Management because Polarion found no Work Items of type Requirement in the project against which the CMMI report was run. However, you might define and manage requirements in a separate project.

### Access and run the CMMI Report:

The CMMI report is calculated for the current project, so you need to open the project for which you want to see the CMMI report. Open the desired project in the Open Project or Project Group dialog box. See Access Projects.

To run the report, expand the Quality topic in the Navigation pane and select CMMI. The report is generated on demand and displays in the Content Pane.

### **Report Structure:**

The generated CMMI report has the following structure:

```
Key Process Areas + Percentage of Completion
    Specific/Generic Goal + Percentage of Completion
          Practices + Implementation Rating
```

### Implementation Rating codes are as follows:

FI: Fully Implemented

LI: Largely Implemented

PI: Partially Implemented

NI: Not Implemented

N/A: Not rated

# Polarion for developers

# **Access Projects**

When you log on to Polarion, the first thing you normally see is the Repository Home page. (At least this is the case in a default installation. Depending on the way your Polarion administrator set up the system and your system permissions, it is possible you could see something different.)

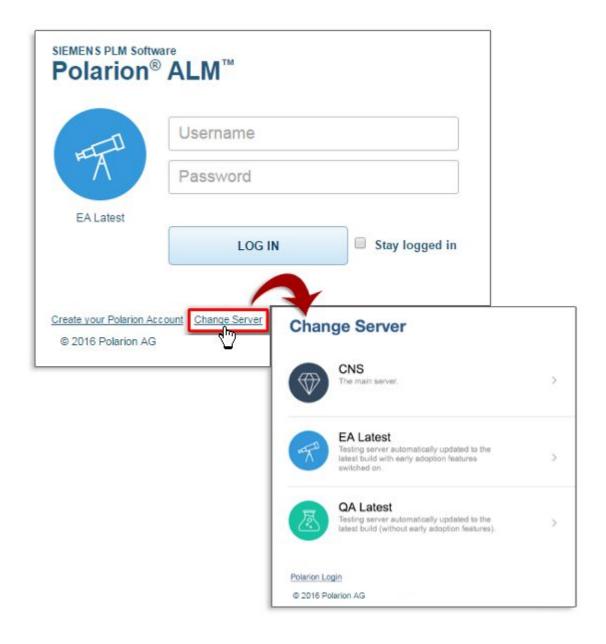
As a developer, you need to access one or more Projects. In order to do that, you need a Polarion user account, but you must also be assigned a role in each Project you have access to. Project roles are assigned by a Polarion administrator. (See Configure User Roles). Once you are set up administratively, navigate to the Project you want to work on.

The projects managed with Polarion are listed in the All Projects tab of the Open Project or Project Group dialog box. Depending on how projects are organized in your system, the project you are seeking may be a top-level node in the dialog box, or it may be under a project group or even a Project Sub-group. (You should be familiar with the organization of your system. For more information, see Access Projects.)

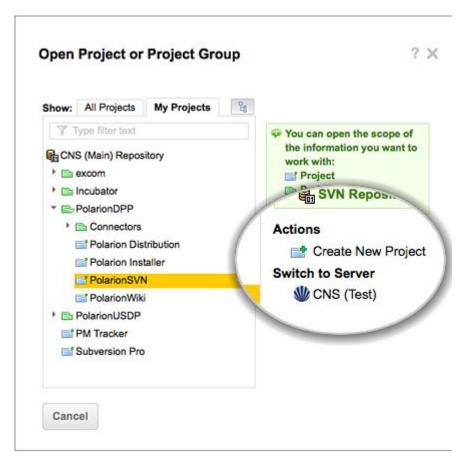
When you locate the project in the Open Project or Project Group dialog box, select it and click OK. The project's Home page begins loading in your browser. You can either wait for it to load so that you can review its information, or you can immediately begin navigation to the Work Items topic or any other topic in the Navigation pane.

### **Note**

Some Polarion systems may be configured to run multiple Polarion servers, and you may be assigned to projects hosted on different servers. On such systems, when you log on you can select which server you want to log onto:



While working in a project on one server, you can switch to a project hosted on a different server. In the topics view of Navigation, click the current project name, and then click Open Project or Project Group. In the dialog box, the available servers are listed under Switch to Servers.



# **Locate your Work Items**

Work Items are issues you and others need to process to complete the project. Work Items are of different types representing different artifacts. For software development, common types include Requirement, Defect, Change Request, and Task. The types and their definitions are fully configurable, both globally and for each project, so you may see different types in projects you work on.

There are three basic ways you can locate Work Items:

- Use Search features.
- Use Shortcuts.
- Use Topics.

### Use Search to find Work Items

If you know the ID of the Work Item you want or some text in its description, you may be able to quickly locate the Work Item using the Site Search in the Polarion header.

### Use Shortcuts to locate Work Items

You can use the shortcuts topics in the main navigation panel to access some subset of Work Items. By default, the Global Shortcuts topic contains a shortcut named **My Work Items** which accesses all unresolved Work Items assigned to you and displays them in the Table view of the Work Items topic. Other shortcuts pull up other sets of Work Items. You can create your own shortcuts from queries you build using the Query Builder in the Table view of the Work Items topic. See Search Work Items.

Once you have accessed a set of Work Items with a Shortcut, you can use the Query Builder in the Table and other Work Item views to refine the list of Work Items. See Search Work Items.

## **Use Topics to locate Work Items**

The **Navigation** pane contains navigational links that pull up sets of Work Items. The **Work Items** topic accesses all Work Items of all types in the project. Beneath this link are several others which may vary according to configuration/customization of your system. By default, the following links are present:

- Task: Accesses all project Work Items of Task type.
- Change Request: Accesses all project Work Items of Change Request type.
- Requirement: Accesses all project Work Items of Requirement type.

#### **Note**

The Work Item types may differ depending on the project template used to instantiate the project and/or any project-specific customization that may have been done.

The list of Work Item types is configurable by a Polarion administrator, so you may see other items such as **Defect** or **Feature Request** which correspond to types of Work Item custom configured for your system. See Configure Work Items. In every case, clicking on one of these items should access all project Work Items of the respective type.

Once you have accessed a set of Work Items with a Topic, you can use the Query Builder in the Table and other Work Item views to refine the list of Work Items.

## **Process Work Items**

## **Configure your User Calendar**

# ALM PRO

The global Working Calendar is used by Polarion's project planning engine to calculate the amount of actual working time available to projects. The Working Calendar defines the working days in a work week, and working hours within each working day. It may also specify special nonworking days and times such as public holidays.

If your organization uses a Polarion product with this feature, you should check your User Calendar before you begin processing Work Items. Your User Calendar tells the system what days and hours you normally work and enables you to report exceptions to that, such as vacation, leave, or other planned time off. This information is used by the project planning engine when creating and

reporting project plans. The global Working Calendar settings, which define the standard working time, holidays, and so forth for your company, appear by default in your User Calendar. You only need to change your User Calendar when your work schedule differs from that defined in the global Working Calendar.

You specify your actual work schedule in your User Calendar. You can also specify planned time off such as vacations or leaves of absence. You can also log temporary schedule differences. For example, if you plan to take several hours per week off over a period of 3 months to attend some classes, you can configure your User Calendar to reflect this.

Anything in your personal calendar that differs from the global calendar is taken into account by the planning engine and reflected in the project plan of each project in which you have Work Items assigned to you. Project managers and teammates can see your working and nonworking time in the Live Plan chart.

Before configuring your User Calendar, you might want to read the section on the Working Calendars feature in Configure the global working calendar.

### **Access your User Calendar**

Your User Calendar is available on your user profile page. Access this page by clicking the My Account link in the tool view of Navigation. Scroll the page until you find User Calendar. Click it to load your User Calendar. You can use the set of graphical controls to make and save changes to your User Calendar.

### **Edit your User Calendar**

The **Regular Schedule** section lists the working days and times of your normal work schedule. By default, the values are from the global Working Calendar. If your regular working time differs from the global schedule on one or more working days, you can change the start and end time for the affected day(s).

- 1. Select the check box beside the day you wish to change. This enables the override of the global Working Calendar. The **Start Time** and **End Time** boxes are now editable.
- 2. Change the Start Time and/or End Time value(s) as desired.
- 3. Click **Save** in the toolbar above the section.

### aiT

You can mark any day in the **Regular Schedule** section as a nonworking day by clicking the icon after the End Time field.

### Revert to the Global Schedule:

Suppose you changed the start and/or end time for several week days in your Regular Schedule because you were working only part time, and now you have started working full time and want to sync your User Calendar back to the global schedule. To do this, clear the check box next to the modified work day(s) and save the change.

### **Define schedule exceptions:**

In the **Schedule Exceptions** section, you can define exceptions to the regular schedule. Exceptions can be permanent, or temporary for a specific period of time. For example, suppose you take time off for a holiday that occurs on the same date every year. This can be set up as a permanent exception to your normal work schedule. If such a holiday falls on a different date each year, it can be set up as a temporary exception.

Schedule Exceptions are of two types:

- **Time Off**: The exception is time that will not be available for work in addition to any nonworking time configured in the Regular Schedule.
- **Time Working**: The exception is time that will be available for work in addition to the working time configured in the Regular Schedule.

### Add a Schedule Exception:

- 1. Enter a meaningful title for the exception in the **Title** field, for example, **New Years Day Holiday**.
- 2. Select the exception type by choosing a value in the **Type** drop-down list, for example, **Time Off**.
- 3. Specify the date when the schedule exception is to begin in the **Date From** field in this format: yyyy-mm-dd. Alternatively, click the calendar icon next to the field and choose a date in the pop-up calendar, for example, 2008-01-01.
- 4. If the exception you are defining is temporary, specify the date when it should end in the **Date To** field, in the same format as mentioned in the previous step. For a one-day exception like our New Years Day example, the value should be the *same date* as in the **Date From** field. If the exception is ongoing, leave the **Date To** field empty.
- 5. If the exception you are defining occurs only on one specific weekday in the time period defined, choose that day in the drop-down list in the **Weekday** column. Otherwise, leave the default value **All days**.
- 6. If the exception you are defining is of the type "Time Working", set appropriate time values in the **Time From** and **Time To** fields, which are enabled when this type is selected. Time From is the time when the additional working hours start, and Time To is the time when the additional working hours finish.
- 7. If you want to add another Schedule Exception, click in the **Actions** column and repeat the above steps on the new row that appears.
- 8. When you are finished defining your Schedule Exception(s), click **Save** in the Working Calendar toolbar. To cancel all changes, click **Revert**.

## Remove a Schedule Exception:

- 1. Click in the **Actions** column of the exception you want to remove.
- 2. Click **Save** in the Working Calendar toolbar. To cancel the remove (before any Save), click **Revert**.

#### View and Edit a Work Item

Once you have located a particular Work Item of interest, you may want to review its details and change one of more of them. See Locating Work Items.

When you have selected a Work Item in one of the views in the Work Items topic (such as Table or Tree) the item's detail appears in the Viewer/Editor pane. On most monitors, at least some of the information is scrolled out of view. The Viewer/Editor toolbar provides three buttons that provide quick-scroll to commonly viewed detail information:

- Comments: Instantly scrolls up or down to the Comments section where you can view all comments for the Work Item.
- Links: Instantly scrolls up or down to the Links section where you can view all the Work Item's links (if any).
- Attachments: Instantly scrolls up or down to the Attachments section where you can view and access any files attached to the Work Item.

### **Edit a Work Item**

The **Edit** button on the Detail pane toolbar triggers Edit mode in the Viewer/Editor pane. When you click this button, all information in the pane is editable, assuming you have write permissions. Make any changes you want and click Save in the toolbar to save your changes. The Cancel button abandons any changes you may have made and restores the Detail pane to Read-only mode.

#### Note

Some fields are editable "in place" without the need to click the Edit button. Hovering over a field for a moment shows if it can be edited in place.

### Link source code elements for traceability

The Resource Traceability feature enables developers to create semantic traceability links between elements in source code (stored in any configured repository) and Work Items. For example, some specific function in the source code might implement one or more Requirement type Work Items. You can create such a link by adding a comment before the function, using a prescribed comment syntax to identify the target item and the link role. Currently supported languages are Java, C, and XML.

For more information about the feature, see Configure Resource Traceability.

For information about the comment syntax to use in source code, see Link a Resource.

### **Estimate Work Item time**

In some development organizations, developers are asked to provide an initial time estimate for Work Items. In other organizations, this is done by a project leader or manager. Polarion provides the capability to estimate Work Items and track the time actually spent to resolve them. Over time, the data collected is highly useful to both developers and managers in estimating time to complete projects and understanding how accurately they can plan.

- 1. Locate the Work Item, and select it to display its detail in the Detail pane.
- 2. In the Detail pane toolbar, click **Edit**.
- 3. Locate the **Initial Estimate** field, and enter the amount of time you think it will take to complete the Work Item.

- 4. Enter the same value in the **Remaining Estimate** field. You adjust this value as you work on the item.
- Click Save to save the edit.

### Time notation syntax

Polarion has a special syntax for entering and displaying time. Using this syntax, you can estimate Work Item time in days (d + integer), hours (h + integer), or a combination of days and hours. You can use 1/2 in front of the day or hour symbol to indicate half.

For some examples, click the ? icon to the right of the **Initial Estimate** field.

Hours as decimal values:

It is possible for a system administrator to override the default factional syntax for hour values, and enable entry and display as decimal values. If this is done, the help dialog box that appears when you click the ? icon appearing with time fields will indicate that entry of decimal hours is possible. If so, then instead of 1/2 h, you can enter .50.

### Report time for individual Work Items

Polarion makes it easy and fast for developers to report their progress to management. There are several fields in the Work Item detail that are related to time tracking and planning accuracy analysis:

- Initial Estimate
- Time Spent
- Remaining Estimate
- Status

In addition to the above fields, Work Items contain the Work Records section. Work Records are useful when you have to spend several blocks of time on different days to complete a Work Item. You create one Work Record for each block of time you spend working on the item. Each Work Record updates the Time Spent and Remaining Estimate fields. You or your manager can then generate a Work Report that shows your Work Records using the **Create Work Report** item on the Actions menu found on the Work Item toolbar.

### **Note**

The administrator of your project may optionally set a locking date, on and after which you cannot add or modify Work Records. If you need to compensate for time over-reported in a Work Record that is locked, you can create a Work Record for the next reporting period with a negative value for time spent.

Your team needs to define the process for using the time estimating and reporting fields and Work Records. You can use these to facilitate progress reporting.

Assuming that the relevant fields are set as discussed in Estimate Work Item Time, there are two things you do as you work on a Work Item:

- 1. When you begin working on an item, go to the **Status** and select **Start Progress** in the drop-down list.
- 2. As you progress with the item, create a Work Record for every block of time spent on the Work Item.

## Work Records in Multi Edit view

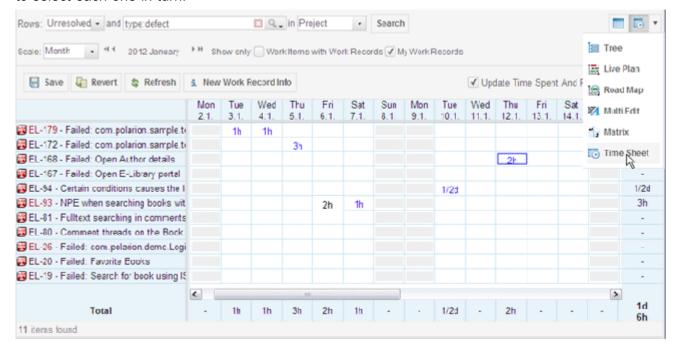
# ALM REQ QA PRO

You can add Work Records in Multi-Edit view. You need to add the **Time Spent** column to the table (see Customize the Work Items Table). When this column is present, a special section is displayed in which you can add Work Records. You can only add new Work Records. Existing work records do not appear and so cannot be modified or deleted.

#### Use the Time Sheet view

# ALM PRO

You can use the **Time Sheet** view in the Work Items topic to report the time you spent on Work Items during a period of time which you specify. You can also use this view to edit time already reported for individual items with Work Records, provided you want to report more time spent on the item than previously reported in the Work Record. If you want to decrease the amount of time already reported in a Work Record, you must edit the specific Work Item in the Work Item form, accessible by clicking on a Work Item displayed in the Time Sheet. The Time Sheet view itself is essentially a convenient and quick way to create Work Reports on a number of Work Items without having to select each one in turn.



### The Time Sheet is comprised of three main areas:

Search Query: Use to fetch a set of Work Items, specify a period of time for which to report time spent, and save or cancel changes.

- Filtering: Filter the set of Work Items according to time, and optionally select only items with Work Records and/or Work Records logged by you.
- Time Sheet grid: Use to report time spent on listed Work Items during the period shown. Rows are Work Items fetched by a normal query.

## Report time using the Time Sheet view:

- 1. Create a query to display a set of Work Items that contains the ones for which you want to report time spent, and to show a specific span of time in which you want to report hours spent.
- 2. Locate the row for a Work Item on which you want to report time spent.
- 3. Enter the amount of time spent in the column for the appropriate date.
- 4. Save the edits when finished.

### Tip

If you regularly split your working time between two or more projects, you can configure time splitting in your user account. This helps increase project planning accuracy in the Live Plan. For information on this configuration see Configure User Time-splitting. You can make this configuration yourself in your own user account. Click **My Account** in the tool view of Navigation.

#### Work Item discussion and comments

You can quickly and easily add any number of textual comments to a Work Item. This can facilitate discussion of the item between team members in the same or a remote location. For example, a developer in Delhi may need some additional information from an architect in New York. The developer can add a comment and reassign the Work Item to the architect, which creates a notification for the architect. The architect can respond with another comment, and perhaps some attachment, and reassign the item back to the developer.

## Once you locate the item you want to comment:

- 1. Click **Comments** in the Detail pane toolbar to guick-scroll to the Comments section.
- 2. At the bottom of the Comments section, click the small **Edit** button. An edit form appears in the Detail pane.
- 3. Type your comment and click **Save**.

You can also edit comments when editing a Work Item.

#### Resolve or Close a Work Item

When you finish working on a Work Item, it's time to indicate it in Polarion. By so doing, the project plan and overall status is automatically updated, and your project manager can follow your progress without the need for you to write a status report.

You resolve a Work Item by changing the **Status** field.

- 1. After locating the Work Item via navigation or search, select it in the Table (or other) view to display its details in the Detail pane.
- 2. Locate the **Status** field, and click the drop-down button to display the list of next workflow actions.
- 3. Select the workflow action that is required by your team's process. Workflow is customizable by a Polarion administrator, so the list of actions may differ from the default. The default actions are Resolve and Resolve and Close. Selecting a workflow action switches the Detail pane into edit mode.
- 4. Before updating the Work Item you may wish to add a comment, attach a file, or create a link to a repository revision. After making any desired edits, finish the resolution process by clicking Save.

## Link a Resolved Item to a Repository Revision

When you resolve a Work Item (for example, a Task, Change Request, or Defect), it can be highly useful for your development organization if you link the resolved Work Item to the repository revision that contains the code that implemented the feature or fix specified in the Work Item. Doing this consistently provides the organization the possibility to trace development from a requirement. through process and changes, to the source code. Aggregations of this kind of data can reveal flaws in the process which can subsequently be corrected to improve the overall quality of development.

The best time to link a Work Item to a revision is when your resolve the Work Item. However, you can create the link any time. If more than one revision contains code that resolves the Work Item, you can link to multiple revisions.

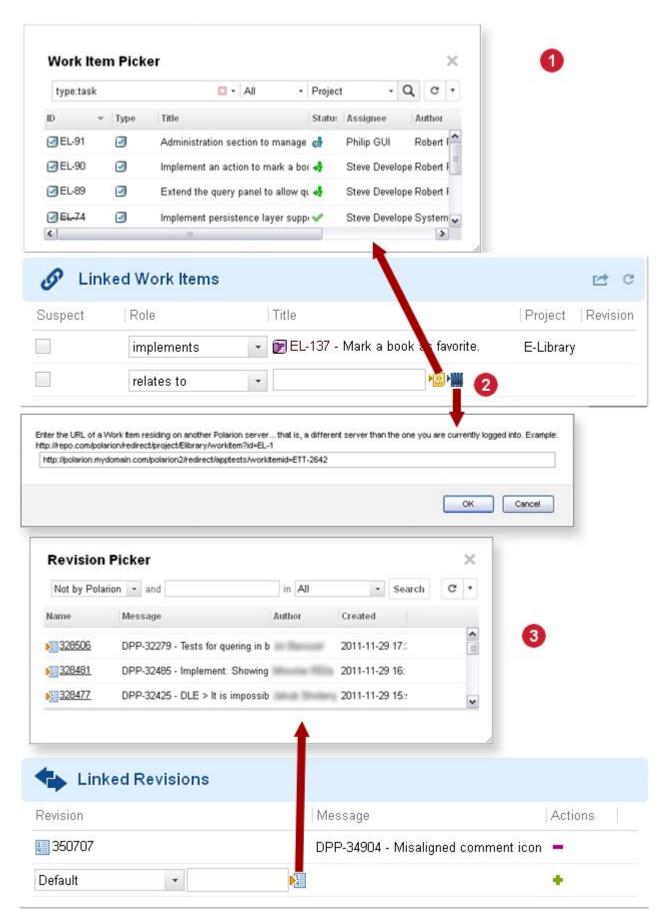
#### Create a link to a Revision

You can create a link to a source code revision in the Linked Revisions section in the Work Item's detail in the Table view. You can quickly navigate to the Linked Revisions section using the Links button in the detail pane toolbar. Click the Edit link at the lower left of the section to access the linking fields and actions.

If you know the revision number, you can enter it directly in the **Revision** field. If you don't know it, you can click the **(Select Revision)** icon next to the **Revision** field and select one or more revisions in the Revision Picker dialog. This dialog box has its own search tool that enables you to locate the desired revision. The Revision Picker contains a preview area. When you click on any of the revisions shown, the preview area displays the changed items in the selected revision.

If your project has been configured to support revision linking from more than one repository, the Linked Revisions section contains a drop-down list from which you can select the repository before selecting the revision. For example, a project might have source code in both Polarion's default Subversion repository, and an external GIT repository. (For more information on this capability, see Using External Repositories.)

With the Work Item detail in Edit mode, you can add or create multiple revision links, or remove an existing link.

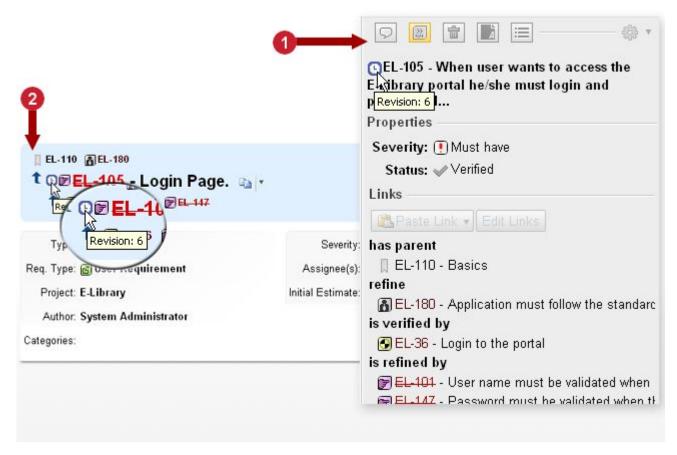


Link to a Work Item Link to item on a different server Link to a Revision

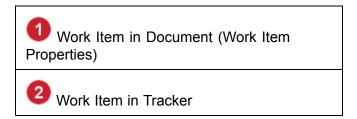
#### Links pinned to Revisions:

If you link a Work Item to some revision of another Work Item other than the Head revision, the link is considered "pinned". If the link is followed and the revision of the Work Item viewed, then some aspects of the Work Item are different than if the link were to the Head revision. Specifically:

- The Work Item cannot be edited.
- The Actions menu contains only the **Print** option. Only the state of the Work Item in the revision will be reflected in the printed output.
- The toolbar of the Work Item form displays the **Show Current** button. Clicking this button switches to the Head revision of the Work Item.
- The browser status bar displays information about the revision when the pointer hovers over the title (Date-time when the item was created and last revised, and the revision number). This same information appears when the pointer hovers over a link to the revision in any Work Item.
- URLs of linked Work Items point to the same revision as the item being viewed, unless they are linked explicitly to some other revision.
- A clock icon (()) appears to the left of the Work Item title to indicate that you are viewing a revision. This same icon appears in links to the revision in any Work Item.
- Backlinks are not displayed.
- Calculated fields do not show any values.



#### Revision that is other than Head



#### Auto-linking via Subversion commit messages

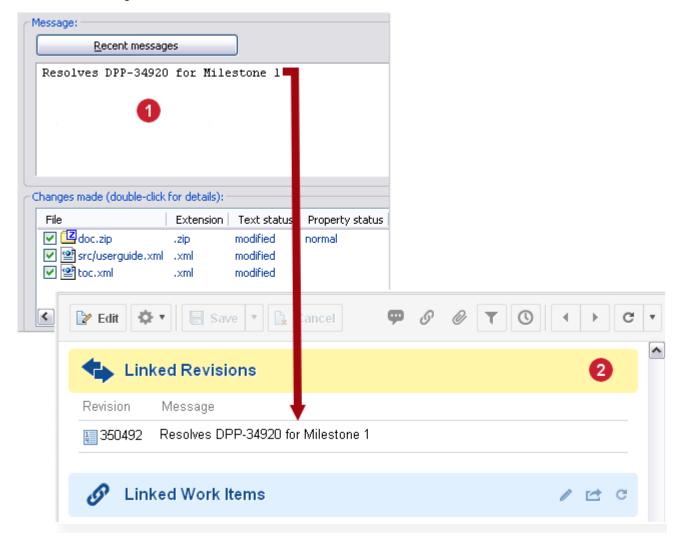
Polarion can recognize Work Item IDs in Subversion commit messages, so you can automatically create a link between a Work Item and a revision simply by including the ID in the commit message. If a commit resolves more than one Work Item, you can include multiple IDs in the commit message and revision links are created for all mentioned items.

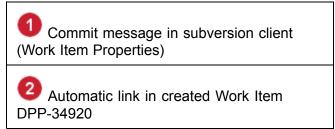
The commit message is scanned for occurrences of strings in form <code>PROJECT\_ID/WORKITEM\_ID</code> or just <code>WORKITEM\_ID</code>, where <code>WORKITEM\_ID</code> has the form <code>PREFIX-number</code> or <code>PREFIX-AZ1234</code>, where <code>AZ1234</code> is the mixture of numbers and <code>uppercase</code> letters.

Example: myProject/PRJ-5678, or just PRJ-5678.

The Work Item identifier must be separated from the rest of the message by a space, or characters: () (parentheses), [] (square brackets), ? (question mark), ! (exclamation mark), , (comma), . (period), " (double straight quote), ' (single straight quote), or : (colon).

Links derived from Subversion commit messages are shown in the Linked Revisions section of the Work Item editor. If a link is incorrect as the result of a typo in the Work Item ID in the commit message, the link can be removed by deleting the relevant row in the Linked Revisions table. If some revision is linked to the Work Item both explicitly by adding it on the Work Item form, and also in a commit message, then it is shown twice in the **Linked Revisions** section on Work Item form.





Revision linking via SVN commit messages

## Tip

If you accidentally specify the wrong Work Item ID in a commit message, resulting in an erroneous revision link, you can manually remove the link. Navigate to the Work Item whose ID you specified by mistake in the commit message, and in the Linked Revisions section of the detail form, use the — icon to remove the linked revision. You should then navigate to the correct Work Item and link the revision manually, as previously described.

The erroneous Work Item ID is not removed from the Subversion commit message, so it's a good idea to leave a comment in the Work Item erroneously referenced to explain what happened and what measures you took to correct it.

## Builds, tests, source code and documentation

### Overview of Builds, tests, source code and documentation

This section covers several more features commonly accessed or used by developers:

- Builds.
- Tests.
- Source code and documentation.

## **Working with Builds**

# ALM REQ QA PRO

## **Note**

## Availability in Polarion Requirements & Reviewer:

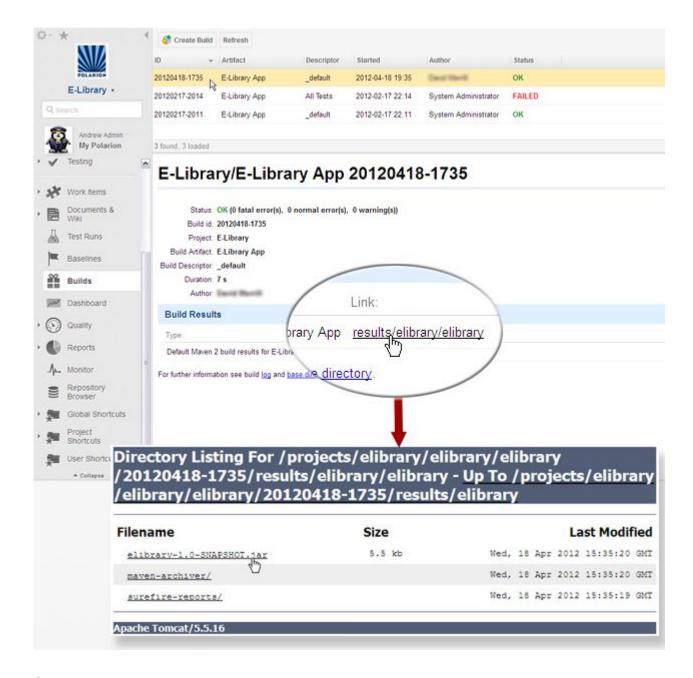
This feature is also available in Polarion Requirements, but it is hidden in the default configuration. The topic can be shown in Navigation by changing the Navigation topics configuration. For more information, see Configure Interface Views.

This feature is read-only in Polarion REVIEWER.

## Scope(s): ■Project

In order to work with builds, you must have the necessary permissions set up by a Polarion administrator. Permission levels are read, create (which allows you to start new builds), and download. You can access project builds via the **Builds** topic. Click this topic to show builds listed in a table (at the top of the page). You can sort the table on any column by clicking the column header. Selecting a build in the table displays its detail in the Detail pane.

Build details include build status, build author (the person who started the build or the name of the system user (default "polarion") in case of initiation by a scheduled Job, and links to the build log and base directory.



#### Start a new Build

If you have create permission for builds, you can start a new build by clicking the **Create Build** in the Builds table. If you are not sure if you have the necessary permissions, mouse over the button and a tool tip tells you if you cannot start a build. When you click the button, a new job is created (which you can see in the Monitor topic), and a new row appears in the Builds table (Builds topic). You can check the build status using **Refresh**. Your user name appears in the **Author** column (and in any build notifications which may be sent as a result of the build).

## **Configure Project Builds**

It is necessary to set up a build configuration for each project. This is done in the \*Administration interface, in the \*Build topic and its subtopics. You need to have administration rights for the project for which you want to configure building.

For detailed information on build configuration, see Basic Build Management.

#### **Back up Project Builds**

See Project Builds.

## **Working with Tests**

Polarion provides summary information about testing, which you can access this information via the Reports topic.

#### View source code and documentation

You can see the source code for any class (for the application or tests) in the current Project and access its Javadoc™ documentation without leaving the web interface. Source code is read-only.

- 1. In the **Navigation** pane, expand the **Reports** node.
- 2. Click the Xref node.
- 3. Locate the class you want to see in the **All Classes** section, and click the class name to open the source in the right-hand frame.

Access Javadoc documentation:

- 1. In the **Navigation** pane, expand the **Reports** node.
- 2. Click the Javadoc node.
- 3. Click any listed Build Artifact. If none are listed, no report is generated. To generate the report, click the Calculate Report button.
- 4. Navigate to the class you want to see, clicking the class name to open its documentation in the main frame.

#### Note

The foregoing assumes that Javadoc has been generated for the classes in your Project. If no Javadoc exists, you see No Data Available main frame of your browser after clicking the Build artifact name.

## Polarion for testing and quality assurance

## Testing and quality assurance overview

Polarion features that support quality assurance and testing activities include:

- Project templates: Preconfigured for test management projects, templates include Work Item types to support testing. They provide for easy linking of test case and requirements artifacts for traceability and impact analysis. As with all project templates, templates with test management support can be customized and/or reused with modifications and customizations, after which users can create projects that have the desired custom attributes as defaults.
- **Test Specification**: Enables import of existing testing assets and artifacts from Microsoft Excel or Word as well as creation of new testing artifacts. Choose a Document based approach, or a tool based approach, or a combination of both.
- Test Execution and Automation: Create Test Runs comprised of references to sets of test cases (manual or automated). Execute Test Runs manually or automatically, integrating JUnit tests with the Polarion Build system. Create Test Runs based on the default Test Run Templates, or derive custom Test Run templates from these and create new Test Runs based on your derivative templates.
- Parametrized Testing: Define Test Parameters in Test Cases, Test Runs, and Test Run Templates that enable an abstract approach to test specifications and efficient running of multiple testing scenarios when executing Test Cases.
- Integrate with External Testing: Execute testing either manually or automatically. Import results of external testing. You can import manually as needed or automatically import results of external automated tests exported to the xUnit framework's XML format for test results. (xUnit export is supported by many third-party automated testing tools). Automatically create Work Items of the Defect type for developers when tests fail.
- Workflow and Electronic Signatures: Configure the system and/or projects with workflow control for Test Runs. Transition actions can be configured to require U.S. 21 CFR Part 11 compliant electronic signatures.
- Excel Round-trip: (Optional) Use the Round-trip for Microsoft Excel feature to export Test Cases for external manual testing and re-import test results stored in the Excel file, automatically creating Defect type of Work Items for developers for failed tests.
- Reporting: Configure standard dashboards and use dedicated macros in custom wiki pages to create dedicated testing/QA report pages for testing status and results.

## Test managment Projects and templates

## **Test Management and Project Templates**

Polarion products with quality assurance (QA) and testing support provide one or more project templates that provide support for creating, managing, and executing test cases. The project templates are preconfigured to provide Work Item types, link roles, and workflows that support QA and testing. Some project templates may also provide support for defining and managing requirements so that these requirements may be easily linked to the test cases that verify them.

This section introduces the standard project templates that include QA and testing support. For a complete listing of project templates with more detailed description, see Default Project Templates.

## **Project Templates**

Template Name	Product / License	Description
Agile Software Project  (Product and Release backlogs, Sprint Management, Quality Assurance, Builds)	ALM	An Agile based Project template that supports the full software application lifecycle from start to finish. It includes the management of product and release backlogs, sprint management and reporting, and test management with full quality assurance coverage.
		It has a standard Java project structure that supports the automated build process managed by Maven-2.
		Project with demo data: E - Library
V-Model Project  (Concept, Requirements, Design, Risks, Planning, Development, Testing, Maintenance, Builds)	ALM	A V-Model (Waterfall) based Project template that supports the full application lifecycle. It provides all that's needed to determine and define requirements, design then assess the risks, plan the development, monitor the development process, test the results and maintain your product.
		It has a standard Java project structure that supports the automated build process managed by Maven-2.
		Project with demo data: Drive Pilot
V-Model Project QA  (Concept, Requirements, Design, Risks, Development, Testing, Maintenance, Builds)	ALM, QA	A V-Model (Waterfall) based Project template that includes Requirements Specifications and their formal approval along with Risk and Test management. You can start by filling in the Requirement Specification documents, and then target requirements for release and analyze potential risks by using the FMEA Worksheet templates. Then it's time to manage test execution. (Both manual and automated are supported.)
		It has a standard Java project structure that supports the automated build process managed by Maven-2.

Template Name	Product / License	Description
		Project with demo data: Drive Pilot QA
V-Model Project REQ (Concept, Requirements, Design, Risks, Maintenance)	ALM, REQ	A V-Model (Waterfall) based Project template that supports Requirement and Test Specification documents. Start by filling in Requirement Specification documents. Then refine them in Test Case Specification documents, and specify the validation and verification procedures using the Validation and Verification Plans. It supports a formal specification approval process.  Project with demo data: Drive Pilot REQ
Specification Project with Variant Management (Requirements, Testing, Variants, Features)	ALM, QA, REQ	A V-Model (Waterfall) based Software Engineering Project configured for Feature, Requirements, and Test Specifications that includes Variant Management with a Propagation option. (If the Propagation option is enabled, the selection of Variant Work Items is determined by the relations to their parents and the restrictions of these parents.)  Project with demo data: Weather Station

#### **Explore template capabilities**

The best way to become familiar with these project templates is to work with a Polarion administrator to set up a sandbox folder in the repository in which you can create experimental projects from the different templates with quality assurance (QA) and test management support. (For a list of the default project templates distributed with Polarion products, see Default Project Templates.) Your administrator may give you permissions to create projects in this folder, or you may have to work with the administrator to actually create your experimental projects in the sandbox depending on your administration policy.

The procedure for creating projects is described in Project Administration. In your sandbox area, try creating a project with each of the foregoing project templates, selecting them on the relevant new project wizard screen. You need to be using a license that provides the template(s) you wish to try. You can always download an evaluation copy of Polarion to try the various project templates, as the evaluation license provides all of them. Remember too that any of the stock project templates can be customized or have copies created which can be customized to suit your exact needs.

#### See also:

- Creating a New Project From a Template.
- Customize Project Templates.
- Configure Work Items.

Polarion for Project Managers.

## Introducing the QA Project template

A project based on the **QA Project** project template has a space named **Testing** (under **Documents and Pages** in Navigation) which contains the following:

- Document: Test Specification. This is a placeholder Document that you can optionally use to
  write up test case specifications. You can mark portions of the content as Work Items of Test
  Case type which are then tracked in the tracker and managed with the project's workflow.
  - For information on working with Documents, see Documents and Pages.
- Page: **Test Case Traceability**. This page is a comprehensive report that shows which Requirements have and do not have linked Test Cases.

#### **Process Overview**

There are three main activities involved in a "QA Project" project:

- 1. Define Test Specifications and Test Cases.
- 2. Link Test Cases for traceability.
- 3. Create one or more Test Runs from a Test Run Template.
- 4. Execute Test Runs.

## Create and manage test specifications

## **Overview - Create and manage test specifications**

In Polarion, specifications for individual tests are broken down into Work Items of the Test Case type. Test cases may be aggregated in a Document or created individually in the integrated tracker. With either approach, the test cases are managed with the project's workflow through each step of your process. You can import existing test cases you have written in Microsoft Office documents, or you can author them using Polarion tools. Once you have Test-Case-type Work Items created, you can link them for traceability. You can link Test Cases to other Test Cases, to Work Items of the Requirement type verified by the Test Cases, and to Work Items of the Defect type that you may create during testing, or any other way to create exactly the traceability you need.

## Import test cases from Microsoft Office

# ALM REQ QA

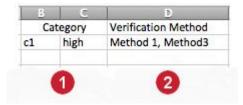
If you have test cases written in Microsoft Word or Microsoft Excel, you can import existing documents to Polarion, creating Test-Case type artifacts in projects which can then be tracked and managed with project workflows. If you import from Microsoft Word, you create a new LiveDoc document, creating Test-Case-type Work Items according to rules you set up to recognize such artifacts from the original document content. If you import from Microsoft Excel, you can import in two ways. If you prefer a document-based approach, import into a LiveDoc Document. If you prefer a tool-based approach, import into the integrated tracker.

For more detailed information on importing information to create Work Items, see Import Work Items.

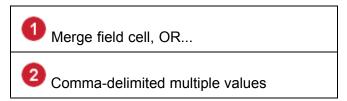
#### Import multivalued data

You may want to specify data in an Excel worksheet to be imported into multivalued fields in Polarion. For example, the Category field can contain multiple values. Your project might also have some multivalued custom fields. If properly formatted in the Excel sheet, the Polarion importer recognizes multiple values and imports them correctly. You can format multiple values in two ways:

- Specify each value in a cell under the column for the field, and merge that column to span all the values.
- Specify all values in a single cell under the column for the field, separating each value with a comma character.



You have two alternatives for formatting Excel data for import to Polarion multivalued fields



Recognition of multivalued data can be switched on or off in the import wizard using the Separate multiple values option. When this option is selected, delimited values in a mapped column are imported to Polarion as multiple values in the target field. Recognized delimiters are comma (.) and semi-colon (;) characters. Detection of merged heading cells in Excel is automatic.

For reference information on supported fields, see Multivalued Data and Fields.

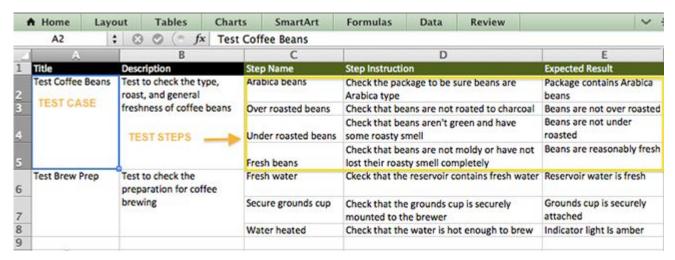
#### **Define Test Steps in Excel**

## ALM QA

When defining Test Cases having discreet test steps, you can format your Excel worksheet in such a way as to facilitate import into Polarion. Here are a few tips:

- Specify a column with a name like **Title** to identify each Test Case. You can include another column such as **Description** for the general description of each Test Case.
- Define columns corresponding to those defined in your projects Test Step Columns configuration, for example, Step Name, Step Instruction, Expected Result.
- Use one row for each step when writing up the test steps. Then, when these are complete, vertically merge the Title and Description columns so that they encompass the test step rows (see figure below).

You may wish to leave an empty row between test cases. This can give you an additional import rule condition when importing your test cases.



When you import the Excel workbook, you can map test steps to the Test Steps field of Test Cases. For more information on importing from Excel and mapping columns to fields, see Import from Excel.

The following sections discuss creating new Work Items in existing Documents and the Work Item Tracker.

#### **Define test cases in Documents**

Polarion's LiveDoc Documents (Documents) are often preferred for defining test cases by teams who are accustomed to work with document-based test case specifications. Content in Documents can be marked as Work Items of the Test Case type. They can then be tracked and managed with project workflow, yet still appear as content in a document.

When you create a Document for specifying test cases, be sure to select Test Case as the Work Item type the new Document should contain. If you accidentally select a different type or you want to have multiple Work Item types in the Document, you can configure the Document after it is created.

To learn all about how to work with Documents, review Documents and Pages Topic.

#### aiT

Projects can be configured by an administrator to include a table of manual test steps in Work Items of the Test Case type. When Test Steps support is enabled, discrete test steps can be defined in this special Test Steps field. The results of individual test steps can then be logged when testers execute manual tests.

When configured this way, the Document's Work Item presentation displays the Test Steps table in Work Items. Rows can be added to the table using the **Add Row** button, which appears when the cursor is located inside the test steps table when the Description field is in Edit mode.

If you do not see the Test Steps in Test Case items in a Document, it is possible that another user changed the Work Item presentation to some setting that does not display test steps. To show the Test Steps table in Document-based Work Items, you can set the Document's Work Item Presentation. Click the drop-down control beside the Work Item

icon in the Document Editor toolbar, and on the menu choose Configure. In the dialog box, select one of the options that includes Test Steps.

For more information, see Executing a Test Run and Enabling Manual Test Steps.

#### **Export test Documents to Microsoft Word**

You can use the Round-trip for Microsoft Word feature to export test specification Documents to Word for review, comment, and collaboration. For more information, see Share Documents Using Word Round-trip.

You should note the following when exporting test specification Documents containing Test Steps in Test Cases:

- A Document that is configured to show Title, Description, and Test Steps, or Title and Test Steps can be exported for Review, Prioritization, or Collaboration.
- When exported for Review or Prioritization, the Test Steps can be commented. Comments are written to the Description field upon re-import to Polarion.
- The content of test steps cannot be edited in the exported file.

#### **Define test cases in the Tracker**

If your team prefers a tool-based approach to defining test cases, you can create Work Items of the Test Case type directly in the integrated Work Item tracker. If your project is based on one of the standard Polarion project templates with QA and testing support, the Test Case type is preconfigured. If your project's configuration is customized, the Work Item type corresponding to a test case may have a different name, which you should substitute for "Test Case" in the procedure that follows.

- 1. Log on and open the relevant project. Your user permissions for the project must allow you to create new Work Items.
- 2. In Navigation, select Work Items topic.
- On the toolbar of the Work Items page, click **Create** and choose **Test Case** on the menu. A new, empty Work Item form appears in the viewer/editor pane of the page.
  - (You cannot create new Work Items in the Matrix and Time Sheet views of the Work Items topic.)
- 4. Fill in the fields, supplying a value for all required fields. Optionally create links to other Work Items, specify hyperlinks to relevant resources, and add attachments.
- 5. Click Create on the viewer/editor toolbar to create the new Test Case item.

### Tip

Projects can be configured by an administrator to include a table of manual test steps in the Description field of Test Case items. When Test Steps support is enabled, the results of the individual test steps can be logged for manual tests. If there are multiple tables in the Description, the table of test steps must be the first table in the field. Rows can be added to the table using the **Add Row** button, which appears when the cursor is located inside the test steps table when the Description field is in Edit mode. For more information, see Execute a Test Run and Enable Manual Test Steps.

#### Link test cases

In the default configuration of projects based on a Polarion project template that provides QA and testing support, test case Work Items can be linked to each other and to other Work Item types such as defects or requirements. Linking is vital for future traceability and impact analysis. You can link items to others in the same project or in different project. In systems configured for multiple concurrent Polarion servers, it is possible to link to items residing on a different server, but the linking is not as robust as when items are linked within or across projects residing on the same server.

Typical links in QA and testing projects include:

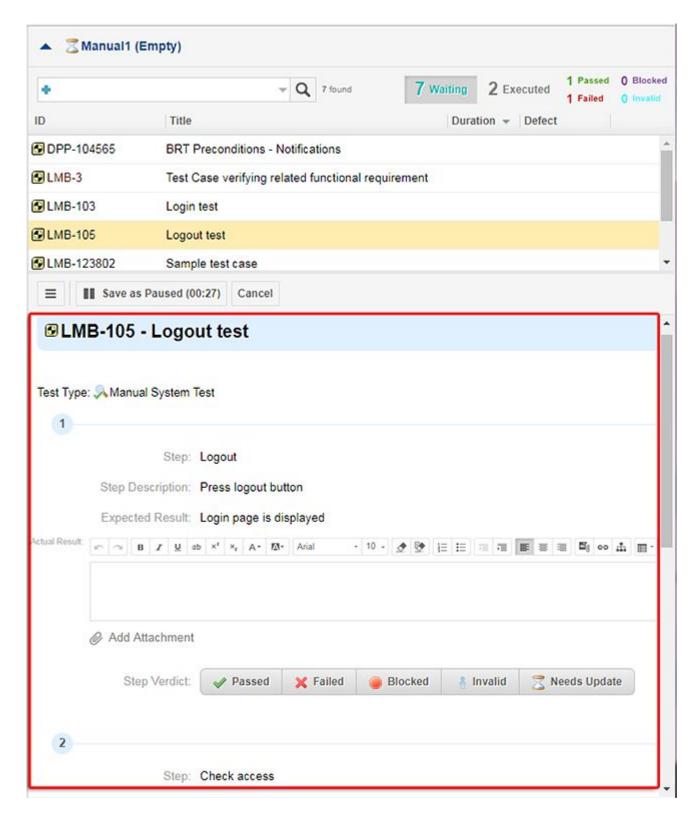
- Linking Test Case items to Requirement items with a "verifies/is-verified-by" link role (relationship).
- Linking Defect items to Test Case items with an "is-triggered-by/triggers" link role (relationship).
   When using test execution features of Polarion's QA and testing project templates, Defect items with such link roles are created automatically upon failure of tests.

Linking of Work Items is discussed in more detail in Link Work Items.

### **Specify manual Test Steps**

You can optionally define discrete *Test Steps* in Test Cases.

The advantage of this approach is threefold:



- Any failure can be linked and easily traced to the test step in which it occurred.
- Test Steps data is separated from the Description field, which can be then reserved for general information about the Test Case.

 If a Test Run is exported to Excel for manual execution of tests offline, only test step and results data are involved in the round-trip. Other Test Case data, such as the Description, cannot be modified by external testers.

## Specify manual Test Steps in the Classic Execution view

You can switch back to the classic execution view method described below by entering the following property in the properties section.

testManagement.useOldExecutionView=true

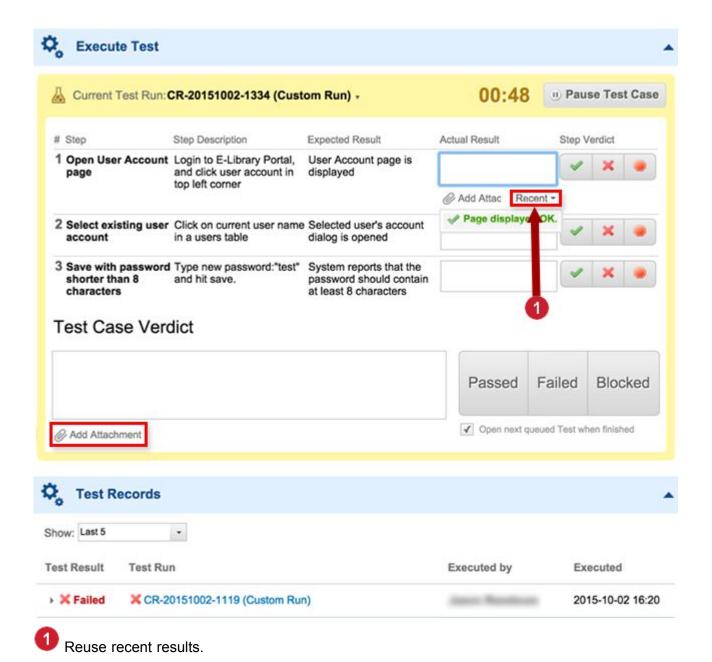
## Warning

To give current testers some time to transition to the new Test Execution View workflow, we will still support the Classic Test Execution View for a few more releases. **Recommendation**: Get to know the improved workflow outlined in the Executing a Test Run Using Polarion section.

Your project administrator can then enable the specification of manual test steps in the project configuration.

## (See Enabling Manual Test Steps.)

When correctly configured, a table of test steps appears in the Custom Fields section of new Test Case items (unless the administrator opted to display the table separately). Testers can specify each test step as a row in the Test Steps table. The table is rendered as executable test steps in a panel in the Execute Test Case section of the Test Case Work Item when the Test Case is executed during a Test Run. As shown below, Testers can log the results of individual test steps.



#### **Edit Test Steps in Documents**

When Test Steps are enabled in your project, they are a field of your Test Case type of Work Items, so the steps can be edited in Test Cases defined in Documents. You need to set the Presentation Configuration in the Document, selecting either **Title and Test Steps** or **Title**, **Description and Test Steps** in the **Content** column of the Configure Work Item Presentation dialog box. See Multiple Work Item Types for more information on this dialog box.

## **Create and manage Test Runs**

#### **About Test Runs**

**Test Runs** log instances of the running or performance of a defined set of **Test Cases**. **Test Runs** may log results of manual tests or automated tests run as part of a build. **Test Runs** also contain information about the status of the **Test Run**, results of the instance of testing that the **Test Run** represents, the test environment, and the build tested.

An individual **Test Run** is based on a **Test Run Template**. Polarion products that support test management come with predefined templates which can be customized or used as the basis for custom **Test Run Templates**.

## **Customize Test Run Templates**

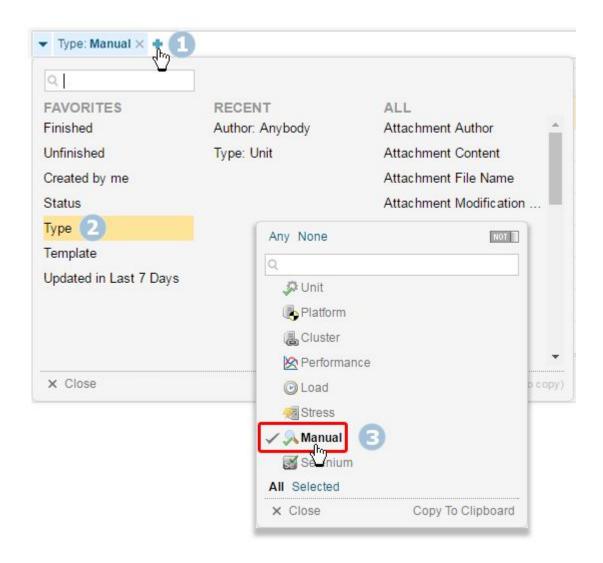
# ALM QA

Test Run Templates can save a great deal of time over time. You configure the properties of the template once, and then all Test Runs based on the template are preconfigured with the same properties (some of which Test Run creators can modify in Test Runs based on the template.)

Access Test Run templates

- 1. Open the project containing the Test Cases that will be tested with Test Runs.
- 2. In Navigation, click **Test Runs**. The Test Runs page opens. The top section of the page displays a table of existing Test Runs.
- 3. On the toolbar of the Test Runs page, click **Manage Templates**. The table of Test Runs is replaced by a table of Test Run Templates.

You can filter Test Run Templates using query bubbles.



A Test Run Template is essentially a Test Run, with configurable properties and page design. The main difference is that a template itself cannot be executed. Only Test Runs created from it can be executed. So your goal in customizing a Test Run Template is to set it up exactly as if it were a Test Run. You can customize any existing Test Run Template by modifying the Test Run Properties, or by modifying the underlying Widgets and so forth that comprise the Test Run page, or both.

Any existing Test Run Template or Test Run can serve as the basis for a Test Run Template. To create one from an existing template, simply create a copy of the existing template and customize that. If you want to customize one of Polarion's default Test Run Templates, always create a copy for customization so you don't lose the default template. To create a Test Run Template from an existing Test Run, simply open it, click (Actions), and select **Save As Template**.

Create a copy of a Test Run Template to customize:

 Access Test Run Templates as described above, and select the template you want to duplicate and customize in the table of Test Run Templates.

- 2. On the toolbar of the detail of the selected template (bottom part of the page), click ⚠ (Actions) > **Duplicate Template** or **Save as Template**. The text of the menu item depends on the Polarion version you are using, and/or the type of the template's page: LiveReport or Classic Wiki.
- 3. In the dialog box, provide a unique ID for the new Test Run Template and a human-friendly title, and then click **Duplicate Test Run Template**.

After creating the duplicate Test Run Template, you can proceed to modify it as described in the next sections. If workflow is configured for the Test Run type, the Duplicate Template action resets the Status to the configured initial status, and invokes the configured initial workflow action, if any.

## **Modify Test Run Template Properties**

Test Run Properties include such data as Test Run type, testing platform and environment, and Test Parameters. Every Test Run created using the modified Test Run Template will have the properties and values as specified in the template. A simple graphical user interface is provided for editing the template properties.

- 1. Access a Test Run Template as previously described.
- 2. With the desired Test Run Template selected in the table of templates, click **Properties** on the toolbar of the template detail pane (lower part of the page).

You can edit the title and the property fields in place, or click **Edit** to place all non-read only fields into edit mode.

The values for some properties like test Type are defined in the global or project Testing configuration in Administration. You can select from configured values, but you cannot change the values themselves. For more information, see Administrator's Guide+ Configure Testing.

In a Test Run template, the Status field is not editable if a workflow configuration for Test Runs exists.

## **Specifying Test Case selection**

In the Test Run Template properties, it is important to specify the method of populating new Test Runs (created by users and based on the template) with Test Cases to be executed. You do this by setting a value for the **Select Test Cases** field. The field presents the following list of options:

- Manually: Creators of new Test Runs based on the template can manually select the Test Cases to be executed, or the Test Cases may be manually selected by the author of the Test Run template, in which case they are preselected when a new Test Run is created from the template. When this option is selected a multivalued field named **Project Span** appears. If you only need to select Test Cases from the current project, you can leave that field empty. If you want to be able to select Test Cases that are defined in one or more different projects, then specify those projects in **Project Span**. If, in addition to the other projects, you also want to select Test Cases from the current project, then you must include it as one of the values specified in the **Project Span** field.
- By Query on Create: Test Cases are selected by a query which automatically runs when a new Test Run based on the template is *created*. The set of waiting Test Cases of the Test Run is static and does not change if new Test Cases that meet the query criteria are added after a Test Run is created, but before it is executed. Nor will any be added to the set if the Test Run is executed multiple times, even if new Test Cases that would meet the criteria have subsequently been added in the system since the last execution of the Test Run.

When you select this option you must specify Lucene query syntax items in the **Query** field that will select the Test Cases that testers must execute in Test Runs based on the Template. You must specify this guery before you can save changes to the Template.

By Query on Execute: Test Cases for Test Runs are selected by a query which automatically runs when a Test Run based on the template is executed. The set of Test Cases waiting to be executed is not static. Waiting Test Cases are taken from the current set of Test Cases in the system meeting the guery criteria at the time a user executes the Test Run, and could even change during the execution each time the view refreshes.

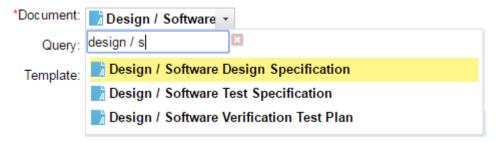
When you select this option you must specify Lucene query syntax items in the Query field that will select the Test Cases that testers must execute in Test Runs based on the Template. You must specify this query before you can save changes to the Template.

When this option is selected the multivalued field **Project Span** appears. If you only need to select Test Cases from the current project, you can leave that field empty. If you want to be able to select Test Cases that are defined in one or more different projects, then specify those projects in **Project Span**. If, in addition to the other projects, you also want to select Test Cases from the current project, then you must include it as one of the values specified in the **Project Span** field. The query entered in the **Query** field should be valid for all specified projects, otherwise some desired Test Cases may not be added to the Test Run, or some Test Cases may be added that are not relevant. For example, if you specify a query like (type:testcase AND status:active AND testType: (manual user)), but in one of the projects the test type is configured as manual ui and not manual user, Test Cases from that project are not be added to your Test Run.

From LiveDoc on Create: All Test Cases contained in the specified Document are selected for execution when a new Test Run based on the template is created. The set of waiting Test Cases of the Test Run is static and does not change if new Test Cases are added to the Document after a Test Run is created, but before it is executed. Nor will any be added to the set if the Test Run is executed multiple times.

Specify the Document in the required **Document** field of the Test Run Template properties. Select a document from the drop-down menu. Preview it by clicking on the icon beside it, or just start typing a word contained within the document's name. Documents located outside the default Space will include the Space name before it. (Separated by a "/"). You can optionally specify a query in the Query field to select a subset of the Test Cases contained in the specified Document. For example, you might query for Test Cases with a Status value that is not "Draft", such as NOT status:draft.

Figure 20.5. Dynamic Document Selector



If the list of documents is extensive and covers several **Spaces**, just type the Space name. Only documents within that space are listed.

• From LiveDoc on Execute: Test Cases appearing in the specified Document, including Test Cases referenced from another Document, are selected for execution when a Test Run based on the template is executed. The set of Test Cases waiting to be executed is not static. Waiting Test Cases are taken from the current set of Test Cases appearing in the Document at the time a user executes the Test Run, and could even change during the execution each time the view refreshes. Referenced Test Case items can be from projects other than the Test Run and the specified Document, or frozen to a particular revision of the referenced Test Case, or both.

Specify the Document in the required **Document** field of the Test Run Template properties. (See From LiveDoc on Create for the drop-down box's behavior.) You can optionally specify a query in the **Query** field to select a subset of the Test Cases contained in the specified Document. For example, for a Smoke test, you might query for open manual type Test Cases, like status:open AND type: manual.

Automation: Test Runs based on the template are created by an automated process and results
of test execution are imported to Polarion.

#### Caution

For 'By Query on Execute' or 'From LiveDoc on Execute' selection types. (Dynamic Test Runs):

- When closing dynamic Test Runs by setting a previously unset finished date, usually by a workflow action, the number of waiting Test Cases gets stored in the Test Run as empty test records and no longer changes.
- When reopening a dynamic Test Run, by clearing a previously set "Finished On" date

   usually also by a workflow action, the number of waiting Test Cases becomes live
   again and corresponds to the current number of unexecuted Test Cases matching
   the Test Run query.
- The query to retrieve or count awaiting Test Cases will therefore differ for finished and unfinished dynamic Test Runs.

#### Modify a Test Run Template's page

Test Runs and Templates are implemented as LiveReport type Pages. You can use the full range of editing features and Widgets to customize the layout and content of a Test Run Template.

- 1. In Navigation, select **Test Runs**.
- 2. On the page toolbar, click **Manage Templates**. Then, in the table of Test Run Templates, select the one you want to modify.
- 3. On the toolbar of the detail pane, click (Actions) Customize Test Run Page.

An instance of the Page Designer opens in the detail panel of the Page. You can then modify the Page according to your needs. For more information, see: Working With Pages.

#### Create a Test Run

## ALM QA

- 1. In Navigation, click **Test Runs**. The Test Runs page loads displaying a table of existing Test Runs.
- 2. In the toolbar of the Test Runs table, click **Create** to launch the **Create Test Run** dialog box.
- 3. In the **ID** field, enter a unique ID. The value you enter must not be used by another Test Run in the project.
  - The ID field does not appear if your project is configured to generate Test Run IDs automatically.
- 4. Fill in the **Title** field with a human-friendly title for the Test Run. The title appears together with the ID in macro output, report Widgets, object enumerations, combo box lists, and generated defects in the Polarion user interface as well as in email notifications. If you leave Title empty, only the ID appears.
  - Providing a title is recommended to help users differentiate Test Runs in a listing, even if they have autogenerated IDs.
- 5. In the **Template** field, select the desired Test Run template from the list of available templates.
- 6. Click **Create Test Run** to create the new Test Run.

## **Modify Test Run properties**

The values in the properties of a Test Run are inherited from the Test Run template from which it is derived. If you have the necessary permissions, you can modify a Test Run's properties. For example, if the template specifies one or more projects from which Test Cases can be selected in the Project Span field, then for a particular Test Run you can either modify the project selection or clear the field entirely so that Test Cases can only be selected from the current project.

- 1. Select the Test Run in the table at the top of the Test Runs page.
- Click **Properties** in the detail section of the selected Test Run.
- 3. After saving all changes to the properties, click the Back button on the toolbar to close the properties.

### Select planning test cases

## ALM QA

If a newly created Test Run is based on a Test Run Template that is configured for manual selection of Test Cases (see Specify Test Case Selection), you must manually select the Test Cases before you or others execute the Test Run. You can select the Test Cases using the Work Item Tracker or select them in a Document (if they are defined in a Document).

Test Cases to be tested by the Test Run can be defined in the current project and/or in one or more other projects. For example, a project named Standard Tests might define a set of Test Cases that must always be tested for every application, while the current project contains Test Cases specific to an application or variant. When Standard Tests is specified in the Project Span field of the Test

Run properties, you are able to select Test Cases from it. If you also want to select Test Cases form the current project, you must include it in the **Project Span** field.

## Tip

The **Project Span** property must specify *all* projects from which Test Cases are selected, including the current project, if applicable. If all Test Cases are in the current project, you can leave **Project Span** empty.

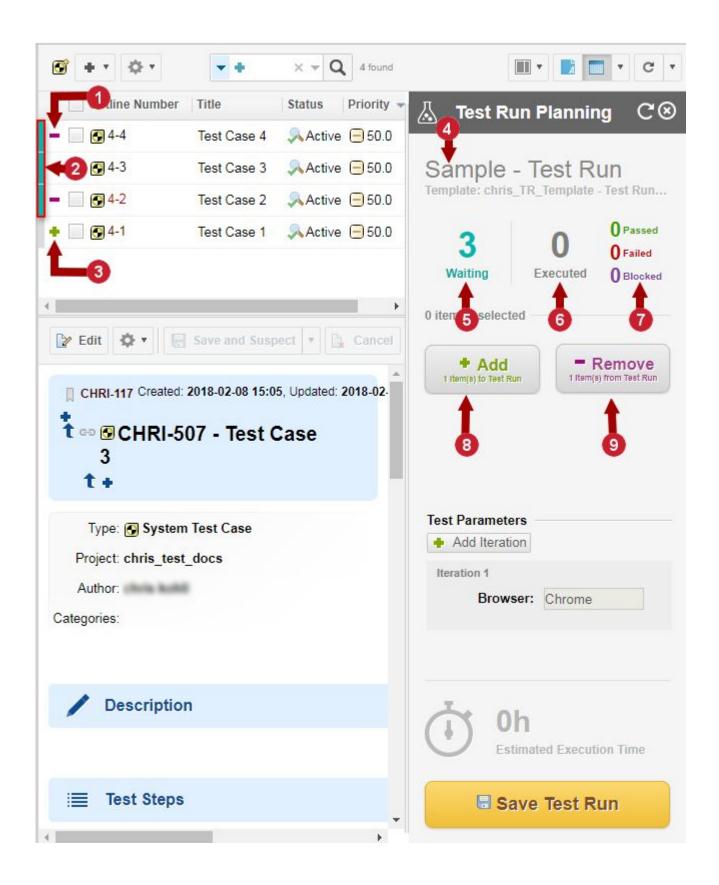
## Tracker approach

The following steps assume there is an existing Test Run configured for manual selection of Test Cases.

### To begin the selection:

- 1. Open your project if not already open.
- 2. In Navigation, select **Test Runs**, and in the upper pane of the Test Runs page, select the Test Run for which you want to select Test Cases.
- 3. In the Test Run detail pane, click (Actions) and choose **Select Test Cases**. This action appears only when manual selection of Test Cases is specified in the Test Run properties.

The Table view of the Work Items topic displays the **Test Run Planning** sidebar. You can add or remove "waiting" Test Cases from the manual selection of Test Cases for a Test Run.



1 Remove from Waiting	6 6 7 Click to filter the table by selected status
2 Selected now Waiting	8 Add selected item(s) to Waiting.
3 Add to Waiting	Remove selected item(s) from Waiting
Navigate to Test Run	

You can browse the Test Cases listed in the table and select those that testers should execute when executing the Test Run. The sidebar updates to show the total number of Test Case executions waiting. If some Test Cases have multiple Iterations defined for the current Test Run, these are reflected in the "Waiting" count, which may then be larger than the number of items selected in the table. Items in the table that are currently added as "Waiting" show a colored highlight at their left border.

The items appearing in the table are the result of a query, which is built and run as a result of the *Select Test Cases* action of the Test Run. The query appears in the visual query builder on the page, and you can optionally modify it so that the table contains all the Test Cases you want for the Test Run.

To add a Test Case to the queue of waiting Test Cases, click the • icon in its row in the table. To remove an added Test Case, click the • icon that appears in its row.

When you have added all Test Cases that should be executed by the Test Run, click **Save Test Run** at the bottom of the **Test Run** panel.

If you want to further configure the Test Run or execute it, select the **ATest Runs** topic in **Navigation**, and then select the Test Run in the list of Test Runs.

#### Tip

- Clicking the Test Run ID shown in the Test Run Planning sidebar navigates directly back to the Test Run page.
- Clicking on the **Waiting** label in the sidebar filters the table to list only waiting Test cases. The **Executed** filters the table to list only executed Test Cases.
- You can select multiple items in the table (use the check box on each row) and add or remove them from the Waiting list all at once using the respective buttons in the Test Run Planning sidebar.
- The Navigation scope can change depending on the setting in the Test Run's Project Span property. If multiple projects are specified in that setting, the scope switches to **Repository**. If just one project is specified, the scope switches to that project. If Project Span is empty, the scope remains in the current project.

To return to the Test Run after selecting Test Cases, regardless of current Navigation scope, click the Test Run name in the **Test Run Planning** sidebar. If not already open, the project containing the Test Run opens and the Test Run page loads in the browser.

## **Document approach**

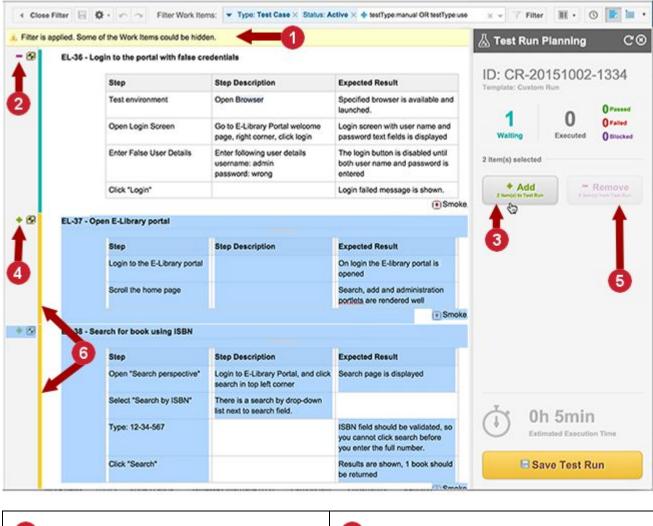
The following steps assume there is an existing Test Run configured for manual selection of Test Cases.

## To begin the selection:

- 1. Open your project if not already open.
- 2. In Navigation, select Test Runs, and in the upper pane of the Test Runs page, select the Test Run for which you want to select Test Cases.
- 3. In the Test Run detail pane, click (Actions), and then choose **Select Test Cases**. The Table view of the Work Items topic opens.
- 4. If it is not already opened, open the **Test Run Planning** sidebar using the **Show Sidebar** menu on the Document Editor toolbar.
- 5. Go to **Navigation**, navigate to the space containing the Document that defines the Test Cases to be selected for the Test Run, and select that Document. The Document opens in the Document Editor, and the **Test Run Planning** sidebar remains open.

If the Navigation node contains too many Documents, you need to locate and open the desired Document via the space Index page or search. After opening the Document, open the Test Run Planning sidebar as previously described.

You are now ready to add Test Cases from the Document to the Test Run. Please refer to the following figure:



Document filtered for relevant Test Cases.	4 + Add to Waiting.
2 - Remove from Waiting.	Remove selected item(s) from Waiting.
3 Add selected items to Waiting.	6 Multiple items selected.

- To add a Test Case, select it by clicking anywhere in its content; the selection is indicated by the gold left border. Then, either click in the left margin or **Add** in the **Test Run Planning** sidebar.
  - The count of waiting Test Cases is incremented in the **Test Run** panel. If the Test Case defines multiple **Iterations**, the "Waiting" count reflects the total number of Test Case executions, which may be higher than the number of Test Cases added to the Test Run's queue.
- To add multiple Test Cases, select all of them (Shift + Click in most browsers), then click the Add button in the Test Run Planning sidebar.

The count of waiting Test Case executions is incremented in the panel by the number of selected items plus additional Iterations, if any. Note that multiple items must be contiguous in order to be selected at once.

- To remove a previously added item from the queue of waiting Test Cases, select the Test Case in the Document body. Click either the - icon in the left margin or Remove in the Test Run Planning sidebar.
- After adding all the Test Cases you want to be tested in the Test Run, click the Save Test Run button in the sidebar.

Testers can now execute the Test Run using the Execute Test button on the Test Run page.

## Tip

When the Document is opened as described above, it is filtered by a guery which is built and run as a result of the Select Test Cases action of the Test Run. You can optionally modify this filter query to increase or reduce the number of items displayed in the Document.

### Plan Work Items from a Document revision

Test managers may sometimes need to populate a Test Run with Test Case items from a specific version of a test specification Document that has been reviewed and approved.

- 1. Open the desired Document revision in the Document Editor. See View Document History.
- 2. With the revision open in the Document Editor, create a new branched Document ( Actions **Menu Pranch**). See Branching Documents for more detailed information.

For example, suppose a Document named *Test Specification* that is under continuous development, and a Plan Version 1.0 that should be populated from Revision 1001, has been reviewed and approved for production. From Revision 1001 of Test Specification, you create a branched Document named, for example, Test Specification Version 1.0 - Frozen.

- 3. Create a new Test Run Template named, for example, "Rev. 1001 Tests", selecting Empty as the template base.
- 4. Set the **Select Test Cases** field to **LiveDoc on create**, and in the **Document** drop-down box specify the branched Document created in Step 2 above. (Just start typing a word in the document name and select it from the options that appear.) Save changes to the template, and click Manage Templates to return to the Test Runs page.
- 5. Create a new Test Run based on the new Test Run Template you just created, edit the properties if necessary, and save it.

When this Test Run is executed, the Test Cases are those from the branched Document that was branched from Revision 1001 of the specification.

**Edit an existing Test Run** 



After creating the new Test Run (or at some other time if something changes before the Test Run is executed), you may want to edit some Test Run properties such as the test environment or Test Run Parameters. After selecting the Test Run you want to edit:

- If you are not already on the Test Runs page, click **Test Runs** in Navigation to access it. Then, in the list of Test Runs, select the Test Run you want to edit. Optionally, use the Visual Query Builder at the top of the page to filter the list of Test Runs. For example, you might query for Manual type tests based on a specific Test Run Template. For more information on the visual query builder tool, see Searching Work Items.
- Selecting a Test Run in the table at the top of the page.
- If you want to edit the Test Run's properties, click **Properties**. If any fields of a Test Run are configured as required fields, a red asterisk appears and the field must be filled before you can save the Test Run properties.
- If you want to modify the Test Run's page, for example the layout, click (Actions) and select Customize Test Run Page.

#### Add attachments to a Test Run

You can attach any type of file to a **Test Run**, either while in the process of creating it, or later on when logging the result. For example, you might attach a file containing test result output from some external testing tool. To add or manage attachments, click (Attachments) on the **Test Run** toolbar to scroll the page to the **Attachments** field.

If your system is configured to support searching on attachment properties and content, you can search on properties of Test Run attachments. In the **Test Runs** topic of Navigation, you can search on attachment title, attachment author ID or name, attachment file name, content, date updated, and size.

In site search in Navigation, you can search attachments to Test Runs via full-text search.

## **Execute a Test Run using Polarion**

# ALM REQ QA PRO

The execution of Test Cases in a Test Run has been revamped in Polarion 18.

See the Classic Test Execution View section to configure Polarion to use the old Test Execution view workflow.

### Warning

## Classic test execution view will become obsolete:

To give current testers some time to transition to the new Test Execution View workflow, we will still support the Classic Test Execution View for a few more releases.

### Tip

Get to know the improved workflow outlined in this section.

Automated tests are run by the Polarion build tool according to the build configuration, so you do not need to explicitly invoke Test Run execution. This section discusses manual execution of Test Runs in Polarion. If manual testing will be done externally to Polarion, see Executing Test Runs Offline Using Excel.

As previously mentioned, a Test Run represents an instance of executing a set of test cases, which have been defined in Work Items of the Test Case type. At some point after a Test Run is created, a tester manually performs the testing steps defined in the Test Cases of the Test Run. During the actual execution, the result of each testing step is recorded, as is the outcome or "verdict" of each Test Case (passed, failed, or blocked).

Even if you use some external tool to execute Test Cases, testers can still execute a Test Run to log the results of external testing so that testing history and traceability are maintained in Polarion across the full life cycle.

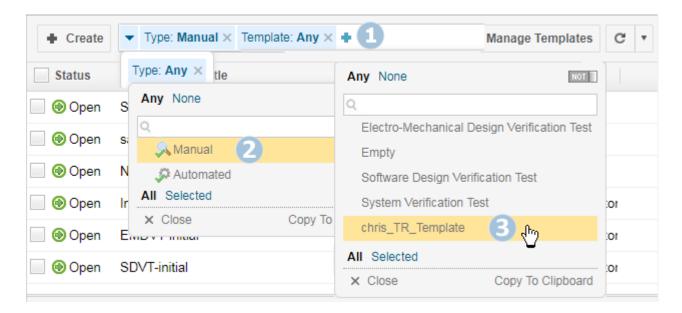
## Tip

It is possible that another user could modify the Test Case or Test Steps while you have paused test execution. If this happens, the **Resume Test Case** dialog box (if Test Case was revised) appears or the **Retest Test Case** dialog box (if Test Steps were revised) appears.

- **Resume Test Case** warns you that you are no longer testing what was planned. For example, the Test Case was planned from HEAD revision, and the Test Case subsequently was updated while the Test Case was in progress. Or, possibly the revision in which the Test Case was planned changed (for example, frozen to a different revision in a Test Run where Test Cases are selected via the LiveDoc on Execute option). In both scenarios, after clicking Resume Test Case in the dialog box, the Test Steps remain the same and you continue working in the revision in which you started.
- Retest Test Case means that the Test Case (including the Test Steps) has been updated and retesting needs to be done in the new version. If you click Retest Test **Case**, the Test Case is reloaded and execution is automatically started.

#### Execute a manual Test Run

1. Open the Test Run's page by clicking ... Test Runs in Navigation, and selecting the Test Run in the table of Test Runs. You can use the visual query builder on the Test Runs page to filter a large number of Test Runs. For example, you might query for only Manual type tests, based on a specific Test Run Template.

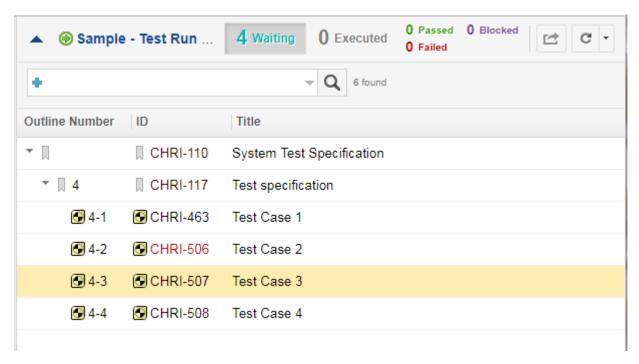


2. The Test Run's details appear below the list.

## Sample - Test Run (chris TR Template - Test Run Template)



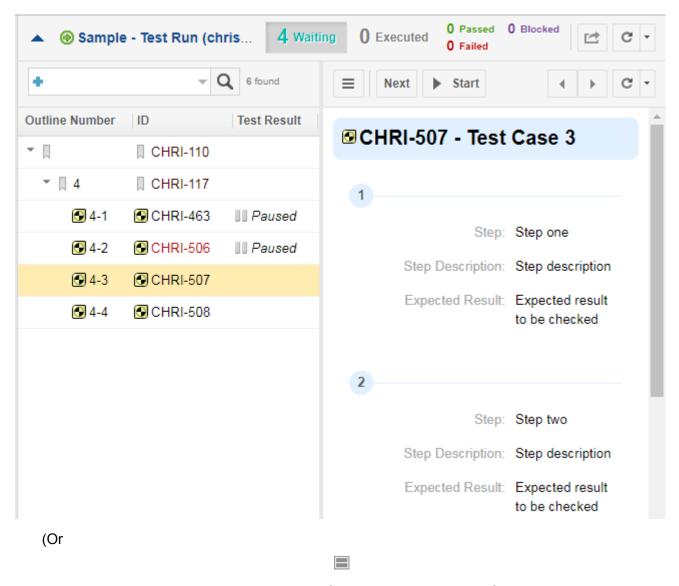
- Click the green Execute Test button.
- 4. A table listing the Test Cases currently associated with the Test Run appears.



The **Outline Number** column appears if the document referred to by the Test Run has **Outline** Numbering turned on.

#### **Buttons:**

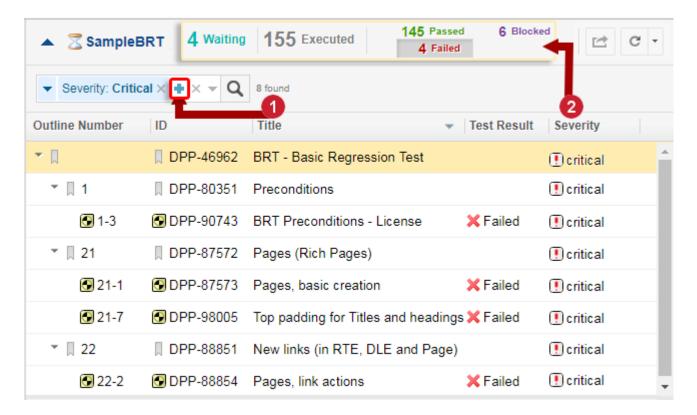
- Refreshes the table and/or Test Case. The top refreshes both.
   When a Test Case is modified (executed/restarted/or paused), the columns and the Test Run statuses ("Waiting", "Executed" etc.) are refreshed automatically.
- **Next** Skips items that can't be executed and jumps directly to the next executable Test Case.
- Cycle through all items, including those that can't be executed, such as Headers.
- Launches a table of your queried results where you can add or remove Test Cases, if the Test Run type allows it.
- Execution Settings.
- 5. (Optional) Click CD, and then Tile Panes Vertically to display the table and selected Test Case side by side.

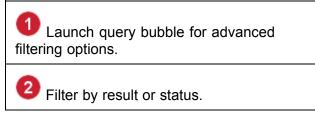


Tile Panes Horizontally to return to the default, table over the Test Case, view.)

6. Use the query builder above the table to search for individual Test Cases or sort by things like Severity.

You cannot set a default query, but you can easily save them for quick reuse.

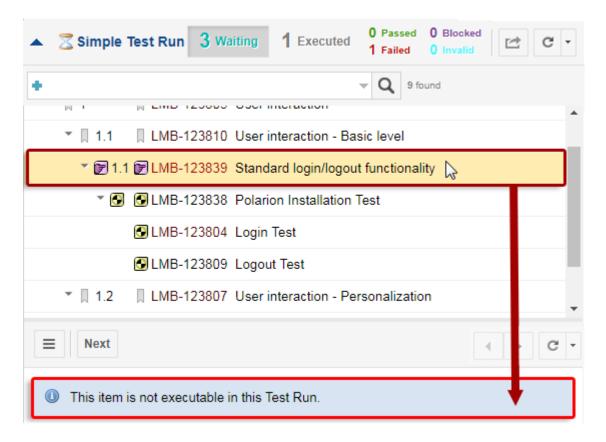




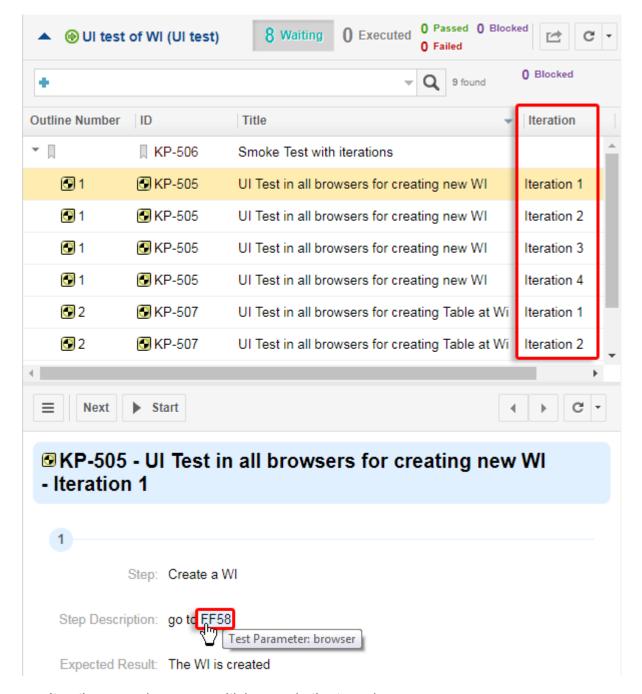
You can also quickly narrow down what Test Cases appear by clicking on a Status or Test Case result.

Click on a selected status or result a second time to remove the filter.

7. If a Test Run item is selected in the table that cannot be executed ( Headings, other Work Items or Test Cases) an informational message appears below the table.



8. The table also supports iterations.

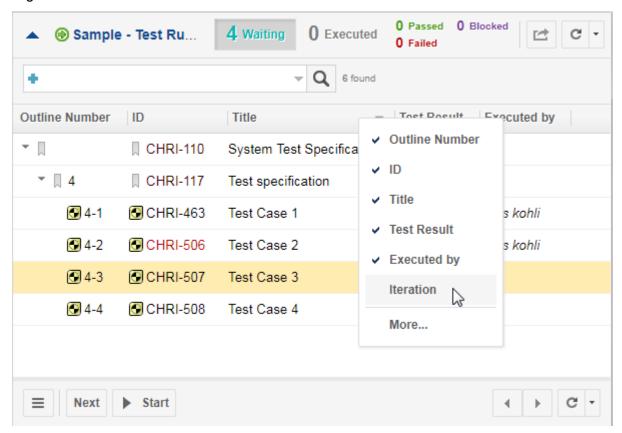


- Iterations are shown as multiple rows in the tree view.
- Only iterations matching the query/filter are shown.
- Child items are shown under the last displayed iteration of the parent. (If the parent has them.)
- If no iteration of a parent matches the current query/filter but some children do, the parent is shown as a row representing the Test Case. But, the parent is not bound to any iteration or Test Record. It does not show any status and so forth.

## **Note**

Adding iterations directly into the execution view is not possible. They can only be added prior to execution using the Select Test Cases command.

9. Right-click on a column to customize what columns are ✓ visible or hidden.



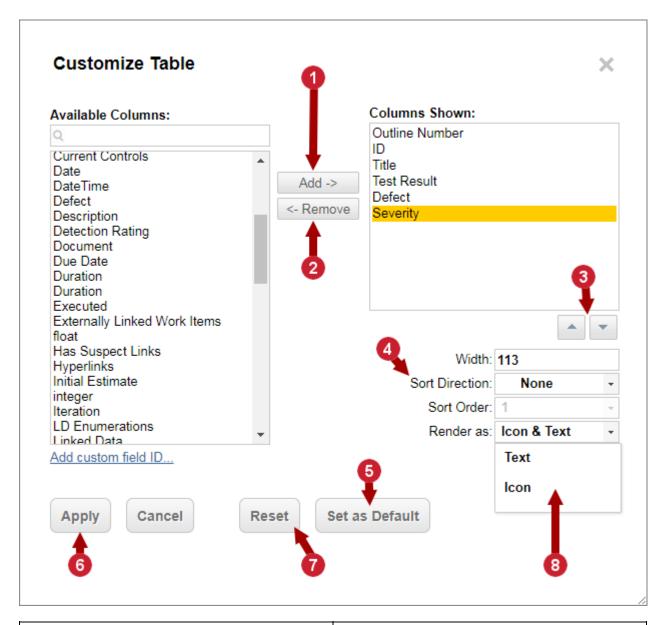
#### Note

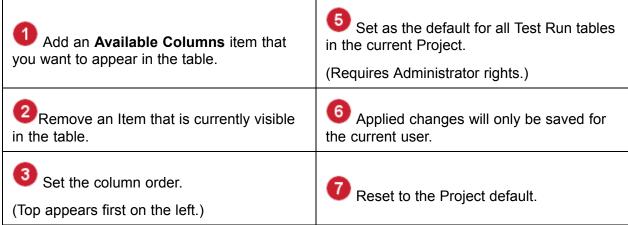
# Set Polarion to remember your table settings:

If you add or remove columns by right clicking and adding or removing a  $\checkmark$  or sort columns by clicking on them, your settings will be lost if the browser is refreshed.

For a persistent setup, click **More**, make your changes in the **Customize Table** dialog box, and then click **Apply**.

10. Click **More** in the column context menu to launch the **Customize Table** dialog box. There you can completely customize what columns are visible, how they are ordered and sorted, and even whether or not to display icons in the columns that support them.







4 Customize how column data is sorted.

What to display in columns that support icons and text.

- When you add a custom field ID, use the following syntax: testCase.fieldID.
  - Just the fieldID is fine for the Work Item table but not the Test Run table.
- When configuring sorting, the Test Record fields must precede any Test Case fields in significance.
- The Execute Test button widget supports a Sort Test Cases by parameter (for Test Runs that are not defined by a LiveDoc), but it has no effect when using execution view. (It is used in the Classic Test Execution View.) Instead administrators should define their desired default configuration here in the Customize Table dialog box. (The default sorting configuration is ascending by Test Case ID.)
- Administrators can also set repository wide Test Record table settings by editing the testrecord-table-settings.xml in the repository.

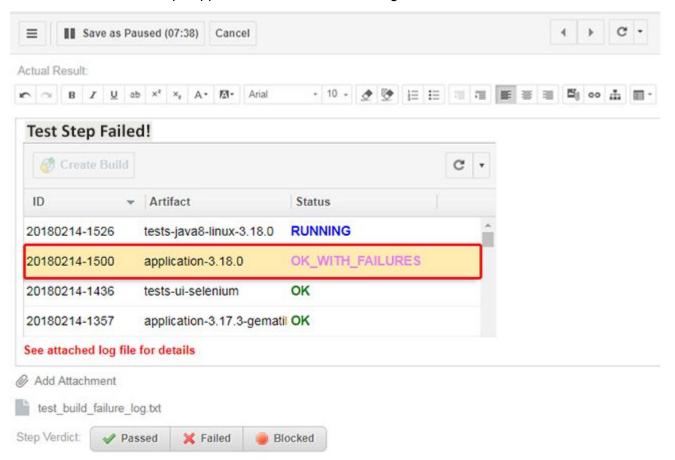
(repo/.polarion/testing/configuration/testrecord-table-settings.xml)

- 11. (Optional) Send a custom link to what you are testing to another user.
  - Click on a Test Case in the table.
  - b. Hover over the Test Case's title below the table and click ▼ when it appears.
  - c. Click Copy Link to this Test Record.
  - d. The browser link is copied to your clipboard.

- URL parameters also support iterations. For example, in record=projectID/testCaseId\$2, the \$2 is the iteration number.
- The link can also be opened in a Baseline.
  - Or, combined with a Test Run revision to show the corresponding Test Record for a Test Case retested at a later date.
- The target Test Record appears highlighted in the table, even when the filter settings are set to hide it.
- If the target Test Record cannot be found in the Test Run, a warning message appears.
- The record's parameter remains in the URL until the selection is changed by the user.
- Click on the selected Test Case or Heading's title when it appears below the table to open it in a new browser tab.



- 12. Click **Next** to jump to the next available Test Case in the table.
- 13. Click Start to execute the selected Test Case.
  - If the selected Test Case was already started but Paused, Start will be replaced with Resume.
  - If the Test Case has been run but needs to be run again, a Retest button appears.
  - If the button is disabled, hover over it for a tooltip that explains why it is.
- 14. The Test Case's Test Steps appear below and the timer begins or resumes.



- Click Save as Paused to pause and save Test Step without a verdict.
- You can enter and format rich text using the Document Editor toolbar and even add images directly into the result field of the test step.
- Click Add Attachment to attach a reference file.

Hover over an already attached file and click 8 to remove it.

#### Caution

# Limited rich text editing for offline test execution:

Offline test execution does not support the following rich text editor features:

- Images inside the text. (Like the example above.)
- Hyperlinks within the text.
- Tables within cells.
- Bullet lists.

These limitations also apply to the Excel Round-trip feature.

- 15. Click on a Test Step Verdict (**✓ Passed**, **X Failed** or  **Blocked**) and continue on to the next test step.
- 16. (Optional) Click **Recent** to select from a list of previously entered Test Step or Test Case verdict comments.



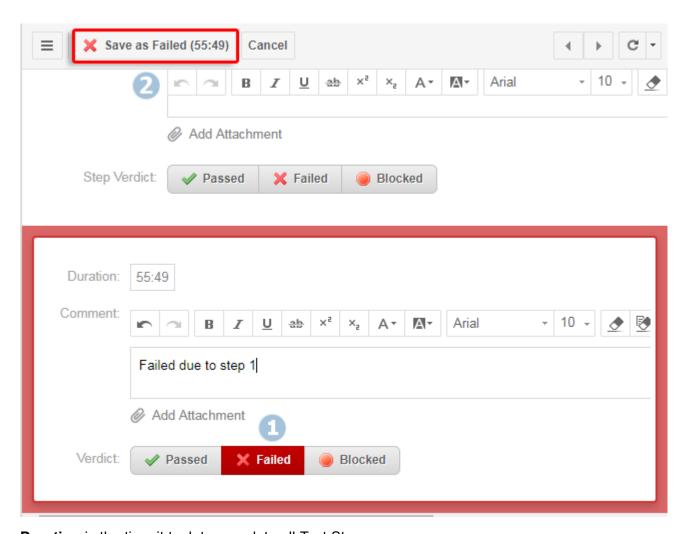
The Recent menu displays comments from any Test Run (by any user), that contains the selected Test Step or Verdict.

Comments from a previous iteration in the same Test Run are also included.

#### **Note**

The selectable comments support rich text formatting and offer an image preview, but only insert a text placeholder for the images.

17. Below the final step you'll see a timer that displays the total **Duration** of the entire Test Case.



**Duration** is the time it took to complete all Test Steps.

A timer is also visible at the top of the Test Case, just below the table.

18. Click on a verdict button and click Save as [Result].

The [Result] changes according to the verdict you selected.

19. If the project was created using any of the following templates, then a customizable verdict banner appears below the Test Cases title:

(Templates: Agile, E-Library, V-Model, Drive Pilot, V-Model QA, Drive Pilot QA.)



(1 Taken from the comment entered in the verdict.)

Tip

Administrators can also configure what appears in the section above the Test Steps.

See 

✓ Administration 

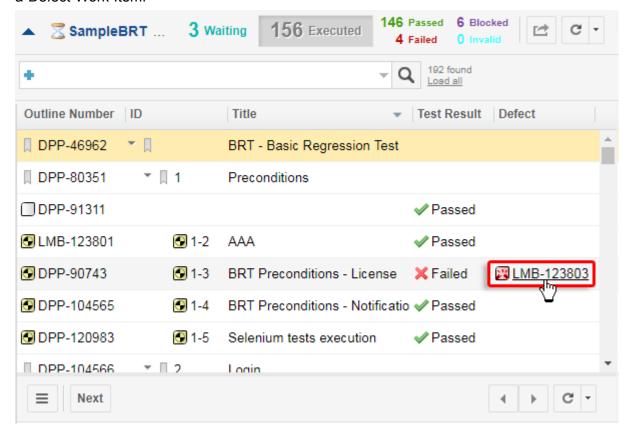
✓ 

▲ Testing 

✓ 

▲ Test Execution Form for details.

20. With the default configuration, any Test Case **Saved as Failed** will automatically create a Defect Work Item.



Click on the defect in the table to launch it in a new browser window.

21. You can manually proceed to the next Test Case by clicking **Next** under the test execution table.

#### Note

#### Test execution and Baselines:

You can view Test Runs executed in a previous Baseline, but you cannot execute them.

However, the **Waiting** button can be used to see the last Test Cases that were waiting at that point in the history and open the Test Run's execution view.

(Only for LiveReport based Test Runs created with the execution view implemented in Polarion 18.)

#### **Note**

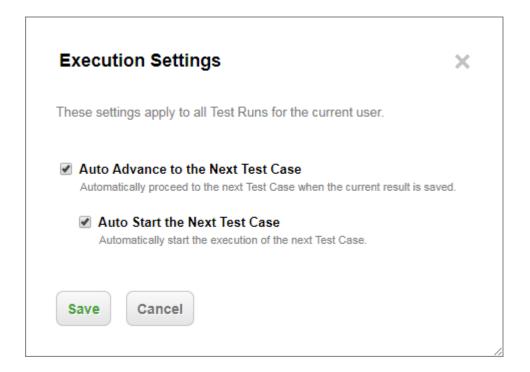
# **Empty Test Step values:**

Empty Test Step values are omitted to save space and increase readability.

# **Auto Advance and Auto Start the next Test Case**

You can edit Test Run **Execution Settings** so that Polarion automatically jumps to, and even starts the next Test Case.

- 1. Click ≡ at the top of the active Test Case, and select **Execution Settings**.
- 2. The **Execution Settings** dialog box appears.

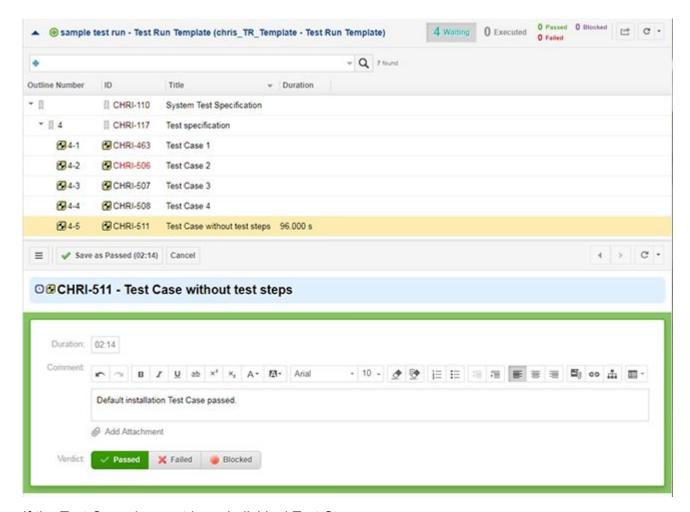


- 3. Check or clear the automatic execution options of your choice, and click **Save**.
- 4. Your selection is saved and tied to your personal user preferences so that they are the same for all of your Test Runs.

# **Execute test cases without Test Steps**

If your project is not configured for Test Steps or if a Test Case does not contain multiple steps in a Test Steps table, then the panel in the lower section of a Test Case contains only the Test Case Verdict field and buttons to mark the overall Test Case verdict as Passed, Failed, or Blocked.

You can still enter and format a comment in the Document Editor toolbar, add images directly into the test step result field, and Add Attachments.



If the Test Case does not have individual Test Steps:

- 1. Click the Start Test Case.
- 2. Perform all the necessary testing activity to determine the result of the Test Case.

#### **Execute or monitor multiple Test Runs simultaneously**

Multiple Test Runs can be run simultaneously, even when they contain the same Test Cases.

- 1. Launch two browser tabs with different Polarion **Test Runs**.
- 2. The Test Run's name is displayed at the top of the table, and the **browser URL** includes the selected Test Run.

# Sign executed Test Cases

Projects may be configured to require testers to electronically sign executed Test Cases. If your project is so configured, then when you mark the Test Case Verdict, a dialog box displays requesting you to sign by entering your user name and password. See also Signatures for Test Case Execution.

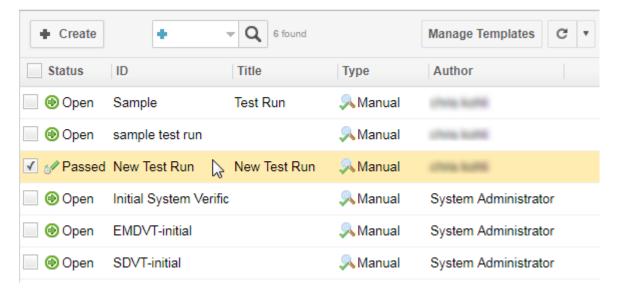
# Reuse defect or issue Work Items

By default, any unresolved defect Work Item (or equivalent type) from any Iteration in the current **Test** Run, or from any iterations in the previous Test Run (or the previous Test Run in the same group), is reused for failures of the same test case if the test case failed in the most recent execution.

In Test Runs that use test cases from different projects, this works differently. A defect item can be reused only if it is linked to a test record in a Test Run in the same project as the current Test Run. This also applies to Failed in previous and Failed in group reuse strategies. For example, suppose Project A uses the defect reuse strategy always. You execute a test case in a Test Run in this project, and it fails. The previous Test Run in which the same test case was executed is in Project B. This previous execution failed, creating a defect item that is not resolved yet. That defect, in Project B, is not reused. Rather, a new one is created in the project specified in the testing configuration in Project A.

# Show an executed test for a finished Test Run

1. Select a completed **Test Run** from the table of available Test Runs.



- Click ♣, and then ♣ Test Run Records. Or, scroll down to the bottom of the page, and click Browse All Test Run Records.
- 3. The Test Run's Test Record list appears.

# Tests - sample test run

Test Result	Test Case	Defect	Duration	Executed by	Executed
→ ✓ Passed	CHRI-463 - Test Case 1		18.000 s	*	2018-03-08 13:28
▶ X Failed	CHRI-506 - Test Case 2	CHRI-513	3 21.000 s	<b>R</b>	2018-03-08 13:29
→ ✓ Passed	CHRI-507 - Test Case 3		14.000 s	*	2018-03-08 13:30
→ ✓ Passed	CHRI-508 - Test Case 4		15.000 s	2	2018-03-08 13:29
✓ Passed	CHRI-511 - Test Case wi	thout test step	s 108.000 s	2	2018-03-08 13:29

4. Click on a Test Case to view its details.

(Test Cases can not be retested in a finished Test Run.)

#### Classic test execution view

# ALM REQ QA PRO

testManagement.useOldExecutionView=true

## Warning

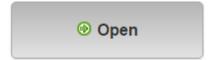
To give current testers some time to transition to the new Test Execution View workflow, we will still support the *Classic Test Execution View* for a few more releases.

**Recommendation**: Get to know the improved workflow outlined in the Executing a Test Run Using Polarion section.

Execute a manual Test Run with the classic Test Execution View

- 1. Open the Test Run's page by clicking on the **Test Runs** in Navigation, and selecting the Test Run in the table of Test Runs. If there are many Test Runs, you can use the visual query builder on the Test Runs page to build a query to filter the table. For example, you might query for only Manual type tests, based on a specific Test Run Template.
- 2. Click **Execute Test**. The Work Items table opens in a new browser tab with the Test Cases selected by the Test Run configuration listed in the table.

Test Run Status: (click to modify)





- 3. Select the first Test Case in the table. It's detail loads in the lower half of the page.
- 4. Scroll down to the **Execute Test** section. If your project is configured for Test Steps support, this section contains the Test Steps from the Test Steps table in the Description field, and you can log the results of each individual test step.
- 5. When you are ready to begin executing the test, click **Start Test Case**.

While executing the Test Steps for a Test Case, you may find it necessary to pause the test execution and resume testing later. After you start executing Test Steps, the Pause Test Case button appears in the Test Steps. When you pause the test execution, the results of completed Test Steps are stored and the pause is recorded in the Test Run history. To resume testing, click Resume Test Case.

#### Tip

It is possible that another user could modify the Test Case or Test Steps while you have paused test execution. If this happens, the Resume Test Case dialog box (if Test Case was revised) or **Retest Test Case** dialog box (if Test Steps were revised) appears.

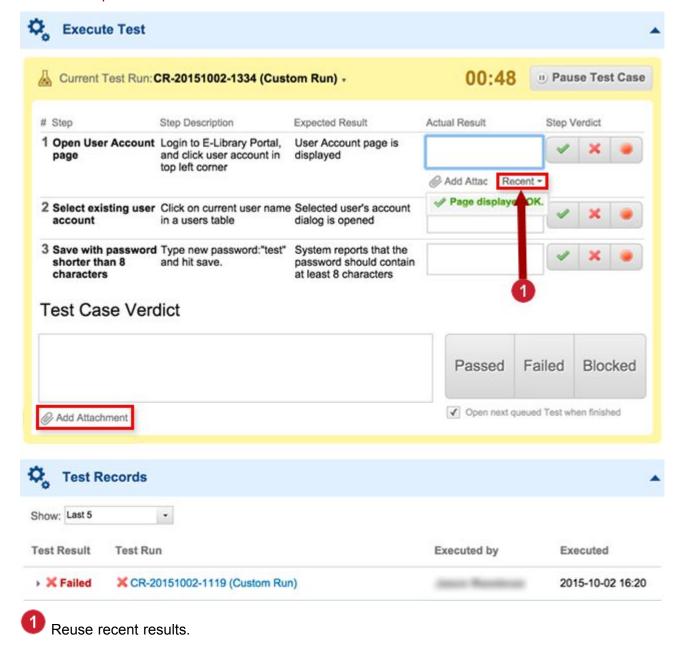
- **Resume Test Case**: Warns that you are no longer testing what was planned. For example, the Test Case was planned from HEAD revision, and the Test Case subsequently was updated while the Test Case was in progress. Or possibly, the revision in which the Test Case was planned changed (for example, frozen to a different revision in a Test Run where Test Cases are selected via the LiveDoc on Execute option). In both scenarios, after clicking Resume Test Case in the dialog box, the Test Steps remain the same, and you continue working in the revision in which you started.
- Retest Test Case: Means that the Test Case (including the Test Steps) is updated and retesting needs to be done in the new version. If you click **Retest Test Case**, the Test Case is reloaded and execution is automatically started.

# **Executing Test Steps**

If the **Execute Test** panel has multiple Test Steps:

- Click Start Test Case.
- 2. Execute the first test step, note the result of the step in the **Actual Result** field, and click the button that indicates the result of the step: Passed, Failed, or Blocked. You can select from a list or previous results for the current step by clicking **Recent** (shown below).

You can optionally attach a file (a screenshot image, for example) to the test step. An **Add Attachment** button appears when you click the **Actual Result** field. For more information, see Test Step Attachments.



## Caution

If the Test Case's **Description** field contains a table containing Test Steps, while it's possible to merge the cells in the table this not recommended.

It may render the Test Steps unusable.

- 3. Repeat the above for all Test Steps listed in the section.
- 4. If there are multiple Iterations of the Test Steps, continue testing until you have completed all steps in all Iterations, or until something fails.
- 5. (Optional) Enter a comment in the **Test Case Verdict** field and click on the button marking the overall Test Case result: Passed, Failed, or Blocked. If your project has been set up to require it, you are asked to log your electronic signature before proceeding further. A record of the signed test execution is then visible in test records and displayed in Test Cases, Test Runs, testing report pages, and so forth.



If there are untested Test Cases in the table, and the open next queued Test when finished option is selected, the next Test Case in the Test Run is automatically selected and ready for testing to begin on that Test Case. As long as the option is checked, this happens until all Test Cases in the Test Run are executed. In Test Cases with Iterations in the steps, the process proceeds through the Iterations before moving on to the next Test Case. The statistics in the Test Run Planning sidebar update to reflect the progress of the testing session.

#### Note

It is not required to mark results for any of the Test Steps. You can mark all, some, or none. You need only log result of the Test Case: Passed, Failed or Blocked. However, Actual Result of every step is still written to the test record even though the test step has no verdict.

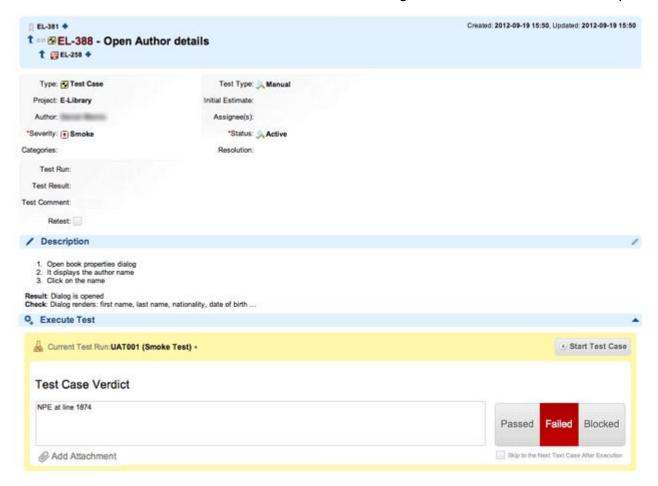
#### **Note**

The Navigation scope can change depending on the setting in the Test Run's Project Span property. If multiple projects are specified in that setting, the scope switches to *Repository*. If just one project is specified, the scope switches to that project. If Project Span is empty, the scope remains in the current project.

To return to the Test Run after selecting Test Cases, regardless of current Navigation scope, click on the Test Run name in the **Test Run Planning** sidebar. If not already open, the project containing the Test Run opens and the Test Run page loads in the browser.

#### **Execute test cases without steps**

If your project is not configured for Test Steps or if a Test Case does not contain multiple steps in a Test Steps table, then the panel in the **Execute Test** section of a Test Case contains only the **Test Case Verdict** field and buttons to mark the overall Test Case verdict of Passed, Failed, or Blocked. You can also add attachments to the Test Case Verdict using the **Add Attachment** link in the panel.



If the **Execute Test** panel does not have individual Test Steps:

1. Click Start Test Case.

- 2. Perform all the necessary testing activity to determine the result of the Test Case.
- 3. (Optional) Enter a comment in the **Test Case Verdict** field, and click on the button marking the overall testing result: Passed, Failed, or Blocked. If there are untested Test Cases in the table and the Skip to the next Test Case after execution option is selected, the next Test Case in the Test Run is automatically selected and ready for testing to begin on that Test Case. As long as the option is selected, this happens until all Test Cases in the Test Run are executed.

# Execute an unplanned test case

It is possible to execute a Test Case that has not been selected in a Test Run (That is, it has not been planned for execution in a Test Run). When the Test Case item is open and selected in the Work Items table, you can select a Test Run via the Current Test Run menu in the Test Steps panel of the Execute Tests section of the Test Case. When you click the Start Test Case button, if the Test Case has not been planned in the selected Test Run, a confirmation dialog box appears informing you that the Test Case is not planned in the Test Run. You are then asked to confirm that you want to execute the Test Case. You click **Execute** to proceed, or **Cancel** to cancel the execution.

# **Execute or monitor multiple Test Runs simultaneously**

Multiple Test Runs can be run simultaneously, even when they contain the same Test Cases.

- 1. Launch two browser tabs with different Polarion **Test Runs**. The **Test Run Planning** Sidebar and **browser URL** displays the selected Test Run.
- 2. The URL parameter, (containing the name of the originally selected Test Run), determines which Test Run is selected in the **Current Test Run** drop-down box when one of the Test Cases is opened or <a> refreshed</a>.

# Sign executed Test Cases

Projects may be configured to require testers to electronically sign executed Test Cases. If your project is so configured, then when you mark the Test Case Verdict, a dialog displays requesting you to sign by entering your user name and password. See Signatures for Test Case Execution.

#### Reuse defect or issue Work Items

By default, any unresolved Defect (or equivalent type Work Item) from any Iteration in the current Test Run, or from any iterations in the previous Test Run (or the previous Test Run in the same group) is reused for failures of the same Test Case if the Test Case failed in the most recent execution.

In Test Runs that use Test Cases from different projects, this works differently. A Defect item can be reused only if it is linked to a test record in a Test Run in the same project as the current Test Run. This also applies to Failed in previous and Failed in group reuse strategies. For example, suppose "Project A" uses defect reuse strategy always. You execute a Test Case in a Test Run in this project which fails. The previous Test Run in which the same Test Case was executed is in "Project B". This previous execution failed, creating a Defect item that is not resolved yet. That defect, in "Project B", is not reused. Rather, a new one is created in the project specified in the testing configuration in "Project A".

#### Show an executed test for a finished Test Run

- Select a finished Test Run from the list of available Test Runs.
- Click Passed, Failed or Executed in the Test Run Status section.

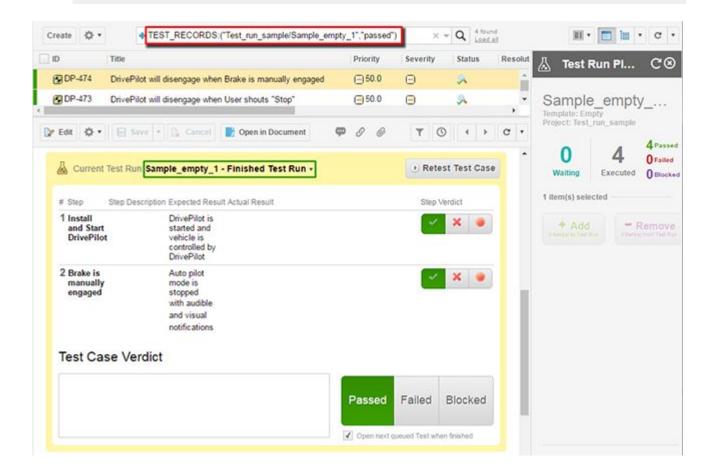
The Test Run's results appear in a list.

- Click Show in table.
- 4. The Test Run's details appear.

#### Tip

If you know the target Test Run you can also enter it directly as a search parameter.

(The results are read-only. Test Cases can not be retested in a finished Test Run.)



# **Execute a Test Run offline using Excel**



You can export Test Runs to Microsoft Excel, manually execute test cases and associated Test Steps offline/externally, and then import the results into Polarion via the Excel file previously exported. This

approach automatically creates test records in the system for all exported Test Case items. The approach can be useful if you use external testers who do not have access to your Polarion system.

You can also use the same method to manually record the results of externally-run automated tests in an exported Excel file, importing the results back to Polarion. However, if you use automated testing tools, you may prefer to set up automated import of automated test results into Polarion. See Integrating with External Testing Tools for more information.

## Before you begin

On export, all Test Cases selected for the test run are exported. The export includes any Test Cases from other projects selected via the Project Span field of the Test Run or the underlying Test Run Template. On import, Test Records are created for all the previously exported Test Cases. If Test Case selection spans across different projects, the following limitations currently apply:

- The Excel export template must be stored in the project that contains the Test Run or in the Repository scope.
- If the Test Run selects Test Cases from multiple projects (selected via the *Project Span* field of the Test Run or the underlying Test Run Template), then all the projects containing Test Cases and the project containing the Test Run must have the same Test Steps columns defined and the same ID for the Test Steps field of all Test Case type Work Items selected. (This assumes that the Test Steps field of Test Cases is exported to Excel.)

These limitations may be removed in a future Polarion release.

# **Exporting test cases to Excel**

Begin the external testing process by exporting the Test Run to be executed externally to a Microsoft Excel workbook file.

- 1. Open the project containing the Test Run you want to execute externally.
- 2. In Navigation, select **Test Runs**. On the Test Runs page, select the Test Run you want to manually execute (or for which you want to manually record results of external automated tests).
  - All relevant Test Cases should already be selected. If not, select them as described in Select (Planning) Test Cases.
- 3. In the Test Run detail, click **Export Tests to Excel** and save the exported Excel file to your local file system. The default file name is the name of the Test Run. You can optionally save with a different name without affecting subsequent re-import of test results.

You can send the exported file to someone else who will actually perform the manual tests (or manually record the results of externally-run automated tests). The tester can execute each of the Test Cases of the Test Run in whatever way that is normally done, performing the steps by hand or running some tool and logging the result of each Test Case in the Excel file. When manually executing tests, as each Test Case is executed, the tester logs the result of each test step in the Step Verdict column of the Excel sheet for each Test Case and records the overall result of each Test Case in the **Test Verdict** column. Testers can optionally add or change the content of the **Comment** column. These columns, and those for logging results, are the only ones external users can edit. All others are protected.

# Importing test results from Excel

After all tests for the Test Run have been performed and their results entered and saved in the Excel file, the tester returns the file to the Polarion user who exported the Test Run (or any user with the same permissions), who then imports the results file back to Polarion so that the results can be recorded and tracked.

- If you make changes to the Excel file before importing, be sure all changes are saved before initiating the import.
- Open the same project from which the Test Run was exported, if you do not already have it open.
- On the Test Runs page of Navigation, select the Test Run from which the Test Cases were exported to Excel.
- In the lower pane of the Test Run page, click the Import Results from Excel link.
- In the **Import Results from Excel** dialog box, click **Browse** and select the previously exported Excel file on your local file system.

If you select the **Retest previously executed Test Cases** option, test records in the Excel file will overwrite any test records that may already exist in Polarion. If cleared, any existing test records in Polarion are preserved and those in the Excel file are ignored.

• Click **Preview**. If changes are detected, a preview appears. Click **Import** to complete the import process. If no changes are detected, the dialog informs you and you can click **Cancel**.

## Tip

If your project is configured to require an electronic signature for Test Case execution, you are prompted to provide your e-signature. A signature must be provided for the process to finish. Nothing is imported if you cancel the import without providing the requested signature.

When the process is finished, the dialog informs you of the result: successful import or import failure. Assuming success, it displays statistics about the number of Test Cases executed and the number that passed. In every case, there is a link to the import log file which displays the process log in a new browser instance.

A successful import creates a new Test Run record. You can access records via the (Actions) menu button in the header of the Test Run detail pane on the Test Runs page.

#### Note

Adding new Test Cases in an Excel workbook exported for external test case execution is not allowed in the exported file. If you want to use Excel to define new Test Cases, use the Import from Excel feature to define new test cases in Excel and import them into a Polarion project.

#### Test export templates

Polarion provides a default template for exporting Test Cases to Microsoft Excel for external testing. It should be sufficient for most situations. However, administrators can download the default template file which can then be used as the basis for one or more custom export templates. One customization that may be desirable when exporting Test Runs that select Test Cases from multiple projects is to show the source Project of Test Cases in a dedicated column.

An administrator can upload customized template files, which users can subsequently select as templates for exporting Test Runs to Excel. For information, see Configure Test Export Templates.

# Sign a Test Run

Polarion can support your formal testing review and sign-off process for Test Runs, using secure electronic signatures compliant with U.S. 21 CFR Part 11. A Polarion administrator can map your review and sign-off process to the Test Run Workflow configuration. Each step in the process can optionally be set up to require an electronic signature before a Test Run can move to the next step in the workflow.

To be able to sign a Test Run you must have permission to READ and permission to MODIFY Test Runs. Signing a Test Run takes place when you invoke any status change that is configured in the project workflow as requiring a signature. For example, setting the status to Passed might be so configured.

- 1. Open Test Runs topic (Navigation \*\* Test Runs) and select the desired Test Run.
- 2. On the toolbar of the detail pane, click **Properties**.
- 3. Go to the **Status** field and select the desired transition action. The actions in the list can vary from project to project depending on the project's workflow configuration. For example, one project might have an action "Perform action Mark as Passed" while another project might have an action "Mark Executed Without Failures".
- 4. On the pane toolbar, click **Save**. If the selected action is configured to require a signature, the Enter User Name and Password dialog box appears with the Username field focused.
- 5. Enter your Polarion user name and press **Enter**, or click **Next**. The **Password** field appears.
- 6. Enter your password and press Enter or click Sign.



#### Tip

- You can show Test Run signature information in Pages using the Test Run Signatures widget.
- In the rare case where logging a signature conflicts with concurrent changes to the Test Run made by another user, Polarion displays a dialog box with information and options for proceeding. You can review the other user's changes before saving yours.

# Adding Comments to a Test Run

The Properties page of a Test Run contains a section for comments. When signing a Test Run, or making other changes, you can add a comment. Test Run comments work in the same way as comments in Work Items. Other users can reply to a comment, there can be multiple comment threads, and comments can be marked as Resolved.

On the Test Runs page, users can search for a Test Run by comments by typing a guery containing fields comments.title, comments.text, comments.author.id, or comments.resolved, or via full-text search.

**Example**: status:failed AND comments.title:scheduled AND comments.author.id:rproject

#### **Test Run Comments**

Test Runs support comments and comment threads. If you have the required user permissions (assigned by an administrator to your user role in your project), you can:

- Add new comments to a Test Run (requires permission to comment Test Runs).
- Reply to existing comments (requires permission to comment Test Runs).
- Resolve comments or comment threads (requires permission to resolve comments in Test Runs).

Note that permission to read Test Runs is also required for all of the above.

Access Test Run comments:

- 1. In Navigation, select ATest Runs.
- 2. On the Test Runs page, select the Test Run you wish to view.
- 3. In the Test Run detail, click **Properties**, and then click the (Scroll to Comments) icon.

#### **Delete old Test Runs manually**

# ALM QA

At some point you may decide you don't need to preserve all the executed Test Runs. This is especially likely if you have integrated automated test execution with Polarion's building feature, creating new Test Runs with every build. You can delete one or more Test Runs from the LaTest Runs topic. You must have permissions for this action, otherwise you will see a message and the delete operation will be canceled.

Delete one or more Test Run(s):

- 1. Select the **ATest Runs** topic in **Navigation**.
- 2. In the table of existing Test Runs, select the Test Run(s) you want to delete. The page provides an instance of the Visual Query Builder that you can use to visually build a query to filter the list. This instance of the tool is limited to a set of query elements applicable for querying Test Runs, and does save queries for later reuse. (A visual guery built with this tool can be converted to text for use in reports and so forth.)

To select multiple Test Runs for deletion, select the check box on the row of each Test Run to be deleted. To select all Test Runs appearing in the list, select the check box in the header of the Status column.

3. If you are deleting a single Test Run, go to the lower pane, click (Actions), and choose **Delete**.

If you selected multiple Test Runs to delete, a button labeled **Delete [N] selected Test Runs** appears in the lower pane of the Test Runs page (where [N] is the number of Test Runs selected in the upper pane). Click the button to delete all the selected Test Runs.

# **Automated Cleanup of Old Test Runs**

If you have many Test Runs as a result of automated testing, it may not be practical to delete old Test Runs manually. In this case, Polarion provides an automated job that deletes Test Runs you don't need to keep. For information, see Configure Cleanup of Old Test Runs.

# **Preserving Test Runs in History**

Every Test Run has a **Keep in History** field, accessible in the Test Run's properties. When checked, the Test Run is preserved in Baselines. Before starting automated cleanup of Test Runs, you should mark those Test Runs to preserve in history for future reference. For example, you might keep all Test Runs for verification tests that were executed at some project milestone. Project managers should be aware of when the automated cleanup job is scheduled to run, and should mark any new Test Runs to preserve in history before the next scheduled Test Run cleanup.

Marking is simple when there are only a few Test Runs and only one or two to be marked. You can browse your project's Test Runs and mark each Test Run individually in its Properties. If there are many Test Runs in the system, and/or many to be flagged for preservation in history, then you need to use bulk marking of Test Runs with the Keep in History flag.

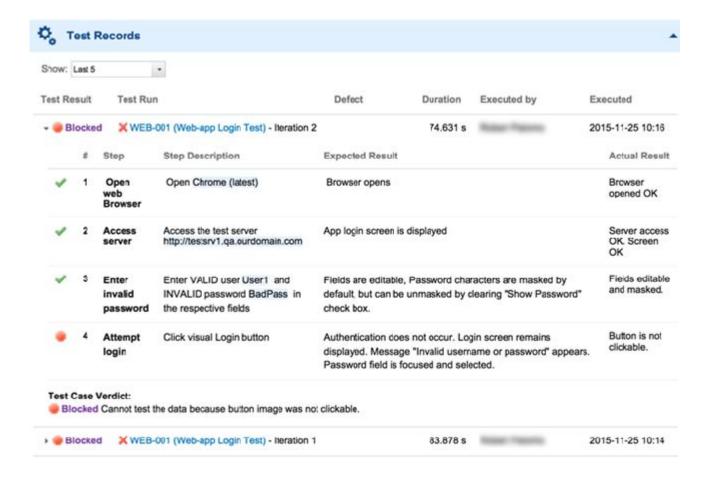
Bulk mark multiple Test Runs:

- 1. In **Navigation**, open the **ATest Runs** topic.
- 2. On the Test Runs page, in the table of Test Runs, check the box on the row of every listed Test Run that you want marked with the **Keep in History** flag.
  - If there are many Test Runs, you can run a query to filter the table. If your query returns only Test Runs that should be marked, you can mark all Test Runs in the table by selecting the check box in the table header.
- 3. In the selection panel in the lower portion of the page, click **Keep in History** to set the "Keep in History" flag on all selected Test Runs. (The selection panel appears when more than one Test Run is selected in the table.)

#### **Review Test Records**

Polarion maintains a record of all instances of the execution of a Test Case during any Test Run which includes the Test Case. You can review the most recent records for any Test Case in the **Test Records** section of the Work Item Editor when a Test Case is selected. You can view the past test execution records up to the limit set in the system configuration, selecting the period in the drop-down list provided. The Test Records section shows the results of the reported test executions, including **Iterations** of Test Steps and provides links to each of the Test Runs. The current content of the Test Records section is included if the Test Case is sent to Print. If the record table contains additional details, its content is expandable.

Each Test Record is linked to the revision of the test case that was executed during the testing session that resulted in the creation of the record.



#### **View Test Run history**

- 1. With the relevant project open, click **Large Runs** in Navigation, and in the table of **Test Runs**, select a Test Run by clicking on its row. The detail of the Test Run loads in the lower part of the page.
- 2. On the toolbar of the Test Run detail, click the W **History** icon. A table of the historical revisions to the test run replaces the Test Run detail information.

You can access the detail of any revision by clicking on the revision number. To exit the history, click the W History icon again.

#### **Use Test Parameters and Iterations**

#### **Overview of Test Parameters and Iterations**

In manual testing, it is common for the testing steps of a test case to be executed multiple times with variations in the test procedure. For example, the same set of steps might be run on desktop and mobile operating systems, and/or with different browsers on the different operating platforms, or an electromechanical component might be tested with different input voltages. Polarion's Test Parameters (or "Parameterized Testing") feature can help make the test specification process efficient and flexible.

The discussion of this feature assumes you are already familiar with the information in Creating and Managing Test Runs.

# Test parameters overview and example

Rather than create separate test cases, or different sets of steps, it is possible to capture the variables in the testing process as Test Parameters, which can be applied by testers during different executions of the same Test Case. Test Parameters are placeholders for actual values. They enable test case authors to write test cases and steps in an abstract way, thereby enabling testers to execute them with specific variations.

For example, suppose you are testing a logon operation for a web application. Your steps might be as follows:

- 1. Open Browser on your computer
- 2. Access the URL of server
- 3. Enter User name and Password in respective fields
- 4. Click the Login button

Suppose you want to run this test using different browsers/versions and/or different user accounts. Rather than write separate test cases with similar steps, Test Parameters named Browser, Username and Password can be created and inserted into test steps when writing up a test case.

Authors/planners of Test Runs can supply values for the Test Parameters in the Test Cases planned into individual manual Test Runs. Examples of parameter values in the above simple test might be <code>Firefox 40, sampleuser</code>, and <code>samplepass</code> in one Test Run, and <code>Chrome</code>, <code>sampleuser</code>, and <code>samplepass</code> in another Test Run. During execution of Test Runs, testers see these values and perform the steps accordingly. Test Records are automatically created showing the step results with Test Parameter values applied.

If a single Test Case needs to be executed multiple times with different variables applied, multiple *Iterations* can be defined for the Test Steps with different parameter values specified and applied for each execution of the steps.

## **Visibility of Test Parameters**

Changes to Test Parameters are visible in Polarion in several ways:

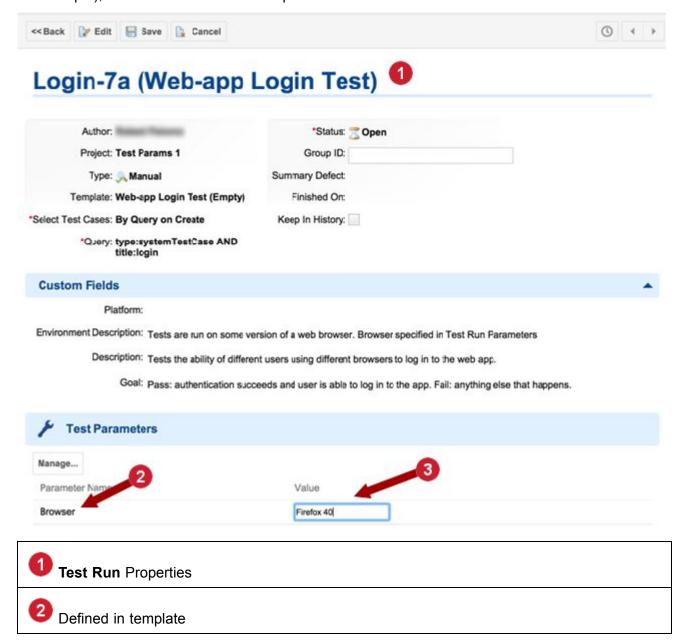
- All changes to Test Parameters, such as creating new, renaming, and changing values, are
  written to the History of the artifact in which the change takes place like Test Run Template,
  Test Run, or Test Case item.
- Changes to Test Parameters are shown in the Activity stream. The entry shows the user who
  made the change, what artifact was changed, and what parameters changed.
- By default, changes to Test Parameters are included in the information sent via notification events for Test runs

Also, Test Parameters can optionally be shown as a column in the Test Runs table via the **Customize Table** action on the toolbar of the Test Runs table. In the dialog box, you add **Test Parameters** to the **Columns Shown** list.

# **Parameters in Test Run templates**

This topic is mainly for testing managers or team leaders who plan testing activities and develop standards or best practices for a team or project by means of Test Run Templates. You can configure Test Run Templates with Test Parameters (and, optionally, default parameter values), so that when a Test Run is instantiated from a template, the Test Run contains the parameters (and values, if configured) defined in the template. When the instantiated Test Run is executed, the respective values from the Test Run are prefilled in Test Parameters in any Test Cases that have the same parameter in the Test Steps.

In the above example, it would make sense to define the Browser parameter in the Test Run Template because it is the testing environment and therefore global for all the Test Cases in all Test Runs based on the template. Then, in each Test Run, a value for Browser can be specified ("Firefox" or "Chrome" for example), which becomes the default parameter value in the Test Cases selected for the Test Run.





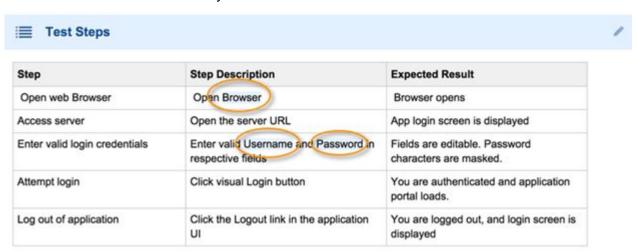
# Specified in Test Run

#### **Test Run Template Example**

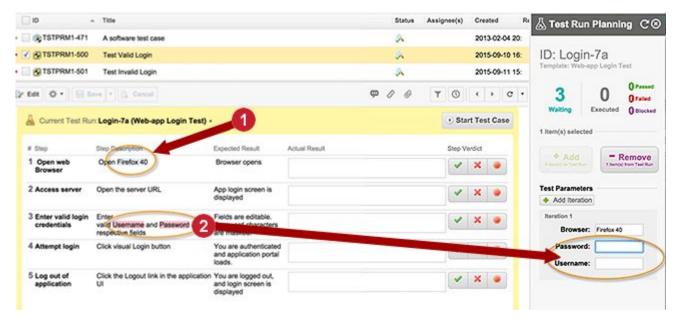
Consider a simple example that illustrates how this feature works, continuing with the idea of testing the logon function of a web application. The test environment is always a web browser. The testing steps direct the tester to open the browser, access the application URL, and enter the logon credentials user name and password. The test checks both valid and invalid credentials. Let us assume that a parameter *Browser* is defined in a Test Run template named "Web-app Logon Test". When a Test Run is created from that template, it contains the *Browser* parameter, but no value for it. This parameter can be saved to the Test Parameters Library, making it available to Test Case authors for insertion into Test Steps.

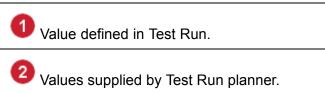
A default set of Test Parameters can be defined by an administrator in project Administration (Administration: Testing: Test Parameters). The parameters defined in this Test Parameters Library appear in the Add Parameter list in the Test Steps table of Test Cases, and Test Case authors defining Test Steps can select a parameter in the list to insert into a Test Step.

Now suppose you, the test planner, want a Test Run instantiated from this template to execute Test Cases using Firefox version 40. You supply Firefox 40 as the value for the *Browser* parameter in the Test Run you create for this test. Let's assume the existence of a Test Case in which the author created Test Steps containing Test Parameters *Browser*, *Username*, and *Password*, and that you have selected it for execution in your Test Run.



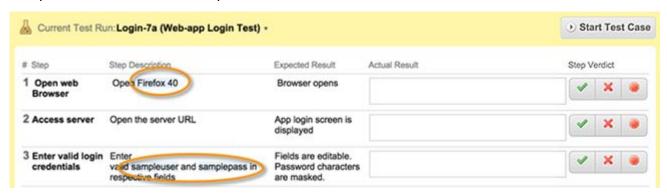
When you open the Test Case via **Actions** Select **Test Cases** in the Test Run, you see the value "Firefox 40" in place of the Browser parameter in the Test Steps. This value is from your Test Run. You now need to supply values for other parameters in the Test Steps, so testers executing the Test Run know how to actually execute the Test Case. You can do this in the Iteration 1 panel of the Test Run Planning sidebar.





After specifying all parameter values, you save the Test Run via the button in the sidebar. (You can optionally add Iterations and provide different values for the parameter.) At this point, the Test Run is ready to be executed by Testers.

When a tester executes the Test Run via **Execute Test**, the Table view of Work Items opens in a new browser tab with the Test Run Planning sidebar visible. The table lists the Test Cases selected in the Test Run. Test Parameters in the Test Steps are replaced by the actual values supplied by you, the Test Run author/planner. The tester proceeds to execute the steps according to the information. The parameter values in the steps are written to the test record.



## **Note**

A tester with the requisite permissions can modify parameter values during execution of the Test Run. Any changed values are written to the test record, so the actual values used to execute the test are always traceable.

#### **Define Iterations**

Test Steps of Test Cases often need to be executed multiple times with variations. In the example we have been using, this might mean running a different browser or browser version, or a logging on to a different user account with different access permissions. When defining a Test Run, the Test Run author/planner can add another Iteration in the Test Run Planning sidebar of any selected Test Case and supply the different values for the Test Parameters in each Iteration. You can add as many Iterations to be executed to cover all possible scenario variations.



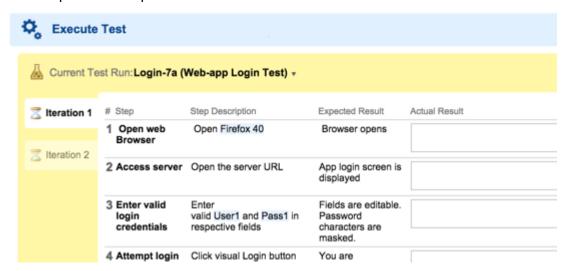
After Iterations are defined and the Test Run saved, a tester can execute the Test Run, performing the Test Steps multiple times using different values each time as specified in each Iteration. Test Step results are recorded as usual as each Iteration is executed, and Iteration results are part of the overall test record.

Testing can be paused and resumed any time until all steps for all Iterations are executed. When there are multiple Iterations, a tab for each one appears on the left side of the steps in the Execute Tests section of the page. The process moves the tester to the next Iteration when one is completed. (The tester can disable this automatic progression by clearing the checkbox option Open next queued Test when finished at the bottom right of the Test Steps in the Execute Test section of the Test Case.) When all Iterations are executed, the process moves the tester to the next Test Case, if any are still waiting.

## Warning

Iterations are only supported in Test Runs where **Select Test Cases** is set to Manually, (this includes Test Run Templates), or if they are selected via the Query on Create or LiveDoc on Create Options.

Test Steps with multiple iterations:



# Set up Parameterized testing

Generally, you want to perform the following operations in the order described, though there may be some concurrency, and some steps are optional, as noted.

- (Optional) Define the Test Parameters Library. (Project Administrator)
- (Optional) Define default Test Parameters in Test Run Templates. (Testing Manager or Lead)
- 3. Insert Test Parameters in the text of Test Steps in Work Items of the Test Case type. This step may take place concurrently with the next step. (Test Specification Author)
- 4. (Optional) Create Test Runs and define Test Run Parameters in the Properties. This step may take place concurrently with Test Parameter specification in Test Cases. (Testing Manager or Lead)
- 5. In the Test Runs, provide actual Test Parameter values, optionally adding Iterations, for the Test Cases selected in the Test Run configuration. (Testing Manager or Lead)
- 6. Execute Test Cases following the Test Steps through all Iterations. (Tester)

## **Define the Test Parameter Library**

# ALM REQ QA

When composing Test Cases, test specification authors who have the requisite permissions can optionally save Test Parameters they create to the project's Test Parameters Library. For example, if a

common test environment is a web browser, it would make sense to add a parameter named *Browser* to the library. The parameters saved to the library appear as items in the Insert Test Parameter select list in the Test Steps table of Work Items of Test Case type, when a Test Case author is editing the table and defining Test Steps. It is also shown in the Parameter Name column of the Manage Test Parameters dialog box, accessible in Test Runs and Test Run Templates.

Administer the Test Parameter Library:

An administrator can access this library in the project Administration and perform several operations including defining new Test Parameters and modifying or deleting existing parameters. In the Administration configuration, you can only specify Test Parameter name and type. Parameter values are specified by users who create Test Runs.

## Tip

In first releases of the parameterized testing feature, only the String type is supported.

Access and manage test parameters in the library:

- 1. Open the project while logged on with administrator permissions and enter Administration.
- 2. Expand the Testing topic and select **Test Parameters Library**.

#### **Note**

Users must be granted repository access permissions for the test-parameters-library.xml file in the project repository in order to add parameters to the Test Parameters Library. Writing to that file is not granted by default for nonadministrator users.

The path in the access management client is:

[PROJECT FOLDER]/.polarion/testing/configuration/test-parameters-library.xml

## **Define Test Parameters in a Test Run template**

# ALM QA

You can configure Test Run Templates with Test Parameters so that when a Test Run is instantiated from a template, it contains the parameters defined there. In essence, you define the default Test Parameters for all Test Runs based on the Template. You can add Test Run Parameters in both new and existing Test Run Templates. You can optionally provide default values for the parameters you define in a Test Run Template. The values appear in Test Runs instantiated from the template.

In a Test Run instantiated from a template, users with the necessary permissions can:

- Remove default parameters provided by the template.
- Change any default parameter values provided by the template.
- Define new parameters and values in the Test Run. These do not affect the template.

To work with Test Run Parameters in Test Run Templates, you must have permission to MODIFY Test Runs, and permission to DEFINE TEST PARAMETERS in Test Runs.

- 1. Open the project, and in Navigation select Test Runs.
- 2. In the table of existing Test Runs, click **Manage Templates**. The table now lists all currently defined Test Run Templates.
- 3. In the table of Test Run Templates, select the Test Run Template you want to configure.
- 4. In the detail of the selected template, click **Properties** and scroll down to the **Test Parameters** section. That section displays a table of currently configured parameters.
- 5. Click the section header, and then click **Manage**.
- 6. In the Manage Test Parameters dialog box, click the 🕈 icon to add a new Test Parameter. Select from the list of parameters currently in the Test Parameters Library or enter the parameter name and type (if multiple types are available). Click **OK** when finished adding Test Parameters.

The table in the **Test Parameters** section now contains the parameters you added. You can optionally enter a value in the **Value** column for any or all parameters. This value appears in Test Runs instantiated from the Template. When finished, click **Save** to preserve changes, and then click **Back** to exit from Properties. If you want to configure another Test Run Template, select it in the table and repeat the foregoing procedures.

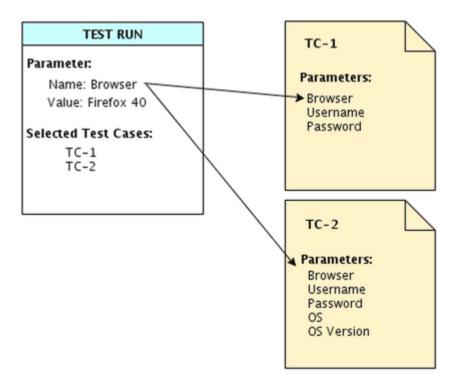
When finished with all templates, click **Manage Templates** to switch it off and return to the Test Runs management page.

#### **Define Test Parameters and Values in Test Runs**



This section applies to anyone who plans tests, creating and configuring Test Runs. When you create a new Test Run, you can define Test Parameters and their respective values. In the Test Cases selected for the Test Run, if the Test Steps contain a parameter of the same name as one defined in the Test Run, the Test Run supplies the value when the Test Cases are executed by testers. The Test Run author/planner must supply values for Test Parameters in Test Cases that do not have an equivalent parameter defined in the Test Run. Alternatively, testers can be given the necessary permission to modify parameters when executing the Test Cases of a Test Run.

For example, Test Case *TC-1* might contain a step open Browser where "Browser" is a Test Parameter. A Test Run might then define a parameter *Browser* with a value Firefox 40. When *TC-1* is selected for the Test Run, the tester sees the step as Open Firefox 40 - the value "Firefox" is supplied by the Test Run during execution.



If *TC-1* contains a step <code>Enter Password in password field</code> where Password is a Test Parameter, and an equivalent parameter is not defined in the Test Run, then the Test Run author/planner must provide a value (such as <code>Pass999</code>) when configuring the Test Run, so the tester will see <code>Enter Pass999</code> in <code>password field</code>.

To specify Test Parameters, you should be familiar with the Test Cases or the needs of Test Case authors who are writing the Test Cases and Test Steps that will be tested when executing the Test Run. For example, if some Test Cases need to be executed in different web browsers, you can understand that it's useful for the Test Run to have a parameter named *Browser*.

Define Test Parameters in a Test Run:

- 1. In your project, navigate into the Test Runs topic.
- 2. On the Test Runs page, select the Test Run you want to configure or create a new Test Run.
- 3. In the detail of the selected Test Run, click **Properties** and scroll down to the Test Parameters section. This section contains a table of currently configured **Test Parameters**. In a new Test Run, any parameters configured in the Test Run Template on which the Test Run is based appear in the table.
- 4. Click the **Test Parameters** header, and then click **Manage** to open the **Manage Test Parameters** dialog box.
- 5. Click **Select Parameter**. The list contains Test Parameters in the **Test Parameters Library** that are not currently used in the Test Run.
- 6. If you want to add one of the library parameters to the Test Run, select it in the list, and select the type (if multiple types are listed).

If you want to define a new parameter in this Test Run, select Add New in the list. In the Add New Parameter dialog box, enter the parameter name and specify the type (if multiple types are available). If you want the new parameter to be available project wide, select the Add to Library option (enabled only if you have permissions to add to the library).

- 7. After adding all Test Parameters, if you want to provide values for any of them, provide them in the **Value** field of the respective parameters.
- 8. Click **Save** to preserve changes, and then click **Back** to exit from Properties.

Your next task is to open the Test Cases selected for your Test Run and provide values for any Test Parameters they contain for which there is no equivalent Test Parameter in the Test Run. For example, if some Test Case has a parameter *User* that is not also configured in the Test Run, you need to provide a value for it at the Test Case level.

Supply values for Test Parameters in Test Cases:

- Select the Test Run on the Test Runs page and make sure Test Cases are already selected. (Selection mode must be Manual, By query on create or From LiveDoc on create.)
- 2. With the Test Run selected in the table, click (Actions) and choose **Select Test Cases**. A new browser tab opens displaying the Work Items table, which lists the selected Test Cases. The Test Run Planning sidebar also opens in the view displaying information about the number of waiting executions and other Test Run statistics.
- 3. For each of the Test Cases, in the sidebar panel labeled **Iteration 1**, enter values for all empty parameter fields.
- 4. Click **Save Test Run** to update the Test Run.

By default, each Test Case has one Iteration. If a Test Case needs to be executed multiple times with different parameter values, click **Add Iteration**. A sequentially numbered Iteration panel is added to the sidebar, with parameter values prefilled from the previous Iteration. Modify the parameter values as needed so that testers process the steps correctly. Add additional Iterations until all necessary variations of the Test Case are covered. Be sure to save changes using **Save Test Run**.

When all Test Cases of the Test Run have values, you can close the browser tab and return to the Test Run, which is now ready for testers to execute unless you wish to make other changes.

# **Use Test Parameters in Test Steps**

# ALM REQ QA

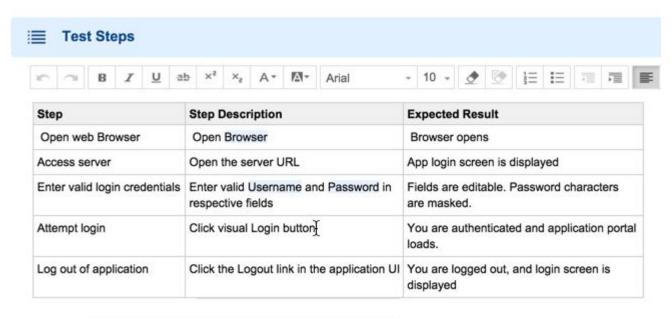
This section applies mainly to authors of Test Cases. The Test Parameters feature (sometimes called "parameterized testing") enables you to write abstract test procedures, inserting Test Parameters in place of some concrete specifications. When selected in a Test Run, a test planner, manager, or a tester with the requisite permissions can supply concrete values for the parameters in the Test Steps. These values become part of the automatically created test records.

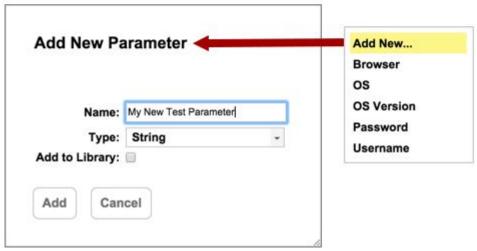
For example, in a Test Case that describes a test that must be run on different mobile operating systems, instead of writing "Android", "iOS", or "Windows Mobile" in a Test Step, you could insert a parameter named os. When planning the Test Case to a Test Run, a test manager can then supply a concrete value in place of os, depending on which platform is being tested.

The Test Parameters can be defined in any or all of several different places:

- In the project's Test Parameters Library in project Administration
- In Test Run Templates
- In individual Test Runs
- In Test Steps of Test Cases as they are being written

When writing a Test Step in a Document or in the Work Items Table, you can insert any existing Test Parameter (already defined in the Test Case, or the Test Parameters Library) into your text. Or, you can define a new Test Parameter and insert that, optionally adding it to the library. You must have access permissions granted for the relevant folder in the Polarion repository in order to add Parameters to Library. Writing to this location is not granted by default for nonadministrator users. (See define the Test Parameters Library.) You must also have the Polarion user permission "ADD TO PARAMETERS LIBRARY".





The list includes parameters already defined the Test Case and/or Test Parameter library. Insert a Test Parameter in a Test Step:

- 1. Click the header of the Test Steps table in the Test Case. The table transitions to edit mode.
- 2. Place the insertion cursor at the point in the table where you want to insert a Test Parameter. This is typically in the **Step Description** column.
- 3. In the drop-down options list beneath the last row of the table, click **Insert Test Parameter**.
- 4. Select one of the parameters in the drop-down list; you can filter the list by typing. Alternatively, create a new parameter by clicking the Add New item and entering properties for a new parameter in the Add New Parameter dialog box.

## Provide parameter values in Test Cases



This section applies mainly to a testing manager or team leader creating and setting up Test Runs. During planning of a Test Run, the author/planner must supply values for any Test Parameters defined only on the Test Case level (that is with no equivalent parameter in the Test Run). For example, if the Test Case author defined a parameter OS, the Test Run author/planner must replace that with some concrete value, such as ios or Android, if there is no parameter in the Test Run named OS.

When the **Select Test Cases** field of the Test Run is set to *Manually*, *By Query on Create*, or *From* LiveDoc on Create, you can click Waiting in the Test Run Status section of the Test Run to load the selected Test Cases in the Work Items table. Then you can begin supplying parameter values per the steps outlined below. When the Test Case selection mode is set to By Query on Execute or From LiveDoc on Execute, Test Parameter values cannot be specified at the Test Cases level. Values must be set in the Test Run, from which the Test Cases selected on execute will inherit the values. Also, multiple iterations per Test Case are not supported for the "on Execute" selection modes.

Provide Test Parameter values in the Test Cases of a Test Run:

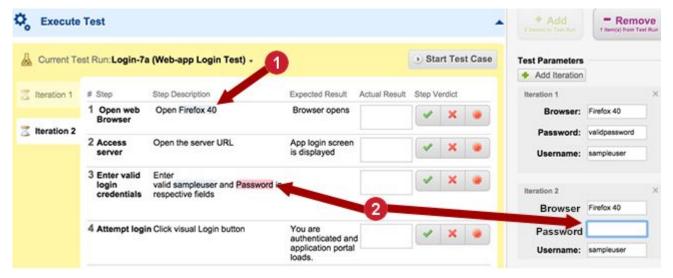
- 1. Select the Test Run in the Test Runs topic in your project.
- 2. Click Waiting in the Test Run Status section of the Test Run. A new browser tab opens and displays the Work Items table, with the Test Cases of the Test Run listed and the Test Run Planning sidebar open.
- 3. Select the first Test Case in the table, and in the Parameters section of the sidebar, specify values for any Test Parameters that do not yet have one. Any filled-in values are coming from the equivalent parameter in the Test Run. You can overwrite these values in the Test Case, if desired.
- 4. If the Test Steps need to be executed multiple times with different parameter values, click Add **Iteration** and provide the necessary parameter values. Create any additional Iterations needed. See Defining the Iterations, below.
- 5. Click **Save Test Run** in the sidebar to update the Test Run.
- 6. Click the (Refresh) icon in the editor toolbar of the Test Case and check that all parameters are replaced with actual values in all Iterations in the Execute Tests section of the Test Case. Any parameters missing a value are highlighted in light red.

Repeat these steps for all other Test Cases of the Test Run listed in the table until you have provided Test Parameter values for all of them.

Define multiple Iterations of Text Steps in Work Items of the Test Case type:

- 1. Be sure you have specified all missing values in **Iteration 1** (always present and cannot be removed).
- 2. Click **Add Iteration** in the sidebar to create another Iteration.
- 3. Fill in values for all Test Parameters. Repeat this step to create as many Iterations as you need to run the Test Steps with all possible parameter value variations.
- 4. Click **Save Test Run** in the sidebar to update the Test Steps.
- 5. Click the (Refresh) icon in the editor toolbar of the Test Case.

You should now see all actual values for all Test Parameters in the Test Steps, highlighted in light blue. Any undefined parameters are highlighted in light red. You should see tabs on the left-hand side of the Test Steps, one for each Iteration you defined. You can select the tabs and check that the Test Parameters all show the correct values.



Fix any still-missing parameter value(s) in the Iteration(s) via the Test Run Planning sidebar, save the changes, and refresh the view to check again that all parameters show the correct values. Repeat these procedures for all Test Cases of the Test Run, selecting each on in the Work Items table in turn.

#### Warning

Testers cannot execute a Test Case until all Test Parameters in all Iterations have a value. If execution cannot proceed for this reason, a dialog appears informing the tester about missing Test Parameter values.

# **Execute Parameterized Test Runs**



This section is mainly for testers who execute Test Runs that include Test Cases with Test Parameters in the Test Steps (that is, parameterized Test Cases).

Before executing Test Steps in a Test Case, a value for all Test Parameters in the Test Steps table must be specified, except those whose value is specified in the Test Run you are executing. Normally this should already be done by the creator of the Test Run. However, testers can be granted the requisite permission to modify Test Runs and define Test Parameters. In this case, it is possible to provide values for Test Parameters in the Test Planning Sidebar during execution of a Test Run. You should follow the procedures described in Provide Parameter Values in Test Cases before you begin executing the Test Steps, except in Step 2 you should use the Execute Tests button rather than the Waiting button. The Test Run Planning sidebar opens automatically only if you are using a license that allows Test Run modifications (such as ALM or QA).

#### **Note**

It is possible to override any default parameter values provided by the Test Run if you have the requisite MODIFY permission for Test Runs.

Once all Test Parameter have been supplied with values, you are ready to begin executing the Test Case. Launch the Test Run in the normal way using Execute Test. The button panel indicates how many total executions of the Test Cases are waiting. In the first Test Case, click Start Test Case to begin executing the test.

The main difference between executing regular Test Cases and parameterized Test Cases is when there are multiple Iterations. You have to complete all the iterations before the Test Run execution process moves on to the next Test Case. And of course, you perform your testing with the variations specified in the Iterations. You do not have to step through the Iterations in the order of appearance. If you want to perform Iteration 3 before Iteration 1, you are not prevented from doing so.

Also, if you have permission to MODIFY Test Runs, it is possible to add more iterations while executing other Iterations.

# **Warning**

If you do this, be sure to click the (Refresh) icon in the sidebar after adding an Iteration and modifying the default values, in order to update the test record. Also, click the (Refresh) icon in the editor toolbar to refresh the view of Test Steps. Remember that you can pause the Test Case and resume testing later. On pause, the current state of the current Iteration is logged in the test record.

# **Use Round-trip for Excel**

Polarion supports off-line execution of parameterized Test Runs via export/import to Microsoft Excel. For detailed information on this feature, see Excel round-trip. There is essentially no difference when Test Cases are parameterized, except that:

- The exported Excel workbook contains sections for all Iterations of each Test Case. Testers can log results of each step in each Iteration.
- Test Parameter values are rendered in bold font in a light blue color.

• By default, Iteration names are appended to the Test Case title in the Title column, for example, "Test Valid Login - Iteration 1".

It may be useful to keep the following points in mind:

- If any Test Parameters in any Test Case in the Test Run have no value assigned, Excel round-trip export fails with a message citing which Test Cases are missing parameters.
- By default, all selected Test Cases of the Test Run that have waiting Iterations and/or no results are exported to Excel. If incompletely executed, Test Cases are included in the exported Test Run. All Iterations are exported, whether or not they have been executed.
- Excel round-trip files containing only partial results may be re-imported to Polarion, in which case only new Test Records are imported.
- A tester can retest Test Cases and iterations by changing the result in the exported Excel workbook. When a workbook with retested Test Cases is imported back into Polarion, check the option Retest previously executed Test Cases in the Import Results from Excel dialog.
  - Alternatively, the Test Run can be re-exported and the Test Cases re-executed. On re-import, the Retest previously executed Test Cases should be checked.
- The column to which the Iteration name is appended is configurable in the export template, so
  you may see it appended to some other column than Title in the exported workbook. For more
  information, see Excel Round-trip Templates and Specify Column for Test Step Iteration Numbers.

# Integrate external testing tools

# ALM QA

Polarion provides the option of using external testing tools and using Polarion as the central repository for tracking and reporting the results of testing. For example, you might use Eclipse TPTP for automated load and performance tests, and Selenium for automated user interface testing, importing results of every test execution to Polarion where the results are visible to stakeholders and history of testing is maintained. You can use Polarion in this way with any external tool that can export test results to the xUnit XML format.

To integrate xUnit test results from an external testing tool with Polarion, you configure each of your external tools to write an xUnit file to a folder accessible to Polarion, and configure Polarion to periodically check each folder for the presences of a test results file. A scheduled job named **xUnitFileImport** is provided in Polarion which checks a specified folder. If any file is found there, the test results it contains are imported to an automatically-created Test Run. The test results file is deleted from the disk so that the folder is empty, and will be empty until the external testing tool executes another round of testing and writes another results file to the folder.

For information on scheduling and configuring jobs, see Configure the Scheduler. The job may also be invoked explicitly by a user in the Monitor.

#### **Job configuration syntax**

The following listing shows the job configuration syntax. The cron expression is just an example of one you might use to specify when the job should run. For more information on cron expressions, see Scheduler cron information and examples.

```
<job cronExpression="0 * * * * ?" disabled="false" id="xUnitFileImport"</pre>
name="xUnitFileImport" scope="system">
    <path>c:\Temp\xunit</path>
    project>playground
    <idPrefix>TR</idPrefix>
    <userAccountVaultKey>xUnitFileImportUser</userAccountVaultKey>
</job>
```

<path> is the folder where Polarion should check for a test results file.

<idPrefix> is a value to be prepended to the automatically-generated ID of the Test Run which is created when the test results file is imported. The Group ID field of the Test Run is filled with the name of the imported test results file (minus extension).

#### **Note**

The Group ID field of the Test Run is filled with the name of the imported test results file (minus extension).

xUnitFileImportUser references keys in the User Account Vault file (more info follows). If no user is specified in these keys or if the <userAccountVaultKey> element is empty, then the job can only be run by a user invoking it from the Monitor, and the job runs on that user's account. It will fail if allowed to run automatically according to the cronExpression schedule because such jobs run on the system user account which does not have write access.

In order to run the job automatically, it is necessary to configure a user in the User Account Vault on who's behalf the job will run. The User Account Vault is a properties file stored in the file system of the Polarion installation at data/workspace/user-account-vault/accounts.properties. The file contains the following two keys, which are referenced by xUnitFileImportUser when that value is specified in the <userAccountVaultKey> element of the job configuration (see example job listing above):

```
xUnitFileImportUser.user=[USER ID]
             xUnitFileImportUser.password=[USER PASSWORD]
```

Where [USER ID] and [USER PASSWORD] are the ID and password, respectively, of the user on whose behalf the job should run. The specified user must have write permissions for the project specified in the project> element.

#### See also:

- xUnitFileImport Job Parameters, Job Example, and xUnit File Example.
- Executing Tests Offline Using Excel.

# Report testing status and results

Polarion provides visibility and information about the status of testing and testing results. The two main vehicles are:

- Polarion Live Dashboards (Polarion ALM and Polarion Reviewer).
- Polarion Live Report pages.

Dashboards and reports access the most current project and testing data in the repository at the time they are updated. Dashboards are updated according to the configuration of the scheduled job that updates dashboards and can be explicitly updated to show up-to-the-minute information. Report pages are updated when viewed and can be exported to PDF any time. When you use a product license that supports it, the **Quality** topic appears in the Topics section of the **NavigationQuality** Dashboards, Audits, and Metrics panel. This topic provides a number of dashboards summarizing key quality-related information and statistics.

You can create custom report Pages especially for quality assurance and testing reports. The Testing space, available in projects based on one of Polarion ALM's project templates that provide it, is a good place to create your report Pages. QA and testing project templates provide some standard reports such as *Test Case Traceability* and *Testing Dashboard*. You can modify any of the default report pages or copy and paste their content to a new page, which you modify to suit your needs.

## Test management macros

# ALM REQ QA

Polarion products with QA and testing support provide a number of macros designed to support testing activities and reporting. For example, the {import-automated-test-results} macro enables page users to launch import of test results.

You can find a complete reference to these macros, including syntax and usage examples, in the Test Management section of the Wiki Syntax Help, which is accessible when you are editing any wiki page. You can also access it at the following URL on your Polarion server:

http://yourpolarionserver.yourdomain.com/polarion/#/wiki/Doc/SyntaxHelp#HTestManagement.

#### **Note**

Beginning with version 2012 several test management macros were deprecated in favor of new features and removed from the Wiki Syntax Help. However, these macros can still be used. See Deprecated Macros: Test Management.

See also Query for Test Record Work Items.

## Report test results from Baselines

Test results and records are included in Baselines. So, for example, if a Baseline is created at the time of a release, you could go into the Baseline to see what test cases were executed in time of release. In baseline view, the Test Runs topic shows data as they were at the time the Baseline was created.

When working in a Baseline:

- The Test Run detail page is read-only.
- Test Run properties are not present.
- Test Run detail pages can be printed or exported to PDF.
- If you open any Work Item referenced by the Test Run, it is opened in baseline view provided it does not point to a particular revision of the item.

# **Managing Variants**

# **Introduction to Variant Management**

# ALM REQ QA

Polarion provides the capability to centrally manage specifications and other assets for a product line, generating product-specific variant specifications.

Many companies today have highly diverse product lines. They offer different product models or configurations with different features sets to address ever more granular market segments and differentiate their products from competitors. In any given product line there are commonalities and variations among the products that comprise the product line. A critical need in the development of a varied product line is the ability to effectively manage the commonalities and variations in the various specifications. Ideally, the specifications for such as requirements, tests, and risk assessment should be maintained in a single master specification document for each artifact type that covers the entire product line. There should be an easy way to reuse different applicable subsets of these master specifications to create specifications fully covering each product in the product line. A dedicated variant management tool is needed for all but the simplest of product lines.

Polarion VARIANTS is an add-on to Polarion ALM application lifecycle management solutions. It leverages the already robust document-centric technology, adding a special dedicated back-end server component by Pure Systems plus project templates that enable an existing Polarion ALM installation to become a true variant management system.

# **Prerequisites**

In order to use Polarion VARIANTS features with an existing Polarion installation, the following prerequisites apply:

- You must run Polarion version 2015 or later.
- A valid license for Polarion ALM, and/or Polarion Requirements, and/or Polarion QA must be present on the Polarion server.
- A valid license for the Polarion Variants Add-on must be installed on the Polarion server.
- The Polarion Variants Add-on must be obtained from the Polarion ALM web site and installed according to the included instructions.

Variant management in Polarion is centered around Polarion LiveDoc Documents and Work Items contained in Documents. It is recommended that you become thoroughly familiar with these features before you begin working with variant management. .

# **Terminology**

The following terminology is used in the Polarion VARIANTS documentation.

# **Polarion VARIANTS Glossary**

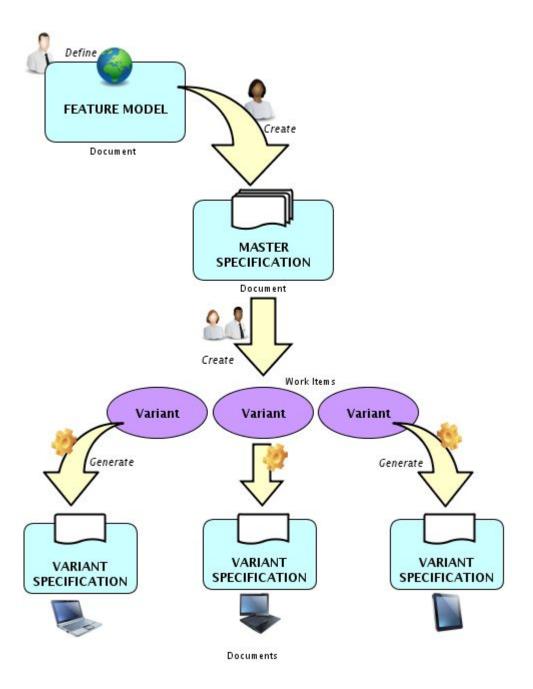
Features	Components of a product, whether they are hardware, software, or other functionality needed. When capitalized in documentation, Work Items of type "Feature".
Feature Model	Describes every possible feature of a product line. It is where a company can model their product line by decomposing all products to see what they have in common and what parts differ. Then, they assign functionality to features.
Master Specification	A Polarion Document containing all possible artifacts of one or more types for a Product Line, for example, all Requirements or all Test Cases. Each artifact is linked to one or more Features in the Feature Model. A Variant Specification contains a subset of the artifacts from a Master Specification.
	Master Specifications can be as granular as needed. For example, you can have Master Specifications for Functional, Non-functional, and System Requirements.
Product Line	Also called product family, it is the collection of all products a company produces. The different products may have some common components that must be in every product and some components that differ from product to product.
Relation	Links between Features that denote additional relationships depending on the link type. For example:
	requires: Feature1 requires another Feature to be selected, which is not already made sure of by the parent-child relationship. This is done by linking it with special link type to the required Feature.
	conflicts: Feature1 requires another feature to be not selected.
Restriction	A property of Work Items in a Master Specification that sets certain restriction(s) or limits on the item's inclusion in Variants. Specified in pvSCL (the restriction language of the Pure Systems back end), for example:  Feature1 AND Not (Feature2).
Validation	Checks by the Polarion VARIANTS back end to ensure that variants contain only valid assets and information. There are two types of Validation checks:
	Feature Model validation: Checks for errors in the construction of the Feature Model, such as inclusion of contradictory Features or failure to include a Feature from a group from which at least one selection is required.
	Restriction validation: Checks the correctness of Restrictions set in Work Item properties linking the item to one or more Features.

Variation Type	A property of a Feature that describes the Feature's inclusion in product specifications. A feature can have one of four Variation Types:
	Mandatory feature: A Feature that must be included in every product of the Product Line, or if a subfeature, must be included if its parent Feature is included.
	Optional feature: Independent Feature that can be, but does not have to be, included in variants.
	Alternative feature: Feature is one in a group, of which exactly one (and not more) must be included if the parent is included.
	Or feature: Feature is one of a group, at least one of which must be included if the parent is included.
Variant	When capitalized, a Work Item type in Polarion VARIANTS that collects Features and assets from Master Specifications to fully describe a specific product in the Product Line.
	When not capitalized, denotes one specific product in a Product Line, for example, "Model ABC is a variant of the WeatherStation product line".
Variant Description Model (VDM)	Assets, such as Requirements and Test Cases, that are mapped to Features that collectively make up a Variant
Variant Specification	A specification Document generated from a Variant and comprised of commonalities and variations from the Feature Model and Master Specification.
ļ	ļ

# **Process Overview**

There are four basic activities in a variant management project with Polarion VARIANTS:

- 1. Define the Feature Model.
- 2. Define one or more Master Specification Documents.
- 3. Define Variant-type Work Items.
- 4. Generate Variant Specification Documents.



The Feature Model is a Document containing Work Items of Feature type structured with parent-child relationships to define features and subfeatures. The Feature Model contains all the features of every product in the Product Line. It must be completed before any Master Specifications can be completed and before any Variant Specifications can be generated. Only one Feature Model is allowed per project.

A *Master Specification* is a Document containing one or more Work Item types that define various development artifacts such as Requirements, Test Cases, and Risks. A typical approach is to have different Master Specification Documents for different artifact types, for example, Requirements. With this approach, you can take advantage of Document Types, defining one for each type of artifact. You gain the possibility of defining different workflows for different Document Types, which in turn

provides the possibility of setting up different approval processes with different groups of stakeholders signing off. A Master Specification contains all the type-specific artifacts for the entire Product Line such as all the requirements or all the test cases.

A Variant is a special type of Work Item that represents a specific product in the Product Line. It is a collection of Features linked to relevant artifacts in the Master Specification(s). It is subject to validation by the Polarion VARIANTS back end to ensure integrity of the Variant Specification generated from it.

A Variant Specification is the complete specification for one type of artifact (for example, System Requirements or Acceptance Tests) for one product in the Product Line. It contains only the requirements, or test cases, or risks for the features of that specific product.

# **Variant Projects and Project Templates**

Polarion VARIANTS comes with a project template and a demo project that have predefined support for variant management. The Specification Project with Variant Management template is specifically for creating variant management projects. The Weather Station demo project is based on this template and it includes data and examples.

## **Using the Variant Project Template**

Creating a variant management project is no different from creating any other type of project. In the New Project wizard page where you select the project template, simply select one of the templates that supports variant management. The template description will indicate this.

If you need information about creating projects, see Creating Projects.

#### **Define the Feature Model**

## **Feature Model Document**

# ALM REQ QA

The Feature Model contains the name and description of every possible feature of every product in a product line. The **Feature Model** in Polarion VARIANTS is a LiveDoc Document configured to contain a single Work Item type called Feature. The Feature items in the Document are structured with parent-child relationships (shown by indent levels) that denote features and subfeatures.

If you create your project from a project template that supports variant management (recommended), the Document, and the Feature Model space where it is stored, are created automatically. One Feature Model Document is allowed per project. For projects created from other templates, some manual configuration is required. For information, refer to the Advanced Topics section of this chapter.

This section discusses several considerations around your Feature Model Document that you may want to think about before beginning to work on it.

- Document Type: Consider whether to define a Document Type for the Feature Model Document. Doing so enables you to have some configurations specific to that type, such as workflow and user permissions.
- Document Workflow: Consider whether you need an approval process for your Feature Model Document. If so, you should configure a Document Type just for the Feature Model, and

configure the Document workflow for that type to require user signatures before one or more status transitions, such as a status like Draft or Approved.

- User Permissions: Consider whether you need to control access to your Feature Model differently
  from other Documents and/or Work Items in the project. For example, you may want to restrict
  who can approve a Feature Model Document or who can create or modify Feature type Work
  Items. If so, work with your Polarion administrator to set up the controls you require.
- Work Item Type: If you know that Documents can be configured to contain more than one Work
  Item type, you may want to configure your Feature Model Document with multiple types. This is
  fine for other types of Documents, but your Feature Model should contain only one type: Feature.
  If your project is created from a template with variant management support, the Feature Model
  document is created automatically, and this type is predefined in the project configuration and set
  up in the Document presentation configuration.

#### **Create a Feature Model**

This section describes how to create a Feature Model in a variant management project. Some topics are only applicable in projects not created from a project template with variant management support and are noted as such.

#### **Create the Feature Model Space**

It is good practice to keep your Feature Model in a dedicated space. A space named "Features" is automatically set up in projects created from templates supporting variant management. You only need to create a dedicated Features space if your project is not created from one of these project templates.

You can create a new space from the Index page of any existing space. On the page toolbar, click **Create New**. In the resulting dialog box, click the **Create New Space** link.

#### **Create the Feature Model Document**

In projects created from Polarion project templates with variant management support, a Document named "Feature Model" is automatically created in the Feature Model space, and there is an icon in Navigation that opens it for editing. In projects not created from a variant management project template, you should create a new Document in your Feature Model space:

- 1. In Navigation, expand the Feature Model space, and click **Index** to open the space's Index page.
- 2. On the page tool bar, click **Create New**, and in the Create New dialog box, select LiveDoc Document. Follow the guidance in the next dialog box. Name the new Document "Feature Model", and select **Feature** as the Work Item type.

The Feature type should already be defined in your project and appear in the list of Work Item types. If it does not appear, cancel the operation and work with your administrator to create this type. (The Feature type is predefined in projects created from Polarion's variant management project templates.)

With the Feature Model Document in place, you are ready to begin constructing the feature model for your product line.

#### **Create Feature Items**

To build your feature model, create a new Work Item of the Feature type for every feature of every product in your product line. In the Document Editor toolbar, click [5] (Feature), or press Enter/Return key when you insert your cursor in an existing Work Item.

In new Feature items, the attributes you need to specify are:

- **Title:** Identifies the feature for all stakeholders. It appears in feature selection lists and restrictions.
- **Description:** (Optional) Allows you to provide additional details.
  - Be careful not to specify requirements for a Feature in its description. Requirements should be defined independently in a dedicated requirements specification Document that has a relevant Work Item type.
- **Variation Type:** Describes the Feature's inclusion in product specifications. In project templates supporting variant management, this is a custom field of the Work Item of the Feature type, of field-type Enum, configured with the options as described in the Terminology section. For information, see Terminology: Variation Type.

#### Tip

The Status field is one of the standard fields of all Work Items of all types, and the Work Item Properties shows it by default in all Documents. However, the value of the Status field in Work Items of the Feature type has no impact on any aspect of variant management. You can safely ignore it when defining Feature items.

#### **Set the Variation Type:**

You specify the Variation Type for a Feature in the Work Item Properties sidebar. If that sidebar is not open, place your insertion cursor anywhere inside any Feature item in the Document, then drop down the **Show Sidebar** menu on the Document Editor toolbar, and select **Work Item Properties**.

If the **Variation Type** field does not appear in the sidebar, take the following steps to show it:

- 1. On the Work Item Properties sidebar header, click (Pane Settings), and choose **Select fields**.
- 2. In the **Select Fields** dialog box, make sure the **Variation Type** field appears in the list on the right. If it doesn't, locate it in the list of fields on the left, select it, and click Add.
- 3. In the list on the right, select the Variation Type field, select show in sidebar only or show in Document and Sidebar,, and then click Apply.

The sidebar configuration is saved per user and the setting persists if you navigate away from the Document, and between logons.

#### **Define the Feature Model structure**

Product-line feature modeling is a discipline in itself, and teaching it is beyond the scope of this help. The Feature Model Document is a tool that enables variant managers to capture the product line features and their relationship to one another. Then, specifications for variant products can be generated, which use a subset of Feature Model features. The main relationship that needs

captured is a parent-child relationship to show that a feature is a subfeature of some other feature. For example, in a car, a parent feature might be Headlamp, which has child features such as Halogen Headlamp or Mercury Headlamp.

You can create the parent-child structure of Feature items in a Feature Model Document by indenting the items. You can decide when to create the structure. You may choose to do it as you create new Feature items, or after all Feature items have been created, or a combination of these approaches. You should undertake a comprehensive review of the Feature Model before beginning to create Variants.

# Tip

An administrator can configure the Document workflow for your Feature Model Document to support a review and sign-off process.

Development of the Feature Model can be a collaborative effort, with different people working to specify different feature areas. For example, for a car product line, different people might specify the features for the braking system, the drive train, and the driver electronics.

#### Link features for validation:

When designing your Feature Model, you can create links between different features with special link roles that define dependency or exclusion between the linked features. Two link roles are predefined in Polarion's variant management project templates, which can be used for this purpose:

- requires: Denotes that, for example, Feature A requires Feature F. There is an opposite role, is required by that can be applied if linking from the other direction: specifying that Feature F is required by Feature A.
- conflicts with: Denotes that one feature cannot coexist in the same specification as another. For example, a feature such as Halogen Headlamp cannot coexist with a feature such as Mercury Headlamp. You can link these features with this link role.

# **Note**

Currently, these are the only link roles supported by the Polarion VARIANTS back-end server in its feature validation process.

You can use these link roles in conjunction with Variation Type to provide more parameters for the back-end process that validates features included in Variant Specifications. Variation Type provides a first layer. For example, two features in a group might have the Variation Type "Alternative feature", which allows only one to be included if the parent is included. These features conflict because of Variation Type, so there is no need to link them with the *conflicts with* role. It would, however, make sense to link features in different groups this way. For example, suppose two features like "Sensor North America" and "Sensor Europe" should never be used in the same variant specification. Linking them as conflicting with each other ensures that the validation process raises an error if the two are included in the same Variant.

For information on how to link items, see Linking Work Items. The section Using Quick-link outlines the approach recommended when working with Documents.

# **Create a Master Specification**

## **Create a Master Specification overview**

In a variant management project, a Master Specification is a LiveDoc Document that contains all the specifications for all the products in a product line. For example, a Master Specification might contain all the requirements, test cases, and risks for a product line. For a large product line, you can have multiple Master Specifications. For example:

- Master Requirements Specification: All the requirements for all the products in the product line.
- Master Test Specification: All the test cases for all the products in the product line.

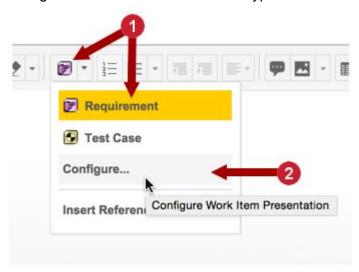
The same goes for other artifact types such as risks and regulations. You can be even more specific with a given artifact type. You might have Master Specifications for different kinds of requirements like functional, nonfunctional, software, electrical, and mechanical. The main thing to remember is that each Master Specification Document should contain all the artifacts of the given type for the entire product line.

You can take advantage of all the features of Polarion LiveDocs in your Master Specifications. For example, you could configure a project to have a different Document type for each type of Master Specification, and then configure appropriate workflows for each type, including an approval process which includes different stakeholders for different Document types. You can also configure more specific user permissions applicable to different Document types.

Polarion's project templates for variant management come preconfigured with a space named "Master Specifications", which contains two Master Specification Documents: "System Requirements Specification" and "Test Specification".

#### **Create the Work Items**

Each Master Specification Document should be configured to contain the desired Work Item type or types, depending on the approach you decide upon. For example, the Document might be configured to contain only Requirements, or both Requirements and Test Cases. An example of a Document configured to contain two Work Item types follows.

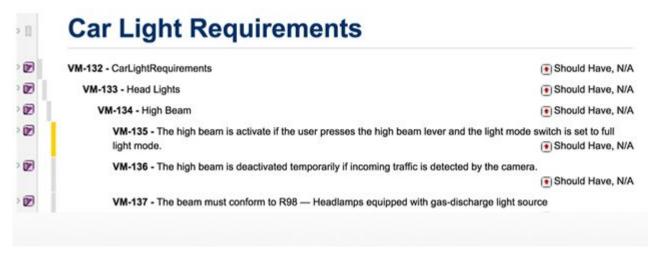


- Create new Work Item
- Configure Work Item types in Document

You can create new Work Items in your Master Specification using the menu on the toolbar of the Document Editor or by pressing your Enter/Return key when your cursor is inside an existing Work Item.

#### Structure the Work Items

You can easily create a hierarchical structure of parent-child relationships between Work Items in your Master Specification by indenting them to different levels. This is a common practice with requirements. The following figure shows an example of a Master Requirements Specification using indentation to create parent-child relationships.



Such structuring automatically creates links in the Work Items with a link role of **has parent** or **is parent of**. These link roles can subsequently be queried, for example, for reports or dashboards.

#### Tip

It is recommended that you complete and review the structure of the Work Items in your Master Specifications before creating Variants that draw items from them.

## **Define restrictions**

Restriction is an attribute of Work Items in a Master Specification. It is what ties any given item to Features in the Feature Model so that variant specifications can be generated from the Master Specification, which contain only those Work Items relevant to the specification for a specific product. Restrictions are specified using the pvSCL language, the language of the Polarion VARIANTS Server Add-on. Reference documentation for the language, developed by Pure Systems and widely used in their variant management systems, is provided in the distribution of the extension

If no Restriction is specified for a Work Item in Master Specification, then it is included in all generated Variant specifications. So you specify restrictions to indicate that, for example, some requirement is

included in Variant specifications only if the restriction expression evaluates to TRUE. Otherwise, if the restriction is evaluated to FALSE, the item isn't included.

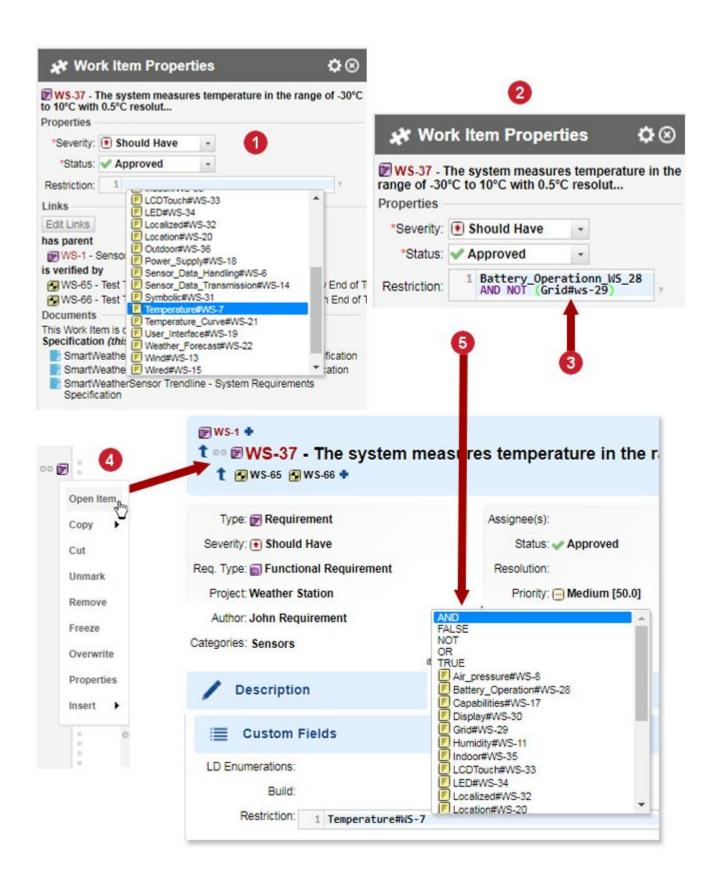
A special editor for specifying Restrictions in pvSCL is implemented in Polarion VARIANTS and appears in the Restriction field of Work Items in Master Specification Documents included in variant management project templates. The Restriction Editor can turn Full Screen mode on and off using the F11 key. It has an auto-completion feature, invoked via the keyboard shortcut CTRL + Space. You can access the Restriction Editor in the Document Editor (see the following section), or you can open Work Items in the Work Item Tracker and find it in the Custom Fields section of the page. The pvSCL editor is case insensitive.

### Set up the sidebar

If you want to use Restriction Editor in a Document, you need to open the Work Item Properties sidebar when focused on a Work Item in a Master Specification Document. The editor is not shown by default, so you need to perform the following steps to show it:

- 1. Place the insertion cursor inside any Work Item in the Document, and open the Work Item Properties sidebar.
- 2. On the Work Item Properties sidebar header, click (Pane Settings), and choose **Select fields**.
- In the **Select Fields** dialog box, make sure the **Restriction** field appears in the list on the right. If it doesn't, locate it in the list of fields on the left, select it, and click Add.
- 4. In the list on the right, select the Restriction field and select show in Sidebar only, and then click Apply.

The sidebar configuration is saved per user, and the setting persists if you navigate away from the Document and between logons.



- Restriction Editor in a Document.
- Example Restriction expression.
- # is used to separate a feature, title and ID.
- Restriction Editor in the Work Item tracker.
- Ctrl + Space for auto-complete.

**F11** for full-screen mode.

**Esc** to exit full-screen mode.

## Specify a Restriction

## Warning

Your Feature Model should be complete before you begin specifying Restrictions in the Work Items in a Master Specification.

The Restriction Editor has built-in completion assistance. The Ctrl + Space shortcut opens a list box containing the Features in your project's Feature Model, and logical and Boolean operators. You can add any item in the list to the Restriction expression in the editor field by double-clicking the item in the list (see the previous figure).

The following steps assume you have a Master Specification Document open, and the Work Item Properties sidebar configured to show the Restriction Editor, as described earlier in this section. You may find it useful to have the Feature Model Document open in another browser tab or window.

#### Specify a Restriction:

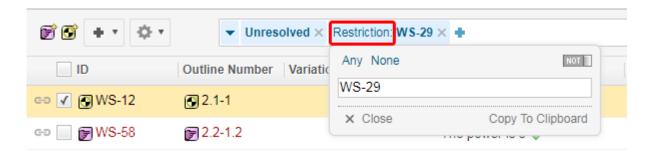
- 1. In the Document body, place the insertion cursor anywhere inside the Work Item for which you want to specify Restriction.
- 2. In the Work Item Properties sidebar, click inside the Restriction field.
- 3. Press Control + Space, and in the list of Features, expressions, and operators, double-click the item you want to add to the Restriction.
- 4. Repeat the above to add expressions, operators, and Features as needed to construct a complete Restriction expression in pvSCL.

If you know pvSCL, you can type the complete Restriction expression in the Restriction Editor, or build it in a text editor and paste the finished expression into the editor.

You need to fully specify all Restrictions in all Master Specifications before you can create Variants and generate product-specific specifications.

#### Tip

You can search for the Requirement using a Restriction query followed by the Feature ID.



### pvSCL expression language

The pure::variants expression language pvSCL is a simple language to express constraints, restrictions, and calculations. It provides logical and relational operators to build simple but also complex boolean expressions. Polarion VARIANTS uses a subset of the pvSCL language to link Features with Work Items and to express relations between Features.

#### **Boolean Values:**

Expressions can resolve to a Boolean value, that is, TRUE or FALSE. An expression is said to fail if its Boolean value is FALSE, and to succeed otherwise.

Syntax: TRUE or FALSE

Example: NOT (TRUE = FALSE)

#### **Logical Combinations:**

Expressions can be logically combined. For this purpose the expressions are evaluated to their Boolean values. It is an error if this conversion is not possible. The logical operator is then applied to the Boolean values resulting in TRUE or FALSE. The simplest restriction is the name of a feature, but it is also possible to build restrictions out of a combination of logical operators and multiple feature names.

The following logical operators are supported:

- AND Binary operator that yields TRUE if both operands are TRUE.
- **OR** Binary operator that yields TRUE if at least one operand is TRUE. If the first operand is TRUE, then the second operand is not evaluated.
- NOT Unary operator that yields TRUE if the operand is FALSE.

#### Syntax:

- expr AND expr
- expr OR expr
- NOT (expr)

**Example:** FeatureA AND NOT (FeatureB OR FeatureC)

# **Create Variants and Variant Specifications**

#### What is a Variant?

Before undertaking operations described in this section, you should complete the Feature Model and Master Specification(s) for your product line.

#### What is a Variant?

Variant is a special Work Item type that collects specification items for the feature set of a single product in a product line. This type is predefined in Polarion project templates with variant management support.

#### Create a Variant

It is possible to create Variant items in a Document configured to contain that type or in the Work Item Tracker. The latter approach is recommended.

Create a Variant in the Work Item Tracker:

- 1. Open the project that contains the Variant work item type.
- 2. In Navigation, expand the Work Items topic, and select Variant. The Work Item Tracker opens with only Variant type items displayed.
- 3. On the page toolbar, click **Create** and select **Variant**. A new Variant item opens in the Work Item Editor.
- 4. Enter a title for the new Variant. The title should identify the product to which it pertains, for example, "Useful Widget Professional Model".
- 5. (Optional) Enter a description for the Variant.
- 6. Click **Create** to save the new Variant.

After creating a new Variant item, the next step is to select the features of the product it represents.

#### **Select Features**

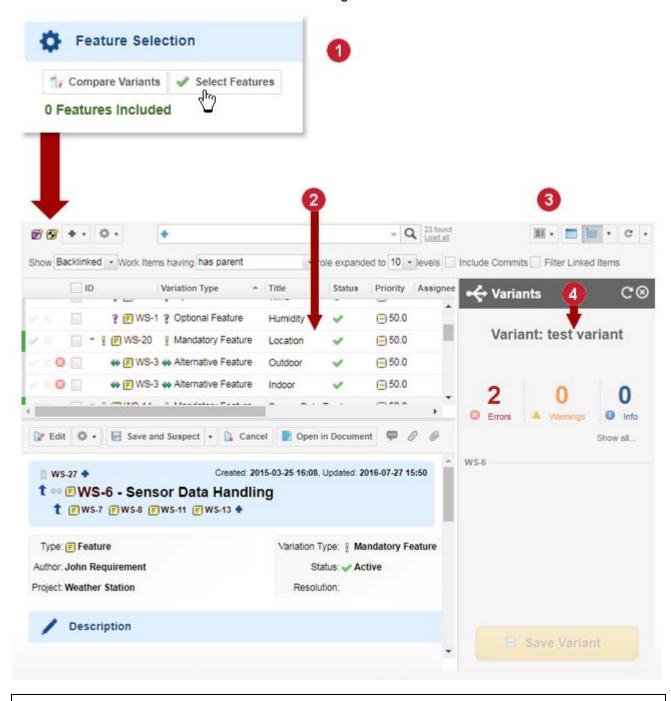
To select Features, you must be using a Polarion Variants license, and you must have the necessary permissions: MODIFY permission for the Work Item, and READ and MODIFY permissions for the Feature Selection field.

The Work Item Editor (lower part of the page) contains a section named Feature Selection. If you have the required license and permissions, the **Select Features** button appears in this section. Clicking this button opens a new browser tab or window with a new instance of the Tracker, which displays a tree-table of all the Feature items in the project Feature Model. The tree is automatically sorted by Variation Type, but administrators can set it to sort by Outline Number by default instead with a simple addition to the polarion properties file. (See the following paragraph.) The tree is constructed on the parent-child hierarchy of Feature items. The order for variation type is **Mandatory**, Optional, Alternative, and finally OR.

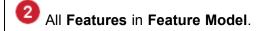
In this instance of the table, you cannot change the sorting via the user interface. But you can enter a query, and you can add/remove columns. It is, however, possible to have the table sorted by "Outline Number" so that features are presented as they appear in most Feature Model Documents. A Polarion administrator can add the following line to the system configuration file, polarion.properties:

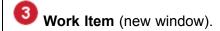
com.polarion.variants.featureSelection.sortByOutlineNumber=true

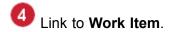
The Polarion server must be restarted for the change to take effect.



Work Item Editor

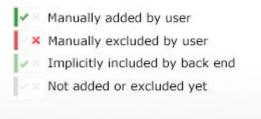






#### Note

Some items can be already shown as selected. This is because the validation process on the back-end server has implicitly selected them according to Variation Types and Restrictions.



The user interface indicates when features are added by the back end server.

The back end adds all required items up to root and Mandatory items below a selected item are added, if they are not otherwise selected.

Select Features for a Variant:

- 1. Open Variants in the Work Item Tracker (Navigation \*\* Work Items \* \* Variant).
- 2. Select the desired \(\forall \) Variant item in the tree-table listing of Variant type Work Items.
- In the Feature Selection section of the Work Item Editor, click 

  ✓ Select Features.

A new browser tab or window opens with a new instance of the tracker, containing a tree-table of all Feature items in the project Feature Model and a sidebar labeled with the name of the Variant item.

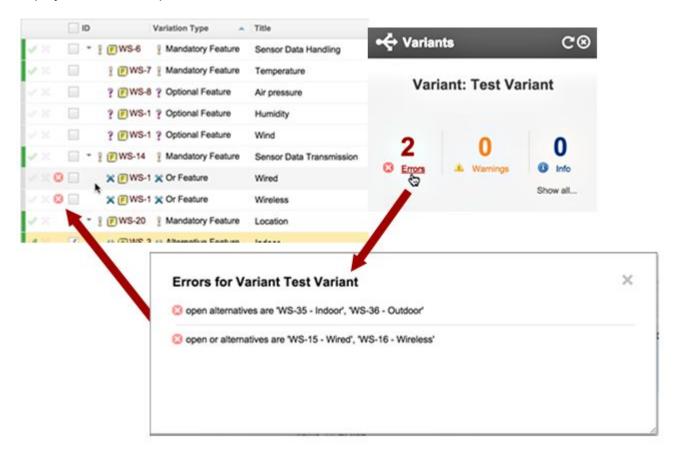
4. Explicitly include Features in the Variant by clicking the green check mark icon on the Feature's row in the tree-table. Explicitly exclude Features by clicking the red X icon. If you change your mind, click again to deselect.

When you include some Features, other Features may be implicitly selected, denoted by a pale green color in the check mark icon on the respective Feature rows. This happens because of the Variation Type set on the different Features.

5. Save the Variant by clicking 🖥 Save Variant in the sidebar. If there are many Feature to be selected, it is advisable to save periodically during the selection process.

#### Validate Features

The **Polarion Variants** Server Add-on provides back-end validation of the feature selection of the Variant each time you save it (or open it). If there are any errors or warnings, then the sidebar displays a count, and provides links to the details.



The **Polarion Variants** back-end server checks the validity of a Variant and issues errors, warnings, and information.

Clicking on the respective **Errors** or **Warnings** link in the sidebar displays a dialog listing the problems. The dialog is modeless, so you can keep it open as you work on the feature selection to correct problems. The dialog is updated each time you save the Variant. Errors result from Variation Type and link roles set on Feature items in the Feature model. For example, a Feature may have subfeatures with Variation Type "Or feature", which require that at least one must be selected, but none is selected. Selecting one and saving resolves the error.

#### Warning

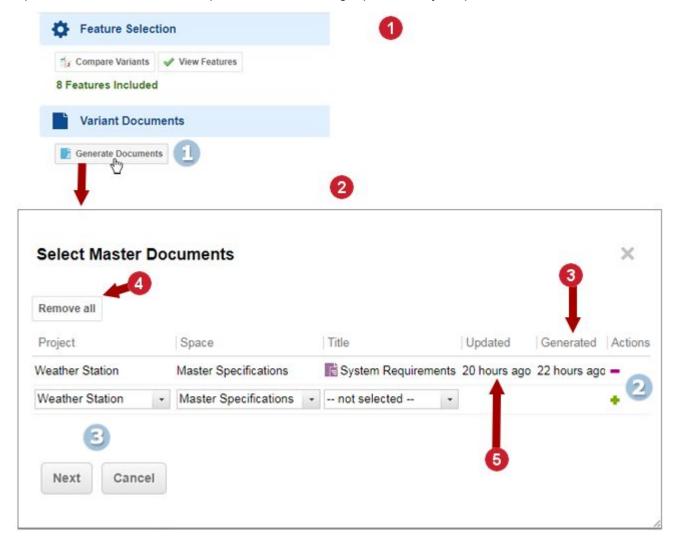
All errors must be resolved before any specification Documents can be generated from the Variant.

#### Set Variant status

Like all Work Item types, the Variant type has a default workflow with defined statuses and actions that transition to another status. And like all types, the workflow is customizable to conform to any process you use. Among other possibilities, you can integrate a review and approval process for Variant items to ensure accuracy and completeness before generating specification Documents from a Variant.

#### **Generate Variant Documents**

The final step in the process is to generate one or more specification Documents based on your Variant. These Variant Specifications (sometimes referred to as "100% specifications") contain all the specifications for one product in a product line. How many Variant Specification Documents you generate depends on how many Master Specification Documents you include when you launch the Document generation process. Generally, you would include all your Master Specifications. For example, if you have one Master Specification for Requirements and another for Test Cases, you might want to include both when you generate Documents from a Variant, resulting in a Requirements Specification and Test Case Specification for a single product in your product line.



- Work Item Editor screen.
- Include all Master Specifications.
- Last generation of the variant specification for the selected master document.
- 4 Remove all preselected documents.
- 5 Last update of the selected master document.

#### Note

It is possible to generate from a branched Master Specification Document. For example, some revision of the Master Specification may be approved for production of version 1.0, but work needs to continue on it for version 2.0, while version 1.0 is in production. By creating a branched Document from the approved version 1.0 revision, Variant Specifications for version 1.0 can be generated from the version 1.0 Master Specification, that is, the branched Document.

## **Generate Variant Specification Documents:**

- 1. Open Variants in the Work Item Tracker (Navigation Work Items Variant).
- 2. Select the desired Variant item in the tree-table listing of Variant type Work Items.
- 3. In the **Variant Documents** section of the Work Item Editor, click **Generate Documents**.
- 4. In the **Generate Documents** dialog box, select one or more Master Specification documents. First select the space where the Document is stored, and then select the specific Document.

Repeat this step to include any additional Master Specification Documents, adding new rows to the table in the dialog box, if necessary, until you have listed all the relevant Master Specification Documents.

When you have added all relevant Documents, click **Next**.

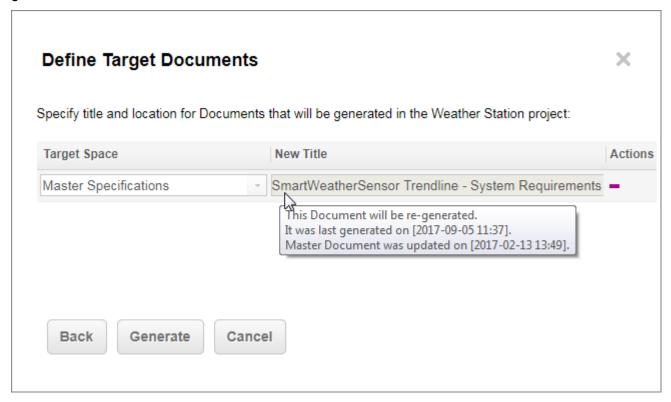
- 5. In the next frame of the dialog box, specify the space where the generated Variant Specification Document(s) should be stored. If there are multiple Master Documents, do this for each one. The space suggested by default is that of each Master Document. It is generally good practice to keep Variant Documents in a different space.
- 6. (Optional) Specify a different title for each Document, if you prefer it to the default suggested title.
- 7. Click **Generate** to launch the system job that generates the Variant Documents.

The dialog box provides a link to the **Monitor** topic, where you can check the status of the job and access the job log. The Monitor opens in a new browser tab or window. If the job should fail, the log

contains detailed information that can help you, your project administrator, or a technical support engineer diagnose and resolve the problem.

#### **Regenerate Variant Specifications:**

- 1. Select **Variant** to regenerate.
- 2. Click Generate Documents as you did when you generated them the first time.
- 3. Notice that the previously generated documents are automatically preselected in the **Select Master Documents** dialog box, and all you have to do is click **Next**.
- 4. The Target Space and New Title fields are read-only and if you hover over the New Title field, a tooltip appears showing the revision date of the Master Document that the previous version was generated from and the Master document's Head revision date.



- 5. Click Generate.
- 6. A **Generating Documents** dialog box appears and provides a link to the **Monitor** to check the progress of the generation. Click **OK** to close the dialog box.

## **Compare Variants**

Polarion VARIANTS provides a quick easy way to compare your Variants. A special matrix view shows which Features are used in which Variants.

- 1. Open Variants in the Work Item Tracker (Navigation → W Work Items → ヤ Variant).
- 2. Select any Variant item in the table of Work Items of the Variant type.

3. In the Feature Selection section of the Work Item Editor, click Compare Variants.

The Variants Matrix opens in a new browser tab or window. The structured Feature Model is rendered as rows and Variant items as columns. You can filter the display using the special instance of the visual Query Builder at the top of the page. For example, you might query for Features with "Obsolete" status and Variants with "Active" status. This would reveal obsolete features in currently active Variants.

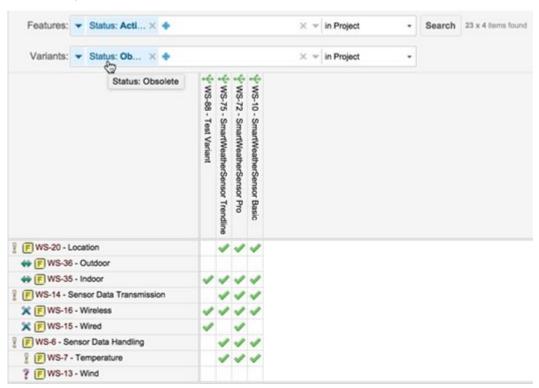
#### **Note**

# **Sorting the Matrix:**

By default the matrix is sorted by **Variation Type**. Administrators can change this to **Outline Number**, the way it usually appears in Feature Model documents, by adding the line below to the **polarion.properties** file.

com.polarion.variants.featureSelection.sortByOutlineNumber=true

There are two properties in the polarion.properties file that define the maximum number of cells in a Traceability matrix or a Variants comparison matrix.



#### **Review Variant history**

Variants are Work Items, and like all Work Items a change history is automatically recorded and maintained. Access the History just as you would for any Work Item, for example, the **History** button in the Work Item Editor.

One particular feature of Variant history is that Feature selection is included. For any revision of a Variant item, you can see if Features were added or removed, and if so, how many, when, and by whom.

In History and Baselines, the View Features button replaces the Select Features button. This is also the case if feature selection is read-only, for example, the user does not have permissions needed to select Variant features.

# **Working with Variant Documents**

This section contains information that may be useful when viewing a generated Variant Document.

# **Identify the Variant Work Item**

When viewing a generated Variant Document, you may want to know the Work Item from which it was generated, and access that item. You can find this by opening the Document Properties sidebar when viewing the Variant Document. The Variant label displays the title of the source Work Item. The title is a link that opens the Variant Work Item in a new browser tab or window.

# **Identify the Master specification(s)**

When viewing a generated Variant Document, you may want to know the Master Specification Document from which it was generated, and access that Document. You can find this by opening the Document Properties sidebar when viewing the Variant Document. The Branched from label displays the title and revision number of each Master Document used to generate the current Variant Document. The title is a link that opens the indicated revision of the Master Document in a new browser tab or window.

#### Search for generated Variant Documents

You can query on variant.id or variant.title to search for all Documents that were generated from a specific Variant item.

# Variant management advanced topics

#### **Configure Variants for existing Projects**

This section describes how to configure existing projects, created with a Polarion version prior to 2015 or templates other than the Specification Project with Variant Management Template or the Weather Station demo project.

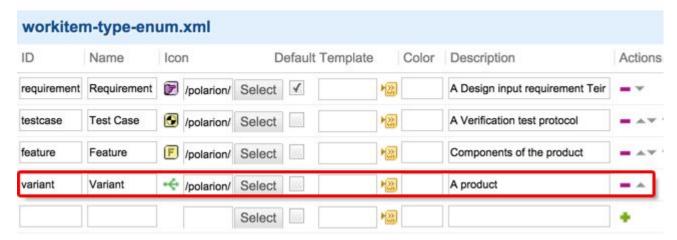
#### System requirements

- You must have Polarion 15.0 (2015) or later.
- A valid Polarion VARIANTS Add-on license installed on the Polarion server.
- The Polarion VARIANTS Server Add-on must be obtained from the Polarion Software web site and installed according to the included instructions.

# Configure the "Variant" Work Item type

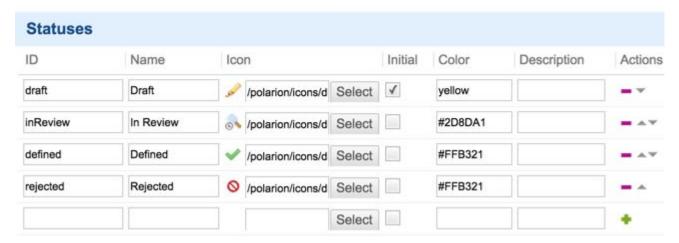
Enter ►Administration in the target Project.

- 2. Click **\*\* Work Items** → Types.
- 3. Enter variant as the ID and Name, click Select, and choose or upload an icon.



You can change variant as the required ID value by editing com.polarion.variants.variantType in the polarion.properties file.

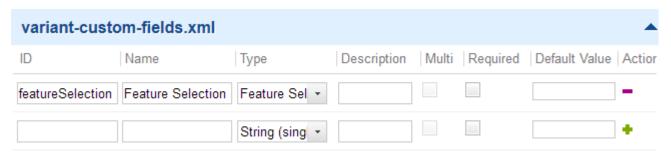
- 4. Enter a **Description**, click +, and then **Save**.
- 5. Now you have to setup a workflow for the newly created type. (Project Administration Work Items Workflow)
- 6. Click **Create** to the right of the newly created **Variant** Work Item to get started. See Configure a Workflow for more details on how to configure a Work Item workflow.
- 7. A recommended **Status** enumeration can be fairly simple, for example, something like *Draft, In Review, Defined and Rejected.*



- 8. Click +, and then Save.
- 9. Define the following custom fields for the new Type. (Project ✔Administration → ✔Work Items → Custom Fields)

Add a Feature Selection as a Custom Field for the newly created Variant Work Item with featureSelection as the ID, Feature Selection as the Name, and select Feature Selection as the **Type** from the drop-down menu.

See Configure Custom Fields for more details on setting up a Work Item custom field.



10. Update the Work Item form layout. Project Administration Work Items Form Configuration)

Add Variant Documents and Feature Selection as separate sections. Exclude the Feature Selection field form the custom fields panel:

```
<field id="description"/>
<field id="featureSelection"/>
<field id="variantDocuments"/>
<!-- Description of panel is mandatory. -->
<panel description="Custom Fields">
<field id="@allCustomFields,-featureSelection"/>
</panel>
```

See Work Item form layout for more details if you are unsure about how to do this.

## Tip

You can also create a new project using the Specification Project with Variant Management template and reuse the configuration files by copying them to your existing projects:

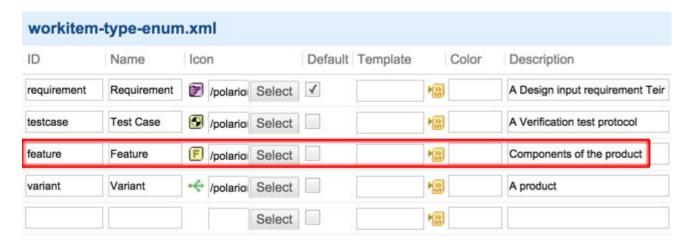
- .polarion/tracker/fields/variant-custom-fields.xml
- .polarion/tracker/workflow/variant-workflow.xml
- .polarion/tracker/fields/variant-status-enum.xml
- .polarion/hats/ default/tracker/variant-form-layout.xml

## 2. Configure the Feature Work Item type

- 1. Enter **Administration** in the target Project.
- Click ₩ Work Items → Types.

3. Enter **feature** as the **ID** and **Name**, click **Select**, and choose or upload an icon.

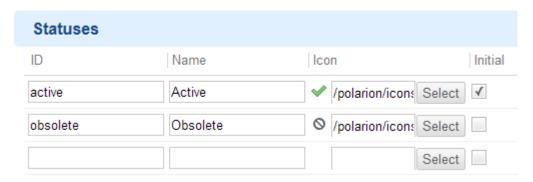
Feature can be capitalized in the name but must be lowercase for the **ID**.



You can change **feature** as the required **ID** value by editing <code>com.polarion.variants.featureType</code> in the polarion.properties file.

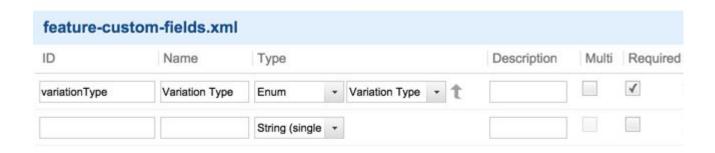
- 4. Click •, and then Save.
- 5. Now you have to setup a workflow for the newly created Feature type. (Project Administration Work Items Workflow)
- 6. Click **Create** to the right of the newly created **Feature** Work Item to get started.

  See Configure a Workflow for more details on how to configure a Work Item workflow.
- 7. A recommended **Status** enumeration can be something like: *Active* and *Obsolete*.



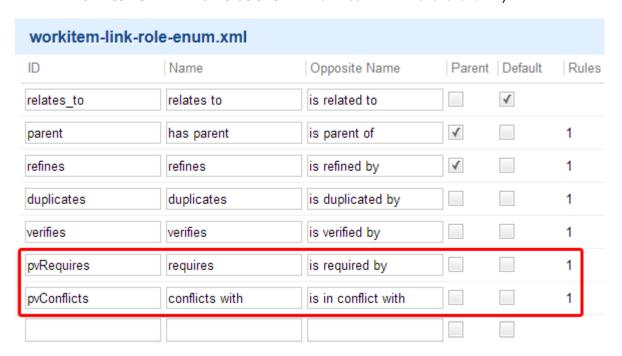
- 8. Click •, and then Save.
- 9. Define the following custom field for the new Feature Type. (Project ✓ Administration → Work Items → Custom Fields)

(variationType as the ID, Variation Type as the Name, Enum as the Type, and select Variation Type from the drop-down menu.)



See Configure Custom Fields for more details on setting up a Work Item custom field.

10. Configure additional link roles between the Feature type Work Items. (Project Administration → Work Items → Enumerations → workitem-link-role-enum.xml)



See Configure Enumerations for more details.

11. Define a link rule for both requires and conflicts with roles so that the links are only used between two Features.

(Project **Administration** → Work Items → Enumerations workitem-link-role-enum.xml)

See Link Roles and Rules for more details.

## Tip

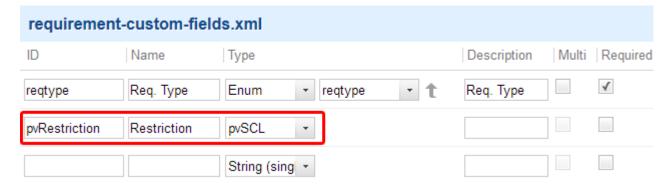
You can also create a new project using the **Specification Project with Variant Management** template and reuse the configuration files by copying them to your existing projects:

- .polarion/tracker/fields/feature-custom-fields.xml
- .polarion/tracker/workflow/feature-workflow.xml
- .polarion/tracker/fields/feature-status-enum.xml

## 3. Configure the Specification Work Item types

Specification Work Item types can be any Work Item type that will be part of your Product Line and will be included in Master Specification Documents, for example, Requirement, Test Case, or SW Requirement.

1. Add a custom field restriction to each Work Item type that will be used as a specification, such as **pvRestriction** as the ID, **Restriction** as the Name, and **pvSCL** as the Type.



## **Cross-project Variant management**

You can manage Variants across multiple projects, and select multiple Master Specification projects that are separate from the Feature Model.

#### Note

Generated Variant Specification documents must reside in the same project as the Variant Work Item.

There are two different methods for configuration cross-project variants. It depends on whether you want to use the same configuration for all Work Items within a Project (the default), or you want to use Work Item Custom Fields to use different configurations for all Variants.

The legacy method of creating a definitionProject custom field is still supported, but not recommended.

## Use the same configuration for all Work Items within a Project

Configure via the Variants section in Administration on the project level.

#### **Note**

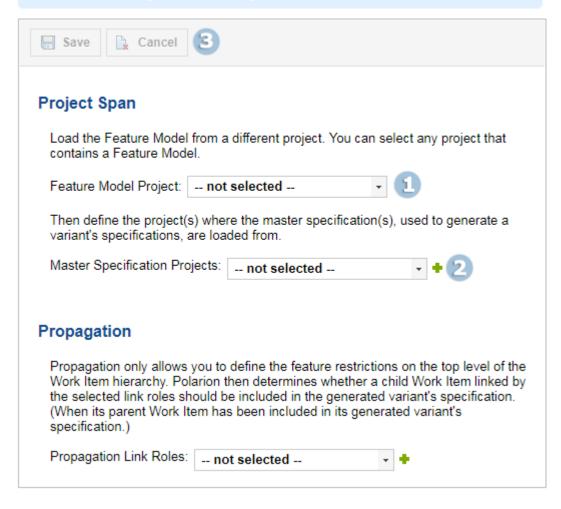
## All configurations are optional:

All the configuration fields in Variants Administration are optional.

Leaving them as **-not selected-** will simply default to the current **Project**.

- 1. Click on the top left corner.
- 2 Click Administration.
- 3. Click Variants.

# Variants Management Configuration



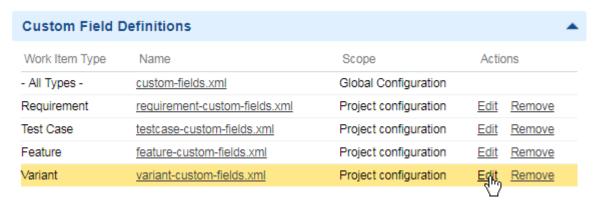
4. Select the Feature Model Project.

- Select a Master Specification Project and click ♣ .
   Add as many Projects as needed.
- 6. Click Save or Cancel to cancel changes.

See the Propagation section for details on how to configure Variant Management so that the selection of Variant Work Items is determined by the relations to their parents and their parents' restrictions.

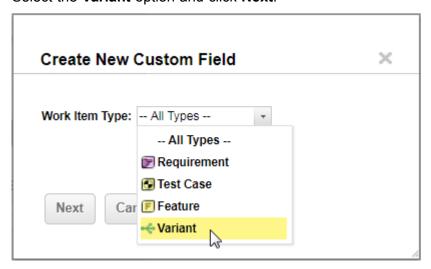
## Use different configurations for all Variants via Work Item custom fields

- 1. Click on the top left corner.
- Click Administration.
- Click Work Item and select Custom Fields.
- 4. Click Edit to edit the existing variant-custom-fields.xml file.

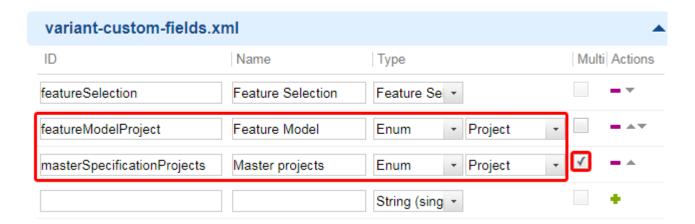


A **Variant** custom field should already exist in a Project created from a template that supports Variants, but if it wasn't, create one as follows:

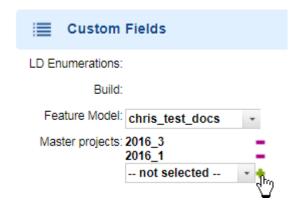
- a. Click Create new configuration.
- b. Select the Variant option and click Next.



- Add (♣) the following IDs:
  - featureModelProject
  - masterSpecificationProjects



- 6. Click **Save** on the top.
- 7. Exit Administration, expand the \*Work Item , select \*Variant and select a Variant from the list.
- 8. The Custom Fields section of the Variant now includes options to add a Feature Model and **Master Projects** that only apply to the selected Variant.



## **Legacy Cross-project Configuration**

The legacy configuration via the definitionProject Custom Field that defines what project both the Feature Model and Master Specifications should be loaded from is still supported, but we recommended using one of the two options above.

## **Comparing Cross-project Variants**

The comparison feature works the same as for single-project variant management. Open a Variant that has the Definition Project field set to a different project, and click Compare Variants in the

**Feature Selection** section of the Work Item Editor. Compare Variants shows all Variants that have same project selected in the Definition Project field.

## **Propagation**

If configured, then the selection of Variant Work Items is determined by the relations to their parents and the restrictions of these parents. The restrictions can only be managed on the highest level of the hierarchy, not on all levels manually. Therefore:

- A child item with no restriction that is linked to a parent item is included in the generated variant specification if its parent item is included. Its restriction is evaluated as TRUE.
- A child item with a restriction that is linked to a parent item:
  - o Is included in the generated variant specification if both the child and parent's restrictions are evaluated as TRUE.
  - o Produces a validation error if its own restriction and its parent's restrictions are evaluated differently.

## Minimize exceptions when migrating existing data sets

When using the propagation of Work Items into a variant selection on existing data sets, the complexity of the migration can be substantial. By default the variant evaluation reports an error if propagation and restrictions contradict each other because the data is not consistent.

To simplify the adoption of propagation, it's possible to *allow* this inconsistency by setting the following property to true in the polarion.properties file.

(The default setting is false.)

com.siemens.polarion.purevariants.propagationDeterminedByRestriction=true

When set to true, propagation is determined by restrictions. If there is a conflict between propagation and restrictions, then the restriction is honored instead of throwing an evaluation error.

#### Warning

This behavior also has a consequence: If a child Work Item in the document structure is included into the variant selection via propagation, then its parent Work Item from the Document structure is also included, even if its restriction does not allow it.

Normally this situation would trigger a validation error, but with com.siemens.polarion.purevariants.propagationDeterminedByRestriction set to true, it will be allowed and not reported to the user.

#### Tip

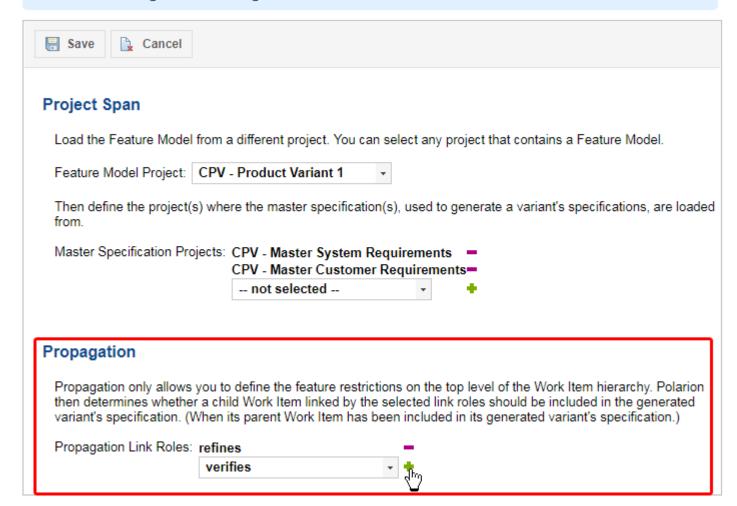
Only use this property during the early adoption of propagation and remove it as soon as your data is in a consistent state.

## **Configure Propagation**

Configured via the Variants section in Administration on the project level.

- 1. Click on the top left corner.
- 2. Click Administration.
- 3. Click **Variants**.

## Variants Management Configuration



#### Note

## The Propagation of Referenced Work Items:

Under **Propagation**, when selecting the Work Item link roles to include, you can also generate from documents that contain referenced Work Items. If you do, then the configuration is taken from the Project that contains them.

#### **Back-end server administration**

The **Variants** topic of global and project Administration enables administrators to restart the Polarion VARIANTS back-end server and clean up the data. The **Initialize Server** button launches the operation.

This action should rarely be needed. It should only be invoked to recover from unexpected VARIANTS Server data corruption, for example, caused by a power outage.

# **Appendix**

## **Using Classic Wiki**

## **Overview - Using classic Wiki Pages**

### Warning

Beginning with Polarion version 2015, this feature is replaced by Pages. You should create all new reports and informational pages using that feature, rather than Classic Wiki (formerly called just "Wiki"). Classic Wiki will be supported through the version 2015 release cycle, and possibly longer. If you have any reports or informational pages that you anticipate needing in the long term, you should plan to recreate these using LiveReport or Info Pages.

Polarion features a built-in Wiki where users can create whatever content they need to support their projects and collaboration. There is a default wiki space for the repository, and a default space for each project, created when the project is created. In projects, you can create additional spaces to help organize your content. For example, you might have one space for guidelines, and another for meeting notes. The wiki provides a highly flexible communication medium accessible to everyone with access permissions for Wiki pages. Wiki pages are stored and versioned in the underlying Subversion repository just like any other development artifact.

The Wiki is fully integrated with other features of the platform, so it is quite easy to pull information from other areas such as the Work Items tracker and display it in useful ways in Wiki pages. A robust library of macros is provided that makes it simple to embed complex information. A complete reference for these and as well as Wiki mark-up syntax is provided online in the Wiki Editor.

This chapter explains the major features of the Wiki. Most interaction is fairly standard. If you have used other Wikis you should have no difficulty working with the integrated Wiki, especially if you use the rich-text editor which is available for working with non-scripted Wiki content.

The entry point of each Wiki space is its Home page, which appears in Navigation under the space name the **Documents and Pages** topic. Within each Wiki space you can add Wiki pages, and/or create additional Wiki spaces, and Wiki pages within those spaces. The Help topics in this section explain how. As you develop pages, you will probably embed links to other pages. In any page, you can find what other pages are linked to the one you are viewing by clicking the **Backlinks** button on the toolbar (click **Expand Tools**) to expose the **Backlinks** section of the page. This section lists all pages linked to the current page and provides links to each one.

## Page types

There are 3 basic types of Wiki pages in Polarion:

- Content pages
- 2. Portal pages
- 3. Active pages

Content pages are analogous to web pages, while Portal pages are analogous to documents. Content pages can contain dynamic content via use of macros (which can make use of queries). Portal pages have content that is essentially static in nature. Active pages contain Velocity or Javascript code and cannot be edited using Rich Text mode.

You can specify a page's type when you edit it in the Wiki editor. See Editing Wiki Pages. The default type for a new page is Content page.

#### **Access the Wiki**

The Wiki is available in both the Repository scope and project scope.

To access the Wiki:

- 1. In the Open Project or Project Group dialog, select either Repository, or select a project, and open the selected scope.
- 2. In the Navigation pane, expand the Documents and Pages topic. The Default Space topic appears which gives access to the default Wiki space. If you have created other Wiki spaces, they appear as child topics under the **Documents and Pages** topic. (Documents also appear in the **Documents and Pages** Navigation topic.)

#### Navigating between Wiki Spaces

When multiple Wiki spaces exist in for the current scope (repository or project), they appear as child topics of the Documents and Pages topic in Navigation. Click any Wiki space name in Navigation to navigate into a different Wiki space. The node expands and the Navigation focus shifts to the **Home** topic of the Wiki space you selected.

Note that if you click on Default Space, the Navigation shifts to the main Home topic - as the project's Home pages is also the main page of the Default (Wiki) Space.

See also: Create a New Wiki Space.

## **Navigating to Wiki Pages**

There are two main ways to locate and access Wiki pages: browsing and searching.

Browsing page names in Navigation is available if the space doesn't contain too many pages. In this case, you can just click on the name of a listed page to navigate to it. If there are too many pages to display in Navigation, (generally more than 20), you can click the Too many documents... text in Navigation to load the space's Index page. The Index page provides a table of all the content in the Wiki space in alphabetical order by name. The name of each item in the table is a clickable link to that item. The table in the Index page can be sorted on different columns by clicking on the column

headers. The **Index** topic appears in Navigation when there are not too many items in the space, and you can click it to access the Index page.

When your Wiki contains many pages, searching may be the best way to find what you want. Start at the Search box in Navigation. For information, see Search the Wiki.

## **Create a new Wiki Space**

You can create Wiki spaces as needed for different types of discussions and collaboration. You can think of Wiki spaces like folders in other applications, except that they are not hierarchical... that is, you cannot create Wiki spaces inside a Wiki space. In each space, you can create as many pages as you need. (You can also create Documents in spaces.)

To create a new Wiki space:

- 1. Open the project in which you want to create a new Wiki space.
- 2. In the **Navigation** pane, select the **Documents and Pages** topic. The project Wiki's main page appears.
- 3. Click the **Create** button on the page's toolbar and select **New Space**. The **New Space** dialog appears.
- 4. Enter a title and name for the new Wiki space in the **Space Title** and **Space Name** fields, respectively. (Name is the system ID and can contain only ASCII characters.)
- 5. Click **OK** to create the new Wiki space with the title you specified. The new space title appears in Navigation, and the new space's Home page opens with default content, which you can edit and modify.

#### Tip

Alternative: You can also create a new Wiki space using the **New Space** item of the **Actions** menu found on the Home page of all Wiki spaces. Note that spaces created this way are *not* created as children of the space in which you are currently. Spaces are not hierarchical, so each new space appears at the same level under the **Documents** and **Pages** topicin Navigation.

## Create a new Wiki Page

There are 2 ways to add a new Wiki page:

- 1. Use **Create New** found on the toolbar of each space's Index page, and choose the **Classic Wiki Page** option.
  - It doesn't matter which Wiki space you are in currently when you invoke the **Create New** operation. You can specify the name of the target space in for the new page in the **Create Classic Wiki Page** dialog.
- 2. Edit any Wiki page and create a link a page name that does not yet exist, and then save the edited page and click the new link.

If you did not specify a space name in the link, the new page will be created in the same space as the page containing the link. Otherwise, the new page will be created in the specified space.

## Tip

When creating a new Wiki page, the **Name** field is the page's system identifier (ID). It must be unique within the space, and only ASCII characters are allowed.

The **Title** field is a human-friendly title that can contain non-ASCII characters. If specified, the value of Title appears in Navigation. (If not specified, the value of Name appears.) You can modify the Title any time using the Change Title action on the wiki editor toolbar.

## Create a new Wiki page via a link

To create a new Wiki page by defining a link to it in an existing page.

- 1. Navigate to an existing Wiki page.
- 2. Click the **Edit** button to open the wiki editor.
- 3. Enter the name of the page you want to create, enclosed in square brackets. For example: [My New Page].

If you want the new page created in a specific space, specify the space name in the link: [SpaceA.My New Page].

- 4. Click the Save and Close button. A link to the new page, formatted in red font color, now appears on the current Wiki page.
- 5. Click on the link to the new page. The new page opens in the Wiki Mark-up editor.
- 6. Use the editor to add content to the page and click the **Save and Close** button when finished.

## Create a Work Item from a Wiki Page

You can use the Extract Work Item feature to create a new Work Item from some selected content in a Wiki page. This is useful when you have been using the Wiki to collaborate on some content like requirements or test cases, and you now want to have these tracked as Work Items so they can be managed with workflow, assigned to people and Time Points, planned and schedules, and so forth.

To extract a Work Item from Wiki content:

- 1. Navigate to the wiki page containing the content that you want to extract into a new Work Item.
- 2. Select the text in the page that you want to be the description of the new Work Item. The **Extract** Work Item button is enabled. (Click Expand Tools of the page toolbar is hidden.)
- 3. Click the Extract Work Item button. The Select Project and Type for new Work Item dialog appears.
- 4. By default the new item will be created in the current project. If you want to create it in a different project, select the project name in the Project list.

- 5. Specify the type of the new Work Item by selecting a type in the **Type** list. The list items are determined by the Work Item types configured in the selected project.
- 6. Choose a destination for the new Work Item. By default it will be created in the Tracker (i.e. the Work Items topic). If you want the new item to be created and contained in a Document, choose **Create in Document** and then select the space and name of the target Document. The Document must already exist... you cannot create a new Document this way.
- 7. The new Work Item is created and opened in the Work Item Editor which opens in a frame in the current wiki page. Edit the new item as desired and click the **Create** button to save it to the repository.

The **Title** field of the new Work Item is filled in with the first sentence of the selected text, up to first period character, or first 20 words. If there is no period among first 20 words, the entire selection is written into the **Description** field and you should supply content in the **Title** field.

After you Save the new Work Item, the Tracker frame disappears. A link to the new item now appears at the point where you began selecting content. You can click this link to navigate to the Work Item in the Work Items topic, Table view. In the **Hyperlinks** section of the Work Item Editor, you will find a link to the wiki page from which the Work Item was extracted.

#### Removing a Work Item Link

If at some point you want to remove the generated Work Item link from the wiki page, you can do so by deleting the generated macro using the Wiki Markup editor. In the editor look for {workitem: projectid/workitem-id} where projectid is the ID of the project containing the linked Work Item, and workitem-id is the ID of the linked Work Item. Delete this tag and save the page to remove the link.

The link no longer displays in the wiki page. Only the link is deleted - the Work Item created from the wiki page content (using Extract Work Item) cannot be deleted except in the Tracker or Document (see Delete Work Items). The Work Item will still contain a link to the wiki page in the **Hyperlinks** section of the Work Item Editor. You can delete the hyperlink from there if you wish.

#### **Edit Wiki Pages**

Once Wiki pages have been created, you can use the Wiki editor to create and modify content. There are 2 options for editing their content:

- "Wiki Markup", using Wiki syntax (default)
- "Rich Text" (WYSIWYG)

Information on using each of these options appears in later sections.

## Launching the Wiki Editor

Access to editing is hidden by default. Click **Expand Tools** to expose the wiki page toolbar at the top of the page. Click the **Edit** button to launch the Wiki editor. The editor opens in the default mode "Wiki Markup", i.e. text-based editing using Wiki syntax.

When you open the Wiki editor, the **Rich Text** button appears on the button bar at the top of the editor pane. Click this button to switch to Rich Text (WYSIWYG) mode. You can switch back to plain text mode using the **Wiki Markup**. You can switch between the two modes any time.

Note that pages containing Velocity macros or other scripting do not show the **Rick Tex** button and cannot be edited in Rich Text mode.

#### **Using Rich Text mode**

If you are not familiar with Wiki syntax, the Rich Text mode is a good way to get started creating Wiki content. This mode does not currently support all content creation and editing features however. For example, to add images you must use the plain text (Wiki Markup) mode. You cannot use this mode to enter Velocity, Javascript, or HTML/CSS code. The embedded Wiki syntax help provides information and examples for these and other elements and constructs.

In Rich Text mode you can format text content with graphical controls similar to those found a word processor application. These controls appear on the Rich Text toolbar (shown at the top of the editor pane when Rich Text mode is active). The toolbar currently provides these capabilities:

- Cut, copy, and paste
- Undo, redo
- Text style picker (formats selected or next text as normal or one of several levels of headings).
- Basic paragraph and heading styles
- Font for matting: bold, italic, underline, strike-through, superscript, and subscript
- Insert horizontal rule
- List formatting: numbered and bulleted
- Paragraph indent/outdent
- Insert Link or Plain Wiki syntax
- Remove existing formatting of selection or entire text
- Insert table and table operations (enabled after insert)

#### Link to other content

To create links to other content, you must use the Wiki syntax for links. Rich Text mode provides the possibility to create links to other pages without switching back to Wiki Markup mode. When you click the Internal Link on the editor toolbar, a placeholder link construct is inserted to the page. Please refer to the embedded Wiki syntax help that appears when you edit pages.

### **Add Attachments**

The **Attachments** section is available in Rich Text mode, enabling you to add or remove attachments to/from the currently edited Wiki page. Note that before you can insert images in text, you must add them as attachments. See working with attachments for more information.

### Using Wiki mark-up

This edit mode enables you to format text, create links, display images, and perform other editing operations using standard XWiki syntax. You can also enter Velocity macros and scripting, Javascript, and HTML/CSS code. This text-based editing is the default mode when you click the **Edit** button in the toolbar of each Wiki page.

Note that in pages that support Rich Text (i.e. they contain no scripting), you can switch between the Wiki Markup and Rich Text (WYSIWYG) mode any time during an editing session. You can use the **Preview** button to see how your unsaved edits will look in the wiki page after saving.

## **Using Wiki Syntax Help**

The **Wiki Syntax Help** link appears at the right-hand side of the screen when the Wiki editor is active in Wiki Mark-up mode.

The panel displays a number of labeled bars which represent different Help topics. The categories are for different document elements, formatting, and macros for embedding tracker and other dynamic content in wiki pages.

Clicking on any of the category bars expands it to reveal the syntax Help for the selected category.

This embedded help provides basic help for most editing tasks. More comprehensive syntax help is provided in a special wiki page. To access this page, look for the **Detailed Syntax Help** link at the bottom of the embedded syntax help in the Wiki editor. You can bookmark that page in your browser for future reference.

## Display images

You can display images in Wiki pages after they have been uploaded as attachments. Use the Attachments section to upload your image file(s) and save the page. Then use the "Insert placeholder {image}" button on the toolbar to insert the syntax for an image reference at the desired place in your text. Replace <code>image-attachment-filename.ext</code> in the inserted {image} macro with the file name and extension of an attached image.

#### **Create tables**

The {table} macro renders a table in the Wiki page. Table rows and columns are defined between two {table} elements. Each line represents a row. Columns are separated by a vertical bar character (|). The first line may be optionally defined as the table header.

Various parameters of the {table} macro provide control over some formatting. The id parameter, if specified, causes the value to be rendered in the "id" attribute of the table's HTML. This can be subsequently referenced in scripting.

For more specific information, consult the Tables section of the Wiki Syntax Help in the Wiki editor.

## **Working with Attachments**

You can attach files of any type to a Wiki page in the **Attachments** section. The page section is hidden by default when you are reading a page. When hidden, or if visible but scrolled out of view, you can jump directly to it by clicking the **Attachments** button on the Wiki page toolbar. Graphical controls are provided for adding and removing attachments.

After attachments have been added to a Wiki page, you can see their links in the page and optionally download them to your local system. Images must be uploaded as attachments before they can be displayed in the page content.

By default, the maximum file size of attachments is limited to 50 MB. The limit is configurable in Administration.

See also: Wiki History - Download Attachments from a Revision.

## Adding an Attachment to a Wiki Page

To add an attachment to a Wiki page:

- 1. When viewing (i.e. not editing) a Wiki page, click the **Expand Tools** Attachments to expose the **Attachments** section. Any existing attachments are listed in a table.
- 2. Click the (Edit) icon at the bottom left corner of the Attachments section. The graphical controls to add an attachment appear.
- 3. Enter a title for the attachment. (Spaces not allowed.)
- 4. In the **File Name** column, click the **Browse** button and navigate to the file you want to attach.
- 5. If you want to add another attachment, click the icon in the same row and repeat the previous step.
- 6. After adding all attachments, click the **Save** icon.

#### **Note**

Your system or project administrator can optionally limit attachment file size. Contact your administrator if you are unable to upload an attachment due to such restriction.

## Update an existing Attachment on a Wiki Page

You can update an existing attachment in the Wiki Markup editor or the Rich Text editor. You do this by adding an attachment in the Attachments section of the editor and checking the Update existing attachment option on the row of the new attachment. The file you upload for the new attachment must have the same file name and extension as the existing attachment you want to update.

## Remove an Attachment from a Wiki Page

If an attachment is no longer needed, it can be removed from a Wiki page. In the table of attachments in the Attachments section, click the - icon on the row of the attachment you wish to remove. The attachment is removed from the page when you save the change. It still exists in the page's history and can be accessed there.

## Download an Attachment from a Wiki Page

In the **Attachments** section, locate the attachment you want to download in the table of attachments. The file name of the attachment appears as a link in the **File Name** column. To download the file, right click on the file name and choose the command that saves a remote file to your local system. For example, in Firefox choose Save Link As.

## Add Dynamic content in Wiki Pages

The wiki has a number of *macros* that can be embedded in wiki pages using the Wiki Markup editor. Some macros merely provide formatting - the {code} macro for example. Others provide content structures such as tables, or display an image. Still others provide content from the tracker or the system which is dynamically retrieved, formatted, and displayed in a wiki page. Such dynamic

content may include subsets of Work Items, or information about project users, or links to other projects and their wiki content.

This section covers several of the most commonly needed macros. The full listing is provided in the embedded Wiki Syntax Help, accessible from the Wiki editor.

## The {project} macro

The {project} macro can be used to get a link to Polarion project. You can find more details and usage examples in the Projects topic in the Wiki Syntax Help, available in the Wiki editor while editing a wiki page.

## The {workitem} and {workitems} macros

The {workitem} and {workitems} macros can be used to extract information about, and create links to Work Items. The {workitem} macro retrieves a single item. {workitems} retrieves multiple items. The content returned, and the format, can be controlled via several parameters, including a parameter that can contain a simple or complex Lucene query, and a parameter than can contain a SQL query (see next section). You can find more details and usage examples in the Work Items topic in the Wiki Syntax Help, available in the Wiki editor while editing a wiki page.

## Using SQL queries to retrieve Work Items:

Polarion has an embedded SQL database layer that can facilitate Work Item queries in traceability reports and other complex reports in Wiki pages where a Lucene query would be highly inefficient and consume undue time and system resources, or not work at all. For example, you could not readily formulate an outer join in Lucene, but it's easily done with SQL. Primary storage of artifact data remains in the Subversion repository, but the SQL layer maintains a special read-only index of the content which can be queried in the {workitems} macro. Results of the SQL query can be read by all users using any product license (assuming they have read permissions for project wiki pages).

You can include the sqlQuery parameter in the  $\{workitems\}$  macro to execute a SQL query over Work Items. When this parameter is used, other parameters that can also be used in the macro are limited to:

- display
- expand
- top
- fields
- width
- height

#### **EXAMPLE:**

```
{workitems: sqlQuery=
    SELECT WI.C_URI
    FROM WORKITEM WI, PROJECT P
    WHERE WI.FK_URI_PROJECT = P.C_URI
    AND P.C ID = 'playground'}
```

## **Note**

If a query's processing exceeds limits set in the system configuration, it may be halted, and a notification sent to the system administrator in order to prevent other users updates from being blocked by long-running queries.

## Tip

The graphical Query Builder in the Work Items topic contains an element SQL. It can be used alone to filter the Work Items table by a SQL query only, or in conjunction with other visual query elements. As with all Query Builder queries, the visual query is transformed to a URL, and can optionally be converted to text via the Query Builder's drop-down menu.

The **SQL** element can be useful if you want to open result of the {workitems} macro filtered by the sqlquery parameter in the Work Items table, instead of entering a fully expanded query with IDs. For more information on using the graphical Query Builder, see Construct Queries Graphically.

## The {users} macro

The {users} macro returns a list of users for a specified project. The content returned can be controlled via several parameters, including a parameter that can contain a simple or complex guery. You can find more details and usage examples in the **Users** topic in the Wiki Syntax Help, available in the Wiki editor while editing a wiki page.

## The {toc} (Table of Contents) macro

The macro generates a table of content based on the headings in a wiki page. The content returned can be controlled via several parameters. You can find more details and usage examples in the **Headings** topic in the Wiki Syntax Help, available in the Wiki editor while editing a wiki page.

#### The {import-automated-test-results} macro

The {import-automated-test-results} macro embedded in a wiki page enables page users to launch import of automated test results from any wiki page, not just Test Run pages. In regular wiki pages, the testrun parameter must be specified. Complete information on parameters is provided in the Wiki Syntax Help, available in the Wiki editor.

#### The {pages} macro

This macro renders a list of pages in a project or project group. For parameters and example, see the **Pages** topic in the embedded Wiki Syntax Help in the Wiki editor.

### The {activity-stream macro}

This macro displays the stream of activity which has occurred on your Polarion portal. The activity stream is similar to the news feed feature on social media sites. The macro's parameter enable you to restrict the amount of information displayed. For details, see the Activity Stream topic in the embedded Wiki Syntax Help in the Wiki editor.

Users of wiki pages displaying the activity stream can enter a query to filter the stream to restrict the display to some specific information... some user's activity, or build activity, or a date, or date range for example. Users can optionally click the provided link to display the Activity Stream page which not only shows the stream and filtering, but provides quick links to filter for commonly needed stream information.

Administrators can tune the system configuration to control the age and/or number of activities maintained in the history. See: Properties for Activity Stream.

## Use queries in macros

Macros such as {workitems} provide a parameter query in which you can specify a simple or complex query to retrieve data. The query syntax can be copied from a visually constructed query in the Query Builder, found in the Table and several other views in the Work Items topic. (See Search Work Items). For example, you can use the {workitems} macro to display a listing of unresolved blocker and critical defects on a wiki page by specifying a query as shown in the following listing:

```
{workitems: query=NOT HAS_VALUE:resolution AND type:defect AND
(severity:blocker OR severity:critical)}
```

You can use the Query Builder to construct the query and then copy the query parameter shown in the page URL into the macro's <code>query</code> parameter.

## Reference fields in queries

Queries can specify which fields should be displayed in the results by including the fields in the query. For example, the following query will display only the Work Item ID, Title, and Severity fields in the results:

```
{workitems: query=resolution:fixed AND type:defect | fields=id, title, severity}
```

#### **Note**

the fields parameter must reference field ID, not field label. Field IDs are case sensitive and do not contain spaces. For example, for the field that logs time spent on a Work Item, a macro query must specify time spent (the field ID), and not Time Spent (the field label).

The same is true in the sortby parameter which specifies on which field the results should be sorted: specify the field ID, not field label. Example:

```
{workitems: query=resolution:fixed AND type:defect | fields=id, title, severity
| sortby=severity}
```

The rule of thumb is that any macro parameter that references one of more Work Item fields must contain field ID(s), not field name(s).

## Display charts in Wiki Pages

Polarion includes the FusionCharts free library that provides many types of charts. Usage is outside the scope of Polarion documentation, but the library's documentation is bundled into distributions. You can find a link to it in the **Dashboard** topic of the Wiki Syntax Help (accessible when any Wiki page is in edit mode).

## **Extend Wiki Pages with Velocity**

Read-only with (REV licenses.)

Users of the Polarion Java API can extend active wiki pages by means of Apache Velocity (see http://velocity.apache.org/engine/1.4/user-guide.html). For information about Velocity variables available via the API, see: Velocity Variables for Active Wiki Pages.

## **Create interactive Report Pages**

All except: (REV licenses.)

A major use case for dynamic-content wiki pages is reporting. For example, a page can be created to show all unresolved critical-severity items for some time point. That can be useful, but what if a report consumer wants to see unresolved critical-severity items for different time points, or assigned to different users?

It is possible, of course, to create separate pages, or separate sections in pages to report on each different set of parameters. But it would be more useful to enable the report consumer to interactively change some key parameters using simple graphical controls and obtain variant results. Polarion provides this functionality through a set of parameterization macros. Users who know how to code wiki pages can easily create report pages for less technical users, enabling them to create variant output for the same basic report, which they can optionally export to PDF just as with any wiki page.

#### **Parameter Macros:**

Polarion provides the set of parameter\* macros which enable you to create interactive report pages:

- {parameter} declares a parameter of a specified type and value(s) which can be referenced in other elements on the page.
- {parameter-form} declares the start and end of an interactive part of the page which displays interactive "widgets" or controls rendered by other macros of this group.
- {parameter-editor} displays an edit control in which user can edit the value(s) declared in a {parameter} statement.
- {parameter-form-submit} provides a button widget for submitting user changes to one or more {parameter-editor}s. This macro is available in Baselines.
- {parameter-form-save} provides a button widget for saving user changes to one or more {parameter-editor}s, overwriting current values in the respective{parameter} declaration(s) in the page source.

The {parameter} and {parameter-editor} macros take one or more arguments. The complete syntax reference is provided in the embedded Wiki Syntax Help (click Detailed Syntax Help at the bottom of the Syntax Help pane in the Wiki Editor).

## **Basic Parameter Usage:**

The {parameter} macro enables you to define a parameter which can be subsequently referenced in other page elements... in the query argument of a {workitems} macro, or the id argument of a {parameter-editor} macro.

You should declare all the {parameter}s for the page first, before any other page content.

##Page Parameter Declarations:

```
{parameter:myParam1|type=string|value=aaa|allowed-values=aaa,bbb,ccc}
{parameter:myParam2|type=user|value=cJames,kRoth|multi=true}
{parameter:query|type=string|value=type:defect}
{parameter:dateFrom|type=date|value=2011-05-05}
{parameter:dateTo|type=date|value=@current}
```

You can then reference any declared parameter by appending its ID to the \$pageParameters variable following a period character (".") delimiter.

## **Examples:**

```
Updated from {parameter-editor:dateFrom} to {parameter-editor:dateTo}
...
{workitems:query=$pageParameters.query $!queryOpen AND assignee.id:
   ($pageParameters.getAsString($pageParameters.user, " OR ")) AND
   updated:[$pageParameters.getDateForQuery("dateFrom")
TO $pageParameters.getDateForQuery("dateTo")]}
```

Here is an example of a report page reporting on Work Items with user-selectable team, Work Item type, resolution, and Work Item update dates:

```
##Parameter Definitions
{parameter:team|type=string|value=teamX|allowed-values=teamX,teamY}
{parameter:query|type=string|value=type:defect}
{parameter:justOpen|type=boolean|value=true}
{parameter:teamX|type=user|value=mTest,aSeller|multi=true}
{parameter:teamY|type=user|value=sDeveloper,pGUI|multi=true}
{parameter:dateFrom|type=date|value=2011-05-05}
{parameter:dateTo|type=date|value=@current}
1 Work Items By Team
{parameter-form}
Select Team: {parameter-editor:team} \\
Query: {parameter-editor:query} {parameter-form-submit} \\
Not Resolved: {parameter-editor:justOpen} \\
Updated from {parameter-editor:dateFrom} to {parameter-editor:dateTo} \\
{parameter-form}
Team members: $pageParameters.render($pageParameters.team,", ")
#if ($pageParameters.justOpen)
#set($queryOpen = " AND NOT HAS VALUE:resolution")
#end
{workitems:query=$pageParameters.query $!queryOpen AND assignee.id:
($pageParameters.getAsString($pageParameters.team, " OR "))
  AND updated: [$pageParameters.getDateForQuery("dateFrom")
TO $pageParameters.getDateForQuery("dateTo")]}
```

For more examples, review the E-Library project that ships with all Polarion products. You can open it like any other project, assuming that your administrator opted to install sample date when installing your Polarion system.

## **Create charts in Wiki Pages**

Polarion integrates the Highcharts library, which enables users to easily create and show various types of simple and complex charts in wiki pages, including Dashboards. You can implement these charts using special Highcharts code. Chart types are documented with examples in the embedded Wiki Highcharts Help. You can access this help via the embedded Syntax Help in the Wiki Mark-up Editor, or directly by its URL in your Polarion portal.

To access Highcharts help via the Wiki Syntax Help:

- 1. Open any Wiki page for editing.
- 2. In the Wiki Syntax Help panel, click the **Wiki Highcharts Help** link at the bottom of the panel.

Alternatively, you can go directly to the Wiki Highcharts Help on your system using this URL:

http://[HOST SERVER].[DOMAIN]/polarion/#/wiki/Doc/HighchartsHelp

... where [HOST SERVER] is the name of your Polarion server, and [DOMAIN] is your internet domain.

**Example:**http://polarion1.yourdomain.com/polarion/#/wiki/Doc/HighchartsHelp

See also: Configure Highcharts Export.

## **Export Wiki content to PDF**

You can export the content of Wiki pages to Portable Document Format (PDF). During the export operation you can set options for page size, orientation, fitting output to page, bookmark generation, and inclusion of a header and footer. You can save the generated PDF to any accessible storage (local or network).

The content of the header and footer in generated PDF documents is configurable in Administration, either globally for the repository, or for a specific project. Headers/footers can include such information as the page name, history revision number, creator's user name, the export date, current page number and total number of pages (e.g. 7/20). For information, see: Configuring PDF Export.

#### Tip

The configuration for PDF headers/footers changed significantly in Polarion 2011. If you update from an earlier version in which you configured PDF export for wiki pages, you will need to reconfigure it in your current installation. See: Configuring PDF Export.

To export a Wiki page to PDF:

- 1. Navigate to the Wiki page you want to export.
- 2. In the page toolbar, click the **Actions** button and select **Export to PDF** on the drop-down menu. The Export to PDF dialog appears.
- 3. In the dialog, select the desired options for the PDF document, and click **Export**.

## Tip

You can change the settings and quickly re-export the same page until you close the dialog.

4. Save or open the exported PDF according to options presented by your computer's operating system.

## Tip

You can export any historical revision of a Wiki page to PDF. The procedure is basically the same as for a Wiki page (i.e. find the export action in the **Actions** button menu), except that you must first open the page's History and open the desired revision by clicking on the revision number.

## Print a Wiki Page

You can print the current Wiki page to any accessible printer.

To print the current page:

- 1. Click the **Actions** button on the Wiki page's toolbar.
- 2. On the drop-down menu, select **Print**. The **Print** dialog appears.
- 3. Wait for the printer-friendly version of the page to fully load in the **Print** dialog, then click the **Print** button in the dialog. Your operating system's Print dialog appears.
- 4. In the operating system **Print** dialog, select the target printer device, set print properties if needed, specify the output file name, and launch the print job.
- 5. Close the Polarion **Print** dialog using the close control in the title bar.

#### Search the Wiki

It can be faster to locate Wiki pages using the Search field at the top of the Navigation pane. However, you need to have some idea of the page name you want to access... at least the first several characters. Typing in the Filter box starts an incremental search of all the content in the project. You can limit the search to the Wiki. After typing one or more characters in the Search field, the Navigation panel presents options for limiting search to different types of content.

## Using global search

You can also search for Wiki content using the global search. Type some text in the Search field and press your Enter or Return key. In the search results page, click the **Wiki** link to filter the results for Wiki pages only.

The search results page provides controls to refine your search or start a new search.

## **Working with Wiki History**

The Polarion Wiki uses the underlying Subversion repository to store Wiki content. Consequently, the content is versioned and history is maintained just as with other artifacts in the system.

This section covers basic operations with Wiki history.

## Finding a specific revision

Each time a Wiki page is changed and saved, a revision is created in the repository and added to the history.

To see the history of any Wiki page:

- Navigate to the space and the page.
- 2. Click on the **History** toggle which appears on the page toolbar.

The History shows all revisions of the current page. There you can access any revision of the page, compare revisions, roll back changes, and access revisions of any attachments.

Revisions are listed by revision number. Revision numbers are clickable, and lead to the specific revision of the page, including any attachments, links, etc. Revision numbers will not necessarily be sequential, as they are for the repository as a whole, not for the specific Wiki page. The most recent revision appears first in the table.

#### Comparing Wiki Page revisions

To compare two Wiki page revisions in History:

- Navigate to the Wiki page whose revisions you want to compare and click the History toggle button in the page toolbar to show the table of revisions.
- 2. In the left-hand column, check the boxes on the rows of the two revisions you wan to compare. (You can only check two boxes at a time.)
- 3. Click the **Compare** button at the top of the page. The comparison page loads in your browser.

The page displays counts of things that were added, changed or deleted between the earlier and the later of the two revisions being compared. Two buttons labeled Previous and Next enable you to navigate among the changes in the page, enabling you to easily see what changed and how.

Three buttons at the top of the page enable you to select different views of the comparison. The default is **Page**. When selected, you see the what the compared pages actually look like in the wiki. The **Source** shows you the underlying wiki markup language, enabling you to compare changes that occurred there which might not necessarily reflect when the Wiki page is viewed. The Attachments button enables you to compare the attachments to the two revisions.

#### Exit a historical comparison:

When you are comparing to Wiki page revisions, the **Back to History** button appears at the top above the two revisions. Click it to exit the comparison and return to the table of revisions on the History view of the page.

## Roll back changes

You can roll back the changes made in any revision of any Wiki page.

To roll back changes made in a revision:

- 1. Navigate to the desired Wiki page.
- 2. Click the **History** toggle button on the page toolbar.
- 3. In the table of revisions, locate the revision you want to roll back.
- 4. Click the **Rollback** link in the **Action** column.

#### **Download Attachments from a Revision**

Each revision of a Wiki page contains a version of any attachments. If attachments are added, removed, or updated, a new revision of the page is created in the repository which includes the attachments current at the time of the revision. Revisions of any Wiki page are shown in its History (see Finding a Specific Revision).

To access the attachments for a specific revision, open the revision in the history (click on the revision number), click the **Attachments** button in the Wiki page toolbar, and locate the attachment in the Attachments section. Right click on the desired attachment and use the save link command on your browser's right-click menu to download and save the attachment to your local system.

## **Using the Activity Stream**

If the page author has included the {activity-stream} macro in the page mark-up, the portal Activity Stream appears on the page. The Activity Stream is similar to the news feed feature found on many social media websites. It displays a stream of activities taking place on the Polarion portal. Activities may have different sources such as builds, or Work Items. Some pages contain the Activity Stream by default: the portal Home page, and users' "My Polarion" pages, for example.

Each activity in the stream displays some basic summary information such as the name of the user who triggered the activity, and when the activity took place. Hovering over an activity highlights it, and clicking on it (or on the **Show More** link, if present) expands the activity to show more detail. Each activity contains a link to the activity source... a Work Item, build, or commit, for example. Clicking the link takes you directly to the source item.

If the activity source is a Work Item, you can post a comment directly in the activity in the Activity Stream by clicking the **Add Comment** link. You must have permissions for the project which allow you to write and save comments in Work Items. If you don't, you will see a message when you try to save the comment using the **Save Comment** link.

The activity stream is refreshed when the wiki page containing it loads or reloads. You can explicitly refresh the stream to see new activities using the (Refresh) icon at the top of the activities.

## **Searching the Activity Stream**

You can enter a search query in the provided field to filter the activities listed in the stream. For example, you might search for all items containing a word or words, or for specific types of activity.

#### Full-text Search:

The simplest and most intuitive way to search the Activity Stream is to enter some word or words. The results will show all activities (if any) that contain the search string. If the search string is a user name, the results will contain both activities generated by the user, and activities containing the name somewhere in the content.

If you enter multiple words, the search logic is "AND". So for example, if you search for "smith language", the query logic is "smith AND language". The results will contain activities generated by any user named Smith, activities mentioning "smith", and activities mentioning "language". You could specify the AND operator, but it is not required. If you want to exclude some term, you can precede it with the operators AND NOT. For example: "smith AND NOT language".

If you enter multiple words and you want results with one or the other, you can enter the OR operator: "smith OR language".

## **Search by Activity source and Type:**

You can search for a specific stream source and type by entering a query in this format: sourceId: [ID]. For example, you could filter for activities around Work Items with this query: sourceId:workitems.

You can refine the filter by including the type parameter in the filter query. For example, you could filter for activities around Work Items which are updated with this guery: sourceId:workitems AND type:updated.

You can also enter values for an activity type like updated, created or build to search for activities with that activity type. You can use AND/OR logic to search on multiple types, or keywords and types: a user name and "created" type activities, for example.

For complete information on the activity sources and types that you can query for, see: Activity Stream Sources and Types.

#### Wiki administration operations

As is usual with Wikis, users can update pages including updating of content and performing administrative operations such as adding Wiki spaces and adding Wiki pages, and also deleting spaces and deleting pages (below). However, in Polarion users must have permissions for these actions in the scope in which the perform the actions (project, project group, or repository).

## Delete a Wiki Page

To delete a Wiki Page:

- 1. Navigate to the page you want to delete so that you are viewing it.
- 2. Click the so button in the toolbar and choose **Delete** on the drop-down menu..
- 3. Respond **Yes** to the confirmation prompt.

## Tip

You can also delete a Wiki page from the Index page of the space in which the page resides. Check the box next to the name of the page(s) you want to delete, and click the **Delete** button on the Index page tool bar.

## **Delete a Wiki Space**

To delete a Wiki space:

- 1. In the **Documents and Pages** topic in Navigation, select the space you want to delete and then navigate to its Index page.
- 2. In the toolbar of the Index page, click the drop-down actions menu (displays a gear icon) and choose **Delete Space**.
- 3. Respond **Yes** to the confirmation prompt.

## **Microsoft Project import-export**

## ALM

You can import tasks from Microsoft Project into a new or existing Polarion project to create Polarion Work Items. You can also import MS Project milestones which become Time Points in Polarion. You should use Microsoft Project's export feature to export a project to an XML file. You will need to have this file available to in a location accessible to the computer on which you are accessing Polarion.

The following points are important to note:

- Milestone summary tasks are imported as a Time Point and also as a Work Item, so that the grouping of its subtasks is preserved. Ordinary milestones are imported as Time Points only.
- Imported milestone summary tasks and all subtasks are assigned to the corresponding Time Point.
- All predecessors of a Milestone task (recursively, including their subtasks and predecessors of subtasks) are assigned to the corresponding Time Point. If more than one Time Point is found for a single task, the earliest one is assigned.
- If all subtasks of a summary task are assigned to some Time Point, the summary task is also assigned to a Time Point, i.e. to the one with greatest time found in any subtask.
- Custom Fields that are not configured are imported as the string type.
- Field mapping is as follows:

MS Project Milestone	Polarion Time Point
Name	name
Notes	Description
Finish	time

To import Work Items from MS Project:

- 1. Create a new project or open the project in which you want to import Work Items from MS Project.
- 2. In the **Navigation** pane select the **Work Items** topic.

- 3. Locate the drop-down menu control on the toolbar above the Work Items table in the Table view.
- 4. Click the control and select **Import** from the menu. The **Import Work Items** dialog appears.
- 5. If you want to import only Work Items, select Without User Creation.xml in the Configuration field. If you want to import both Work Items and users, select **Default.xml**
- 6. Click the **Browse** button and use your file selection dialog to select the XML file exported from MS Project.
- 7. Click the **Start** button to run the import operation and import the Work Items (and users if you chose that option).

See also: Exporting Work Items: Exporting to Microsoft Project.

## **Export Work Items to Microsoft Project**

## ALM

The procedure to begin the export is basically the same as for exporting to a document. When you choose the option to export to MS Project, you have another option Export Time Points as Milestones. If selected, the Time Points to which at least one of the exported items is assigned, are exported as milestone tasks. Mapping of fields is the same as in case of import from MS Project.

Milestone predecessors are tasks that correspond to Work Items assigned to the relevant Time Point. Only the minimal set of predecessors' links is created: predecessors of predecessors are not direct predecessors of a milestone, and also subtasks of predecessors are not direct predecessors of milestones. Exported milestones are always top level tasks. They are inserted at the first possible top level position after all items that belong to the Time Point are exported.

To export a Project to MS Project:

- 1. Open the project you want to export (see: Access Projects).
- 2. If you want to export the entire Project, select the Work Items topic in the Navigation panel. If you want to export a subset of the project, select an appropriate Shortcut in one of the Shortcuts topics, or execute an appropriate saved query, or a custom query built with the Query Builder in the Table view in Work Items.

#### Note

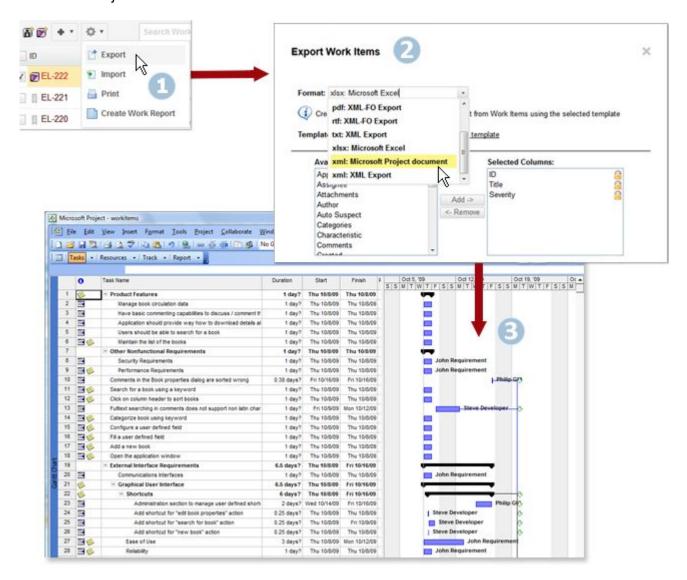
## Sort items by ID in ascending order:

If you want the output from Polarion to be exactly the same as the input from MS Project, then items need to be sorted by ID in ascending order.

(It is not sorted this way in Polarion by default.)

3. Click the we button in the toolbar and select **Export** in the menu. The **Export Work Items** dialog appears.

- Select the desired target format (xml: MS Project document) and character set and click OK. The document will be created and readied for download. When processing is finished the Download button appears in the dialog.
- Click the **Download** button and choose the location to save the data file on your local system.
   The selected data is saved in an appropriate XML format which you can subsequently open in MS Project.



Exporting a Project to Microsoft Project

#### Opening the exported file in MS Project

The xml file that is created as a result of the **Export to MS Project** action should **not** be renamed with a .mpp extension before opening it in MS Project.

There are 2 ways to open the exported file in MS Project:

Using Windows Explorer context menu

Using MS Project File menu

To open the exported file using Windows Explorer:

- Selecting the exported file in Windows Explorer
- 2. Right-click on the file and choose **Open in** on the context menu, followed by **MS Project** on the submenu;

To open the exported file from MS Project:

- Open MS Project.
- Open the Open File dialog.
- 3. Choose **xml format**, select the xml file, and click **Open**.
- 4. Choose the **As new project** option.

## **Deprecated macros**

This section documents Wiki macros that have been deprecated in Polarion products. If deprecated macros can still be used this will be noted in the relevant topics.

#### Miscellaneous macros

Macros noted in this section are deprecated and should be replaced as noted.

```
{includeForm} Macro
```

The {includeForm} macro enabled insertion into one wiki page of the content of another wiki page. This macro was replaced in version 2016 by {include-page} and {include-macros}. See Classic Wiki syntax Help for details on usage of the new macros.

#### Test management macros

# ALM REQ QA

Beginning with version 2012, the following Test Management macros were deprecated and removed from the Wiki Syntax Help. However, the macros are still supported and can be used in Wiki pages.

#### {testruns} Macro

The {testruns} macro renders a table of Test Runs.

#### Parameters:

- query (optional) query used to locate test runs to display (format of the query is the same as for {pages} macro).
- **top** (optional) maximum number of test runs to be displayed.
- fields (optional) comma-separated list of test run fields to display. Supports the same fields as the {pages} macro, and following additional fields:
  - id (ID of a test run)

- o owner (owner of a test run)
- o statistics.<test result> (shows statistics for a specific test result), statistics.executed (shows total number of executions) and statistics (shows statistics for all test results and total number of executions).
- prefix (optional) page name prefix of test run pages. Defaults to "TestRun\_".
- testResultEnum (optional) enumeration that contains all test results. Defaults to "testResult".
- **sortby** (optional) IDs of the fields to use for sorting the test runs. Supports the same fields as the {pages} macro, and the following additional fields: id and owner.
- project (optional) project to load test runs from. Defaults to the current project.
- space (optional) space to load test runs from. Defaults to the current project.

#### **EXAMPLE:**

```
{testrun-results:TestRun 123}
```

#### {testrun-results} macro

The {testrun-results} macro displays results of a test run.

### **PARAMETERS:**

- **TestRun** name of the test run for which macro will display results.
- testResultEnum (optional) the enumeration containing all test results. Defaults to "testResult".
- query (optional) query used to find test cases that belong to this test run if they were not executed.

#### **EXAMPLE:**

```
{execute-tests-button:testRun=$testRun|
  query=comments.title:"Test Blocked in test run:$testRunId"|
  document=Testing/MainTestSpecification}
```

#### execute-tests-button) macro

The {execute-tests-button} macro renders a button which allows users to execute test cases.

### **PARAMETERS:**

- testRun name of the test run that defines the test cases to be executed
- document (optional) name of a Document in which the test cases to be executed are defined
- query (optional if document is specified) query used to filter the test cases to be executed
- **project** (optional if macro is used on project level) project containing the test cases. Defaults to the current project.

- sortby (optional, not available if document is specified) IDs of the fields to use for sorting the test cases. Supports the same fields as the {workitems} macro.
- testRunField (optional) ID of the Work Item field that will be auto-filled with test run. Defaults to "testRun".
- additionalFields (optional) additional fields that will be auto-filled in the form "<field>:<value>,..."

# Chapter 4: Administrator's guide

## **Administration basics**

## **Administration rights**

To be able to access Polarion's administration interface you must have a user account with Administrator level access rights. Access rights are defined by the **Role** assigned to the user account.

## Administration rights can have one of two scopes:

- 1. Global: administrations rights for the entire repository
- 2. **Project:** administration rights for a specific Project

#### The default Administration account

Polarion is installed with one default user account with user name "System Administrator" which has Global administration rights.

## The default log-in credentials for this account are:

- User ID: admin
- Password: admin

#### **Note**

## **Changing the Default Administration Password:**

The procedure for changing the password for the default System Administrator account is detailed in the Installation Guide that comes in the Polarion distribution. Look for: **Changing the default System Administrator password**. For information about passwords: Passwords.

The default System Administrator account provides access to all Polarion administration and configuration features for the Polarion system and all Projects in the Subversion repository, and access to the repository itself.

#### **Defining additional administrators**

The main System Administrator may find it useful or necessary to grant administrative access to other Polarion users. For example, if the main administrator is absent for some reason, someone else in the organization should be able to perform system administration functions with Polarion. Another example: the manager of each Project might be given administrative rights for the Project(s) he/she manages. This enables the manager to add user accounts and assign user roles for the Project, among other things.

To define an additional administrator, first create a user account if one does not already exist. You can specify the new user's role(s) for the global system and/or per Project in the process of creating the account. For an existing user, you need to edit the user's account and specify his/her role(s) for the global installation and/or per Project.

For information on creating user accounts and defining user roles, see Managing Users & Permissions.

## Administration interface

There are two ways that you can perform administrative functions in Polarion:

- 1. Graphical User Interface
- 2. XML configuration files

Most administration and configuration tasks can be done using the graphical user interface. This is the preferred approach for new or less experienced Polarion administrators.

## **Administration scopes**

Administration and configuration operations are generally possible in 2 scopes:

- **Global**: Changes and customizations affect the entire repository. To work in this scope, select **Repository** in the *Open Project or Project Group* dialog.
- **Project**: Changes and customizations affect one project only.

To work in this scope, select any existing project in the *Open Project or Project Group* dialog.

## Tip

There is another level of configuration/customization that controls behaviors and functions of the Polarion platform and features: the system configuration, contained in the system properties file polarion.properties. This file resides on the server's file system (see System Properties File Location), and requires editing outside the Polarion portal. Many users never need to modify the system configuration. The most frequently needed system configuration tweaks are covered in: Advanced System Tuning.

#### Inheritance

- Changes in the global configuration and are inherited by all new projects created after any change/customization. Changes in the global scope do not propagate to projects that already exist before the changes.
- New projects inherit the global configuration. Project-scope copies of all the global configuration files are created in the project's repository location. Changes in the project scope overwrite the content in the configuration files inherited from the global configuration.
- Changes and customizations to a project cannot be rolled up to the global scope, nor can they be propagated to other existing projects. However, a customized project canbe exported as a

project template from which new projects with identical properties, configuration settings and content can be instantiated.

#### **Precedence**

- Project-scope settings always take precedence over global-scope settings.
- Within a project-scope configuration, settings specific to a Work Item type take precedence over generic settings applicable to all Work Item types.

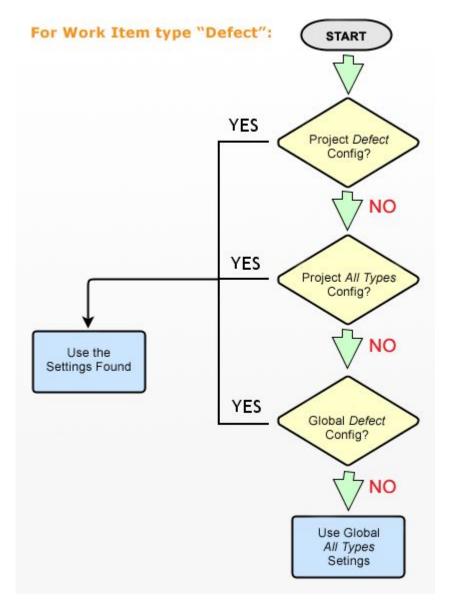
## Warning

Do not remove any project-scope configuration file in order to use the respective global configuration in the project. This will interfere with the system's ability to resolve inheritance and precedence. It can prevent users from performing workflow actions, and cause other problems.

Either restore the original revision of the project-scope file from the repository, or if the global configuration has changed since the original revision, copy the current global-scope file to the project's repository location, overwriting the project-scope file.

**Example**: Suppose a Work Item type **Defect**. When looking for configuration settings for this type:

- 1. The system looks for configuration settings for the **Defect** type in the current project.
- 2. If none is found, the system looks for configuration settings for all **Work Item** types in the project.
- 3. If none is found, the system looks for configuration settings for the **Defect** type in the global configuration.
- 4. If none is found, the system uses the settings for all Work Item types in the global configuration.



Example of configuration scope precedence

#### **Graphical user interface**

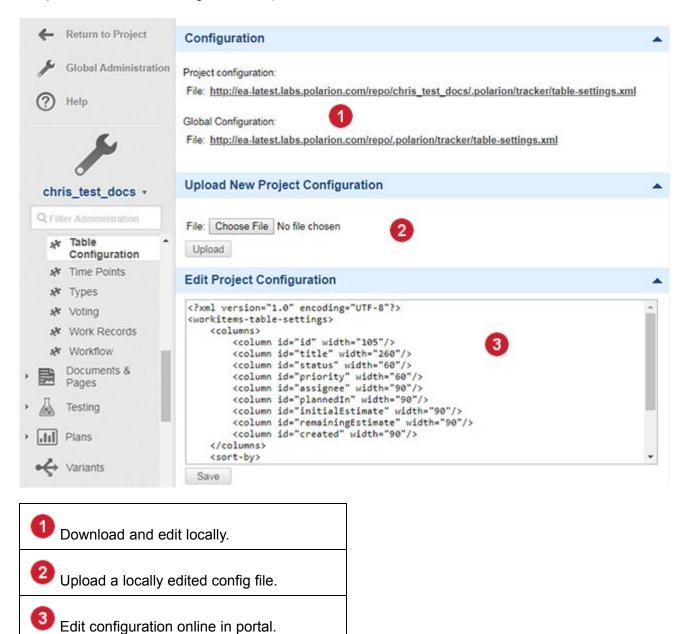
Actually, all administration functions are accessible via the graphical user interface (GUI) of the Polarion portal. However, not all configurations can be performed using graphical controls in the portal. Where graphical controls are not provided, configuration are performed by editing the respective XML configuration file. Such editing can be done on line or off line.

#### XML configuration files

Some administrative functions are performed by editing a relevant XML configuration file stored in the repository. These configurations are generally less frequently customized by most Polarion users. Where some configuration requires editing a configuration file, you have two options:

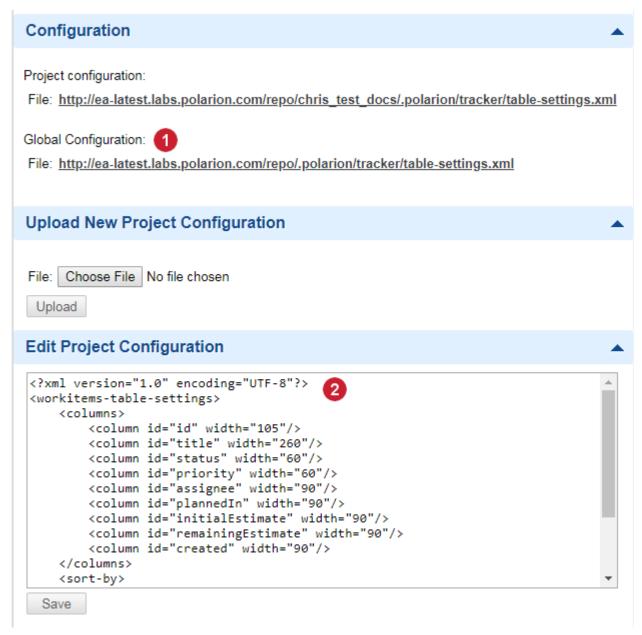
Edit the XML file online using the basic text editor provided in the portal web UI.

Download a copy of the configuration file to your local system, edit it locally, then upload the
modified file back to the Polarion portal. (Upload and download controls are provided in the
portal UI for each configuration file).

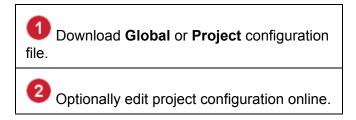


Sometimes configuration or customization is possible both globally, and also for projects (or sometimes project groups). Initially, there will be no configuration file for the more granular scope. In this case, you can download the global configuration file, modify it for the scope in which it applies, and upload the modified file to the repository URL applicable for the scope, or, in some cases, to the same URL but with the file renamed in a way that Polarion recognizes as applying to the more granular scope. You will find help texts in the GUI screen where you access to the configuration file which indicate where you need to upload the modified file, and/or how the modified file must be named to apply in a narrower scope.

You can create some project-scope configurations online without having to download/upload the XML configuration file. Where this capability is available, a labeled control or link is provided creating a project-scope configuration. Refer to help for individual configurations for specifics.



Customizing global configuration for a project



# Resolve problems and troubleshooting

The Polarion platform is designed to be very easy to administrate and configure. But it is a very robust system, and no system is ever perfect. If you experience any problems, we suggest you try to look into the log files first (see Log Files).

If you see that some items are missing, or do not display correctly, you check the Index and Reindex section in this guide. The index repair and rebuild procedures described there can resolve many issues.

If you are still unable to resolve some problem, please contact Polarion technical support. It will expedite matters if you can provide the following:

- A detailed description of the problem and symptoms observed
- Log files from the time the problem occurred, and also log files from any reindex operation if you may have run when attempting to fix a problem.

# Tip

On Linux you can use <code>\$POLARION\_HOME\$/bin/polarion.init tar-log</code> to gather all relevant log files.

# **Setup for Internet Explorer users**

It is required that Internet Explorer be configured to include the URL of your Polarion server in the Local Intranet security zone.

#### To configure Internet Explorer on client computers:

- 1. Log on to the client computer with the permissions necessary to change Internet Options.
- 2. Start Internet Explorer, and on the main menu choose **Tools > Internet Options**, then click the **Security** tab.
- 3. Select *Local intranet* and click **Sites**.
- 4. In the Local Intranet dialog, click **Advanced**.
- 5. In the field **Add this website to the zone**, enter the URL of your Polarion server. For example:

```
http://polarion.mydomain.com Of https://polarion.mydomain.com
```

- 6. Click the **Add** button to add your Polarion server URL to the list of sites, then close the Local Intranet dialog.
- 7. Click **OK** in the dialog until you are back to the Internet Explorer main menu.

#### Tip

Compatibility View is not supported by Polarion systems. Be sure it is disabled, otherwise users will see an error message and will not be able to log in via Internet Explorer until Compatibility View is disabled.

# **Administration topic overview**

When you have administrator permissions for the scope you are working in (Repository or a project), the **Administration** link appears in the tool view of Navigation. Click it to enter the administration interface for the current scope.

When you enter the Administration interface, the **Navigation** pane contains a tree of *Topics* - links that take you into various administration functions and provide access to configuration files and graphical interfaces (where available). Nodes displaying a > symbol are administration categories. Clicking such nodes expands them to show the available topics. Clicking a topic opens the relevant administrative function in the Content Pane.

Some configurations provide a graphical user interface (GUI) that enable you to edit the configuration visually. For example, Workflow and Enumerations have such interfaces. In all other cases, configuration requires editing the respective XML configuration file. When you select this type of topic in the **Navigation** pane, the Content Pane provides access to and information about the configuration file and help texts. Two options for editing the configuration file are also provided:

- 1. Off-line editing: Links and controls are provided for downloading the current version of the configuration file, and uploading the modified file back to the repository.
- 2. On-line editing: A basic text editor is provided that enables you to load the current version of the configuration file, edit the XML on line in the portal, and commit the changes immediately.

Each of the available topics is documented in the Administrator's Guide. Reference information about data fields, defaults, is provided in the Administration Reference.

The following topics are available in the Administration interface:

Topic	Scope	Description
Projects	Repository	Create and manage projects in the repository.
Project	Project	Settings for the currently open project.
Project Templates	Repository	Access to Project templates, including download and upload.
User Management	Repository, project	Manage users, permissions, and user roles per project or system-wide.
Work Items	Repository, project, project group (limited)	Customize Work Item types, fields, Interfaces, assignment, workflow, project planning, time management, voting.
Testing	Repository, project	Test management configuration settings.
Portal	Repository, project	Define Shortcuts.

Topic	Scope	Description
Reports	Repository, project	Configure report descriptors, work Item metrics, and various reports.
Building	Repository, project	Define build artifacts (project only) and descriptors, configure Work Item reports.
Wiki	Repository, project	Configure defaults for exported Wiki pages (header, footer, etc.).
Notifications	Repository, project	Configure how and when email notifications are sent to users.
Scheduler	Repository	Configure scheduled jobs (builds, dashboard updates, system cleanup, etc.).
Maintenance	Repository, project	Check and refresh indexes, clear system caches, and capture Thread Dumps of all Java threads.
License	Repository	License assignment tools; license usage information.
Repositories	Repository, project	Configuration of external repositories.

# Manage users and permissions

# **Overview - User Management**

The Administration interface provides access to all operations for managing Polarion ALM user accounts and permissions. An administrator with global (i.e. repository-wide) administration rights can access all user accounts in the Polarion system. An administrator with project-scope administration rights can access user accounts for users who are assigned to the project(s) for which he/she has administrative rights.

The **User Management** topic in the Navigation panel provides access to the following user management functions:

- User accounts
- User roles
- User time splitting (between projects)
- Permissions
- LDAP interfacing (auto-create users, mapping)

# **Users and Projects**

When you are first beginning to use Polarion, the process of creating projects and user accounts may see a bit like the old "chicken or the egg" question - which comes first?

When you create a new project, you have the possibility to specify a user as the project lead. But if you haven't created user accounts yet, you'll need to come back to the project to specify this. When you create a user account manually (i.e. not automatically via LDAP or user self-create configuration), you have the possibility to specify what Project(s) the user has access to. If you haven't yet defined any projects, you have to return to the user account to assign project(s) after creating those in the system.

For a new Polarion system, you will probably create more user accounts than Projects initially, so the most efficient approach is probably to define projects first and then create user accounts, assigning each user to one or more projects in the process. Just remember to go back and specify a project lead for each project.

## **Create user accounts**

There are two ways to create user accounts in the Polarion system:

- 1. An Administrator creates accounts directly in Polarion
- 2. Accounts are automatically created when new users try to log in

This section will cover both options. These assume that the password file is the desired user authentication method. It is also possible to have user accounts authenticated by an LDAP/Active Directory server, and to auto-create Polarion ALMaccounts from LDAP data. For information on LDAP interfacing, see Integrating Polarion Server with LDAP/Active Directory.

Keep in mind the following scenarios when deciding on your approach to user account creation:

- 1. You have users on the LDAP server and want to synchronize them with Polarion.
  - Use the LDAP Synchronization feature. When the users are synchronized, they will not be able to log in to Polarion until the administrator assigns them the role *user*.
- 2. Users already exist on the LDAP server and you have turned on and set up role assignments for new users.
  - When a new user logs in for the first time, a new user account is created, and the default role(s) are assigned.
- 3. You have enabled the Create Account form in the system configuration.
  - Users can create their own user account before logging in for the first time. All user credentials are written to the passwd file. The role(s) will be assigned according to the properties in the system configuration for auto-creation of users.

#### Create a new user account

This section describes the procedure for creating a new user account directly in the Polarion user interface.

You will need the following information at hand in order to create a new user account:

The new user's name (e.g. "Jean Schmidt").

- The new user's e-mail address. Although not required to create a new account, if it is not specified the user will not receive any notifications.
- 1. Log in with administrator rights (global or project scope).
- 2. Enter the Administration interface.
- 3. If you want to create a new user for a specific project, open that project. Otherwise, select **Repository** in the **Open Project or Project Group** dialog.
- 4. In the Navigation panel, select **User Management**: **Users**.
- 5. In the **Users** table, click the **Create New User** button. A form for entering new user account information appears.
- 6. Enter information in the required fields (marked with a red asterisk) and any other information you want to specify at this time.
- 7. Click the **Create** button to create the new user account.

## Warning

The **Login Name** field must be a unique value in the repository, and should not contain spaces. Acceptable characters include upper and lower case letters, numbers, dash, underscore, period, and the @ sign. This field is case-sensitive for the end user.

New users must be assigned at least one **Global role** in the **Global Roles** section or they will not be able to access any content in the portal.

Also note that login permission is only granted to users assigned a role of user. By default, a new user is assigned the role everyone. If you do not also assign a role of user, either at the repository level, or in at least one project, the new user account will be disabled, and the string [disabled] appears in red text in the user's account page when the new user account is saved. (This applies to normal users. Administrator users do not need explicit permissions assigned, as the administrator role grants them all permissions in the scope.)

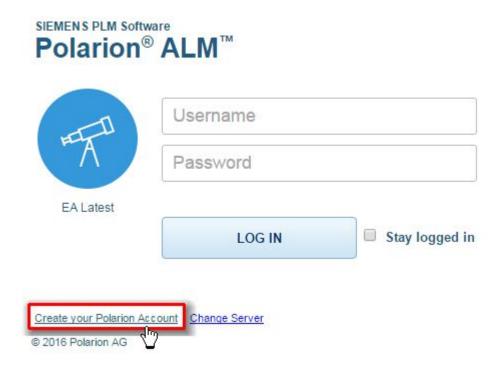
#### **Specifying a License for User Accounts:**

If multiple licenses are present on your Polarion server, the **License Type** field appears on the user account form. The field provides a drop-down list of the licenses currently available. Select the license to be used by the new user you are creating.

After pressing **Save**, the new user is added to the users file (if the type is not the defaultUserLicense type).

#### **Enable the New Account Creation form**

Administrators may optionally configure a Polarion installation to enable new users to self-create a new user account. When the system is so configured, the portal login page displays the following link:



Portal login page may allow new users to explicitly create a new account.

When users click the link, a form appears which enables the user to specify account credentials, including user name, password, and email address for receiving notifications. When the form is submitted, the login screen appears again and the user can log in with the new credentials.

In cases where more than one product license is present on the server, it is also possible to configure which license type will be used by new users who self-create an account from the portal login page.

#### **Enable the Create a New Account form:**

This configuration is performed in the system properties file polarion.properties (follow link for location). To enable the new account form, feature set the enableCreateAccountForm option to true. To specify the license (or license group) for accounts created via the new account form, use the licenseForNewUserAccount property (see comments in the properties file for information about values). You can also specify the minimum length for passwords (minimalPasswordLength option) and role(s) for new users rolesForNewUserAccount option.

#### **Configure User Auto-create**

The Auto-create feature makes it possible for new users to create user accounts by simply logging in to the portal. When Auto-create is enabled, then when a new user enters a user name and password in the portal login screen, a user account is automatically created, provided that the password supplied is valid for accessing the Subversion repository. The new user is also assigned the role(s) configured for new auto-created users. It is recommended to configure the Auto-create feature with at least a role that enables users to log in... the *user* role, for example.

- 1. Log in with administrator permissions for the repository. Enter Administration if you are not already there after login.
- 2. In Navigation, expand **User Management** and select **Auto-create**.

- 3. In the **User Management** Auto-create page, check **Enable Auto-create**. The **Global Roles** list appears. The *user* role is selected by default.
- 4. Specify the global (i.e. repository scope) role(s) to automatically apply to all auto-created user accounts.

## Warning

Be sure you understand the roles and the permissions they grant to users, and recommended best practice for role assignments. For information, see: Default Roles and Permissions.

If more than one license type is present on your server, you may want to configure the license to be assigned to new auto-created users. Specify the license in the <code>licenseForNewUserAccount</code> property in the polarion.properties file. (Windows: <code>polarion/configuration/polarion.properties</code>, Linux: <code>%POLARION\_HOME%/etc/polarion.properties</code>). See the comments for this property in the configuration file for more information. Note that if concurrent license groups are defined in the global configuration, it is possible to specify a concurrent license group in the <code>licenseForNewUserAccount</code> property rather than a license. For information on these groups, see the embedded <code>Quick Help</code> in the License topic of global Administration.

#### **Note**

For Auto-create, the user specified in the login parameter of the polarion.properties file must have write access rights for the repository folders /.polarion/user-management/users and /.polarion/security in Subversion. This is not configured in default access file. You can configure it in Administration User Management Access Management (topic available in Polarion only), or in the SVN access file directly.

#### **Enable users to reset password**

You can configure Polarion to display a link on the portal login page that leads to some page you have set up for users to reset their password. You need to set two system properties in the polarion properties system configuration file:

- com.polarion.ui.login.resetPasswordLinkURL The URL of your password reset page.
- com.polarion.ui.login.resetPasswordLinkLabel Link text to display on the portal login page.

See also: Advanced System Tuning.

#### Modify existing user accounts

This section briefly covers several common modifications an administrator may be called upon to make to user accounts.

#### Access a user account:

1. Log in as administrator and enter the **Administration** interface.

- 2. In the Navigation panel, select **User Management**: **Users** to load the **Users** table in the Working area.
- 3. In the table, select the user account you want to modify. Remember that the user account you want may not be on the first page of the Users table of there are multiple pages. If necessary, use the Search field to search for the user's name.

#### Editing a user account:

After accessing the desired account, click the Edit button in the Detail Pane of the Working Area. The following fields are modifiable:

- Full Name
- Email
- Password

#### Change a User's Password:

- 1. Access the user account as described in Access a User Account.
- 2. Open the account for editing using the **Edit** button.
- 3. Enter the new password in the **New Password** field.
- 4. Enter the new password again in the **Re-enter Password** field.
- 5. Exit the **Re-enter Password** field to make sure your two entries match and correct the fields if they do not.
- 6. Click the **Save** button to change the user's password.

#### Note

For information about passwords and password characters, see: Passwords.

### **Delete users**

Repository administrators can delete a user provided that the user is not the Assignee of any Work Items. This is the only check performed. If a deleted user is Author of any Work Item(s), the ID of the deleted user appears italicized in the Author field of those items.

If Polarion is synchronized with LDAP, and a user is deleted in Administration, then the user's LDAP account is not deleted. The user will not be able to log into Polarion anymore, but can still to access the Subversion repository (via an external client, for example). Delete the user's account on LDAP if the user should not have continued access to the repository.

- 1. Log in with administrator permissions for the repository.
- 2. Go to Global Administration (open Repository, or click Global Administration if you are in project Administration).
- 3. In Navigation, expand **User Management** and select **Users**.

- 4. In the top portion of the Users page, select the user you wish to delete, either by browsing the list, or using the **Search** field.
  - When the user is selected in the top portion of the page, the user's detail appears in the lower portion of the page.
- 5. In the toolbar of the detail portion of the page, click **Delete** and respond to the confirmation prompt. If the button is disabled, it means the user has some Work Item(s) assigned and therefore cannot be deleted.

#### Resolve users who cannot be deleted:

If you find a user that cannot be deleted, you need to locate the Work Items assigned to him/her and either unassign or reassign the items.

- 1. Note the **ID** of the user you wish to delete. Then open the Repository. If you are in **Administration**, click the link **Return to Portal**.
- 2. In Navigation, select Work Items.
- 3. In the Table view, enter the following query string into the Query Builder:

```
assignee.id:[USER ID]
```

Where [USER\_ID] is the ID of the user you wish to delete. This query will retrieve all the Work Items in the repository for which the user is an Assignee.

4. For each item, either unassign the user or reassign the item to a different user. Use Bulk Edit to select multiple Work Items and apply the change of Assignee to all selected items at once.

Run the above query again to be sure there are no Work Items remaining assigned to the user.

After completing the above steps, you can return to Repository Administration where you should now be able to delete the user as described above.

# Configure user roles

User roles control the portal content a user can access, and what level of access the user has to project artifacts and data. For example, a someone assigned a **project\_admin** role generally can make changes to project artifacts, while someone with a **project\_user** role would typically have read-only access. What a user with a given role can access is controlled by the **permissions** configuration. In Polarion, permissions are applied to roles, and roles are assigned to users.

User roles can be assigned in the repository scope, and in individual projects. Repository-scope roles control permissions of users when they log in to the repository. Project-scope roles control permissions of users when they log into a project.

- User roles are configured in the \*\*User Management \*\* \*\*Roles topic in the \*\*Administration interface.
- The level of access for each defined role is configured in User Management
   Permissions Management.

Configured roles can be assigned to users in either User Management Roles, or User Management Users.

#### **Default Roles**

Polarion comes pre-configured with several default user roles for different scopes. These are documented: Default Roles and Permissions.

#### Manage Role Definitions

You can add or delete role definitions for the repository or a project. If you add a new role definition, you will need to configure permissions for it in \*\*User Management \*\* Permissions Management: (see Configure User Permissions). If you assign a role to a user without defining permissions for the role, the user assigned to it will not be able to access anything in the repository. Also, access rights for the Subversion repository should be adjusted for the new role in \*\*User Management \*\* Access Management (topic available only when using a Polarion ALM product license), or in the SVN access file directly.

#### Add a Role definition:

- 1. Log in with administrator permissions for the scope in which you want to define the new role, and enter Administration.
- 2. If adding a repository-scope role, select **Repository** in the **Open** dialog, otherwise, select the project for which you want to add a new role definition.
- 3. In Navigation, **User Management** and select **Roles**. The Roles administration page appears, listing the roles currently configured for the scope in which you are working.
- 4. In the page toolbar, click Create New Role.
  - A new role definition form appears in the lower half of the page.
- 5. Enter a role identifier in the **ID** field. For example: **project\_tester**. Don't use spaces in the ID.
- 6. In the **Users** section of the page, select the user(s) to whom you want to assign the new role. (This is optional you can assign the role to users at a later time.)
- 7. Click the **Create** button to save the new role definition.
- 8. Open **User Management Permissions Management**, click the **By Role** tab, and select the new role you just created in the **Role** list.
- Grant all the permissions you want users who are assigned the role to have, and click Save when finished.

#### Delete a Role definition:

- 1. Log in with administrator permissions for the scope having the role definition you want to delete, and enter the Administration interface.
- 2. In Navigation, expand **User Management** and select **Roles**.

- 3. On the **Roles** page, in the table listing currently configured role definitions, select the role definition you want to delete.
- 4. Important: In the role definition detail form (lower part of the page), check the list of users who are assigned the role you are about to delete, and make sure these users have some other role assigned... at least the repository-scope user role, for example. After you delete the role definition, any users who were assigned only that role and no other will not be able to access anything in the portal.

When you are ready to proceed, click **Delete** and confirm the action when prompted.

#### Assign Roles to users

You can assign one or more roles to any user. A role definition must already exist in order for the role to be assigned to users.

It is possible to assign a user multiple roles in a project, but this is not the recommended practice. In particular, the roles **project\_admin** and **project\_user** should be used exclusively, adding only the **project\_assignable** role if the user should appear in the list of users in the Assignee field. In general, assign a role that provides the user all the permissions needed to perform his/her tasks, and no more.

For example, by default the **project\_admin** provides all the permissions of a **project\_user**, so you do not need to assign it in addition to **project\_admin**.

#### Tip

You can easily check what permissions are granted by a role assignment. Select the role name on the By Role tab in Administration User Management Permissions Management (click Expand All).

- 1. Log in with administrator permissions for the scope in which you want to make role assignments, and enter Administration.
- 2. Use the **Open Project or Project Group** dialog to select Repository, or a project.
- 3. In the Navigation panel, select **User Management A Roles**.
- 4. In the **Roles** table, select the role you want to assign to users.
- 5. In the **Users** table (lower half of the page), click **Edit**.
- 6. In the **Name** column of the Users table, select a user in the drop-down list. The selected user will be assigned the selected role.
- 7. To assign more users to this role, click the icon in the **Actions** column, and select another user in the Name list in the new table row.
- 8. When you have added all the users you want to assign the role, click **Save** to complete the assignment(s).

#### The Assignable Role:

Users must have a role of assignable and/or project assignable in order to have Work Items assigned to them.

If users report that they are missing in **Assignee** lists, an administrator should make sure such users are assigned the **Assignable** role in the appropriate scope.

#### The User Role:

By default, permission to log in to the portal is granted to users having a role of user. If a user cannot log in, check the assigned roles and make sure a repository-scope role of user is assigned. When a new user account is created, this role is not automatically assigned. You can tell if the user does not have this role if the user account page shows [disabled] in red text. See also: Create User Accounts.

# Synchronizing Roles with Repository

The Roles page of Administration provides the **Synchronize SVN Access File** button on the page toolbar. The button updates the access file of the Subversion repository, synchronizing it with the role definitions and user role assignment configurations. (Only the first part of the access file is affected, not the directory rules in the second part). It ensures that access file contains group for every user role and that group membership is in sync with role assignment.

Generally, an administrator should only need to run this synchronization if people with an assigned role cannot access repository resources as expected. This can either be from externally editing the access file, or manually changing the user-roles.xml configuration file in the repository. Run the Synchronize SVN Access File once the manual file modifications are complete.

# Configure user permissions

Permissions are an important security issue. You should plan to spend some time with this topic so that you can be sure to configure permissions appropriately.

Permissions can be configured globally for each repository, and for each project. Projects inherit permissions from the repository scope, and changes to the project settings override the same settings in the repository scope.

Permissions are tied to roles rather than to users. Users are assigned one or more roles and these govern what they have permission to access and do.

# Warning

You should configure roles before you configure permissions

### **Note**

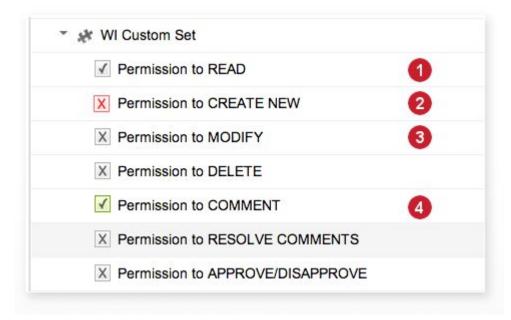
Granting permissions for Polarion does not automatically grant repository permissions. Permission to create projects, edit projects or modify administration in Polarion does not open the required repository access for those actions. You need to make sure your Subversion access permissions are configured so that Polarion permissions do not conflict with repository access permissions. For information, see Repository Access Management.

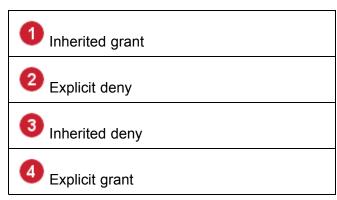
#### To access the permissions configuration:

- 1. Log in with administrator permissions for the scope you want to configure (repository or project).
- 2. If you are not in **Administration** when you log in, enter the Administration interface (via the **Administration** link in the tool view of Navigation).
- 3. Open the repository or project you wish to configure (see Access Projects).
- 4. In Navigation, expand **User Management**and select **Permissions Management**. The Permissions Management page loads.

You can configure permissions either by individual permission, or by role. Select the tab for the configuration approach you want to use (see next sections for details).

When reviewing and editing permissions, keep in mind the color scheme from the following figure:





## By Permission tab

The By Permission tab enables you to:

- View the categories of Permissions, and access individual Permissions within each category. Categories represent areas of the system for which access is controlled by permissions: projects, Documents, or Work Items, for example.
- View all of the roles to which a selected Permission is granted.
- Grant or revoke the selected Permission from one or more roles.

The top half of the page presents a tree of Permissions, organized into categories. Expand the category nodes to see the individual Permissions. For example, the Work Items category contains Permissions applicable to Work Items... permission to view, permission to modify, etc.

When you select a Permission, the lower half of the screen displays the **Applicable Roles** table. This table lists the roles to which the selected Permission may be granted, and shows which roles have actually been granted the selected Permission (by a check mark in the **Granted** column. For example, if you select Permission to READ under Work Items, you can quickly see which roles have been granted this permission.

#### To grant or revoke permission:

- 1. Select the Permission in the **By Permissions** tree.
- 2. In the toolbar of the permissions editor pane (lower half of the page), click the **Edit** button. The table is now editable.
- 3. To grant the selected Permission to a role, check the box next to the role name in the **Granted** column.

To revoke the selected permission from a role to which it is currently granted, clear the check box next to the role name in the Granted column.

4. Click **Save** to update permissions, or **Cancel** to abandon your changes.

#### By Role tab

The **By Role** tab enables you to:

- Select a role and view the Permissions currently granted to it.
- Grant or revoke Permissions for the selected role.

The By Role page presents a tree of Permissions, organized into categories. Expand the category nodes to see the individual Permissions. For example, the **Work Items** category contains Permissions applicable to Work Items... permission to view, permission to modify, etc. The Permissions show the granted/not granted status for the role currently selected in the Roles list. That list displays the roles configured for the current scope (repository or project) in Administration: User Management: Roles.

## To view the Permissions currently granted to a role:

- 1. Select the role in the **Roles** list.
- 2. Expand categories to see the Permissions. If the check box for a Permission is checked, that Permission is currently granted for the selected role.

To grant additional Permissions to the selected role, simply check the box next to any Permission when the desired role is selected in **Roles**. To revoke a Permission, clear the applicable check box.

#### **Define permissions for Artifact Sets**

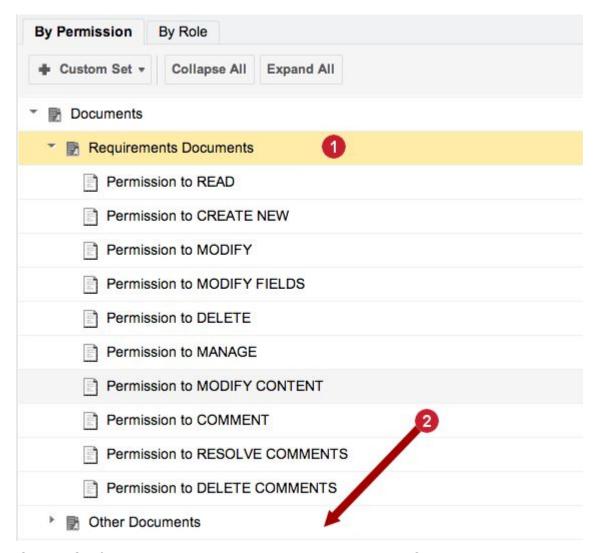
You may sometimes need to control permissions explicitly for certain specific artifacts or some attribute thereof, such as fields of Work Items. For example, a project might require explicit permissions control on some specific Work Items or Documents, which is more restrictive than the general permissions for these artifact types. You can accomplish this more granular level of permissions control with the per-dataset permissions management feature, which enables you to create <code>custom Sets</code> of permissions applied to an explicit set of artifacts.

#### You can create Custom Sets for:

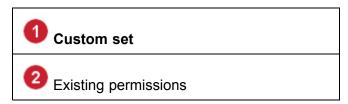
- Work Items
- Documents
- Pages
- Classic Wiki Pages
- Plans
- Test Runs

A **Custom Set** is for one artifact type only. That is, you cannot have a Custom Set that controls permissions for some Wiki pages and some Documents. Each **Custom Set** controls permissions for a set of artifacts that match a query, which you specify when creating the Custom Set. For example, suppose you have permissions defined for Documents in general, but a project contains some Documents that are typed as "Customer Requirements", and management wants more restricted access for these than for Documents in general. You can create a Custom Set containing a query that retrieves these types of Documents as a set, and then define permissions for that set which will override the less restrictive permissions on Documents in general.

When you create a Custom Set, it appears in the relevant artifact category when you expand it in the permissions editor. For example, if you create a Custom Set for Documents named "Customer Requirements Documents", it will appear when you expand the **Documents** category in either of the tabs. Another entry, "Other Documents" will be automatically created, which contains the pre-existing permissions for Documents.



Custom Set for Documents. Existing permissions moved to "Other Documents".



Custom sets can be created both in the Global configuration and per Project. Global-level permissions, together with its Custom Sets, are propagated to the Project scope if the Project does not have any Project-scope permissions set. If a Project has at least one chenge in the permissions (compared to the Global permissions), then newly created Custom Sets in the Global scope do not automatically propagate to the Project.

If you create a new **Custom Set** in a Project that is inheriting Global permissions that include a Custom Set, then the update saves both Custom Sets, i.e. the new set plus a copy of the Global **Custom Set.** 

If a Project inherits a globally-defined Custom set, then changes to it done in the Project scope never propagate to the Global configuration. Instead, a Project-scope configuration is created and used.

If an artifact matches the query in more than one Custom Set in a given scope, then its permissions are controlled by the *first* Custom Set in the permissions category for its artifact type. For example, if a Document matches the query in "Custom Set 1" and "Custom Set 3" in the **Documents** category, its permissions will be controlled by "Custom Set 1".

When querying a Custom Set, it is possible to use query expanders (that is, query elements such as SQL, linkedWorkItems, PLAN, TEST\_RECORDS that are decomposed to list of IDs in the background) in queries. When using these, care must be taken that user has the necessary permissions for all data accessed by the query. It is also possible to use the variable <code>\$[user.id]</code> in a Custom Set query. However, for simple cases of assignee and author, it is recommended to use dynamic roles (author and assignee for Work Items, document\_author for Documents, comment author for Comments and Document Comments, and lead for Project).

#### Create a custom set:

- 1. Open the project you want to configure and enter **Administration**.
- 2. In Navigation, expand ... User Management and select ... Permissions Management.
- 3. Select the **By Permission** tab if it is not the current tab.
- 4. Click the **Custom Set** button (see figure in previous section), and select an artifact type from the drop-down menu.
- 5. In the **Add Work Items Custom Set** dialog, enter a name for the new Custom Set in the **Title** field. Then enter the Lucene syntax for a query that will retrieve the artifacts to be regulated by this Custom Set (see Query Tip, below).
- 6. Click **Create** to finish the operation and create the new Custom Set.

You can now expand the new Custom Set and set permissions in the same way you set permissions for the artifact type, as described earlier in this chapter. Only items that match the query will be affected.

#### Warning

**Query Tip** You need to be sure the query you enter will access all the items that should have their permissions regulated by the **Custom Set**. One way to ensure this is to go to the Work Items topic in the project and use the Query Builder in the Table view to construct the query. You can browse the results and fine-tune the query as needed. Once you are sure you have the correct set of artifacts, you can click the drop-down icon on the Query Builder and select **Convert to Text**. You can then copy the text of the query to your clipboard, and paste it to the **Query** field of your Custom Set.

#### Queries for permission custom sets:

You can also query for Document, Page, Plan, Test Run types and other attributes in permission custom sets.

Let's say that you want to search for a document so that you can assign permissions to it for external users.

id:DOCNAME1 OR DOCNAME2

If you wanted to refine your search to a particular document type you'd use: type: [document type]

# For example:

type:generic

# The following parameters will also work in a query:

attachments	attachments.author.id
attachments.author.name	attachments.fileName
attachments.title	author.id
author.name	branchedFrom.id
comments.text	created
derivedFrom.id	derivedFrom.location
externalWorkItems	id
(Referenced Work Items)	
moduleLocation	moduleName
outlineNumbering	project.id
space.id	status
type	updated
updatedBy.id	updatedBy.name

#### Manage Custom Sets:

When you select a Custom Set, its properties are available in the lower pane of the Permissions Management page. Here you can:

- Edit the name of the **Custom Set**.
- Modify the query string for the **Custom Set**.
- Move the Custom Set up or down in the list of Custom Sets for the artifact type. This is only available if more than one **Custom Set** exists for the type. (Position in the list can be important. If an item matches the query of more than one Custom Set, its permissions will be controlled by the first **Custom Set** in the list.)
- Delete the current **Custom Set**. After deletion, the permissions of items revert to the general permissions defined for the artifact type in the relevant scope.

#### Set permissions for Work Item fields

Administrators can configure the READ and MODIFY permission for most Work Item fields to grant or deny read and/or modify permissions to different user roles. For example, if local laws prohibit

employers from revealing how much time an employee has worked to other employees, an administrator can deny read permission for the Time Spent field to appropriate user roles.

Work Item field permissions are located in: Administration: User Management > Permissions Management: Work Items > Fields. As with other permissions, you can access the specific permission via a listing of the permissions (**By Permission** tab) or by selection of a user role (**By Role** tab).

Note that denying permission to READ also denies permission to MODIFY. Keep in mind that denying one or both these permissions may impact users' ability to perform such tasks as creating new Work Items, reusing Work Items, and importing Work Items. See notes in the following sections for more specific information.

#### **READ** field permission:

When configuring Work Item field permissions, keep in mind the following points:

- If fields **severity**, **priority** and any other fields required for creating new Work Items are read-denied due to the permissions configuration, then affected users cannot duplicate or move Work Items or reuse Documents containing Work Items.
- In the LivePlan, if a user has only items in which READ permission is denied in one of the fields listed in the plan, then no items for that user will appear in the LivePlan.
- For import/export and round-trip operations:
  - o The value of READ-denied Work Item fields are not shown in exported files when a user who is denied the permission exports the item(s).
  - o If a user re-importing a round-trip document is denied READ permission for any field(s), the READ-denied fields are ignored by the round-trip importer.
  - o **Important**: If a user who is denied READ permission for some fields exports Work Items for round-trip, and round-trip re-import is subsequently performed by an different user who is *not* denied READ permission (admin user, for example), then all changed field values in the re-imported file are imported and replace the respective current values in the portal.
  - o If any import/export operation cannot be completed because a user is denied some field-level READ permission, the operation will fail with a message citing limited permissions as the reason.

For a listing of Work Item fields having the READ permission, see Fields With READ Permission.

#### Important permission ramifications

It is important for administrators configuring permissions to understand several ramifications and potential user impacts that may not be readily apparent.

- To be able to modify Work Items in a Document, users must be granted:
  - Permission to MODIFY fields Description and Title.
  - Permission to MODIFY CONTENT for Documents (including Documents in a Document Custom Set).

- By default, users can delete Heading type Work Items without having the permission to DELETE Work Items. To require that permission for deleting Headings, set the system property headingsObeyDeletePermission=true in the system configuration file polarion.properties.
- If Test Steps are displayed in a Document and they are not editable, but Title and Description fields are editable, then user can modify Work Items in the Document, but not the Test Steps field.
- If a user is denied the permission to READ the Attachments Work Item field, then they also won't be able to view images in the **Description** and other rich text fields. (All images within a Work Item are considered attachments.) .

#### **Permissions for Test Runs**

The adding and resolving of comments on Test Runs are controlled by the two permissions: "Permission to COMMENT" and "Permission to RESOLVE COMMENTS". By default, users with a Reviewer, Pro, or Requirements license can add and resolve comments on Test Runs, if their role allows it. If this is not desired, you should explicitly deny these permissions for any roles that you do not wish to allow to have them.

The comment permission is a child of "Permission to MODIFY". If you grant that permission to some role that does not currently have it, the "Permission to COMMENT" is granted as well unless you explicitly deny it.

# **Configure user Time-splitting**

In some cases, users may regularly split their working time between two or more projects (i.e. time sharing). The Live Plan project planning engine can account for this and factor such splits into the live project plan, provided the user's split time between projects is configured in his/her user account. This section explains how to configure user time splitting between projects.

#### Accessing the Time-split configuration

# ALM PRO

- 1. While in the Administration interface, select **Repository**, or any project in the **Open Project** or Project Group dialog.
- 2. In the Navigation panel, expand **User Management** and select **Users**. The table of users for the selection appears in the top pane of the Content Pane.
- 3. Scroll or use Search to locate the user whose time is to be split between projects, and select the relevant row in the table. The user's detail appears in the lower part of the Content Pane.
  - In the table, the **Time-split Assignments** column displays the current split time assignments, if any, for the selected user.
- 4. Click the **Edit** button in the user detail pane to place the user detail form in edit mode.
- 5. Scroll down, if necessary, and locate the **Time-split Assignments** section. This is where you assign the percentage of the user's time for each project.

#### Assigning split work time

Each row in the **Time-split Assignments** section lists a project for which the currently selected user has Work Items assigned. (You cannot split a user's work time into a project for which the user has no assigned Work Items.) You use the **% of Total Time** column to assign of percentage of the user's total working time to two or more projects.

#### Warning

The values you enter in % of Total Time must be integer values (and you do not need to enter the % sign). For example: 25 is a valid entry, while 33.3 is invalid and will result in an error message when you save changes.

It is not possible to save the configuration if the sum of all time-split assignments is more than 100.

The configuration splits the user's total working time between the different projects. That time is a combination of the Global Working Calendar configuration and the selected user's Personal Working Calendar configuration.

#### **Effect on Live Plan**

When a user has working time shared between two or more projects, this is reflected in the Live Plan chart. In the Repository and Project Group scope, the chart shows additional lines for a user with time splitting configured, if there are Work Items assigned to user in the configured projects. The lines are labeled by the user name followed by the configured percent in parenthesis. For example: Jan Almsman (20%). If the sum of configured time-split assignments is less than 100, then an extra line is shown for the remaining percentage if there are Work Items assigned to user in other projects for which for which no time-split assignment is specified in the configuration. The estimate of the Work Items for planning is multiplied by the ratio of the configured percentage. (So 20% means then the estimate is multiplied by 5.)

In the project scope, one line for the user is shown if there are items assigned to this user and there is some percent of time-split configured. If the total time splitting configured for specified projects is 100, and there are unresolved items assigned to the user in other projects, those items are not planned, nor are they displayed in Live Plan, and the warning is logged.

Planning of the Work Items from the different **rows** in the Live Plan which belong to the same user is independent (i.e., planning engine does not compare priorities etc.) except for items with a **depends on** link.

# **Integrating Polarion server with LDAP/Active Directory**

LDAP can be used to verify user credentials (instead of a password file in a stand-alone setup). The access file is used to define the user groups and access rights to individual repository locations. User accounts can be either manually created in Polarion (with user names corresponding to LDAP) or auto-created by Polarion based on existing LDAP users. (All necessary software for LDAP - including Apache modules - is bundled with the Polarion distribution for Windows.) This section explains the basics of integrating LDAP/Active Directory with the Polarion server.

User credentials can be authenticated against the Subversion password file or against an LDAP server. The default configuration supports both local and LDAP users.

## Warning

If you want to introduce LDAP-only authentication in an existing Polarion installation where existing users are currently authenticated against a password file, it is necessary to delete the users from the password file and have them re-authenticate via LDAP. This is necessary only if you want to replace password file authentication with LDAP authentication. (It is possible to configure Polarion to use either method.)

In a new installation, users are authorized using the Subversion integrated policy access functions (directives AuthzsvnAccessFile and AuthuserFile in the polarionsvn.conf file). If you have LDAP infrastructure, you can make Polarion authorize users against the LDAP database. Information on performing this configuration, together with some examples, is provided in the polarionSVN.conf configuration file. The file is located at: [POLARION\_HOME]\bundled\apache\conf\extra\polarionSVN.conf (Windows), OR /etc/apache2/conf.d/polarionSVN.conf Or /etc/httpd/conf.d/polarionSVN.conf (Linux, depending on distro). After modifying the configuration file, the Apache server must be restarted to reflect the changes. For more information about the Apache LDAP modules and their capabilities, visit these web pages:

- http://httpd.apache.org/docs/2.2/mod/mod\_authnz\_ldap.html
- http://httpd.apache.org/docs/2.2/mod/mod ldap.html

#### **Authentication Failover**

The possibility for authentication failover or fall-back is provided by Apache HTTP server. For example, Apache can be configured to switch to LDAP authentication if a user login is not matched in the passwd file, and to fall back to a secondary LDAP server if the primary server is offline. Information on this configuration is provided in Apache documentation (see http://httpd.apache.org/docs/2.2/mod/mod\_authnz\_ldap.html#authldapurl). Pay particular attention to the host:port setting.

#### From the Apache documentation:

"The name/port of the Idap server (defaults to localhost:389 for Idap, and localhost:636 for Idaps). To specify multiple, redundant LDAP servers, just list all servers, separated by spaces. mod authnz Idap will try connecting to each server in turn, until it makes a successful connection."

"Once a connection has been made to a server, that connection remains active for the life of the httpd process, or until the LDAP server goes down."

"If the LDAP server goes down and breaks an existing connection, mod\_authnz\_Idap will attempt to re-connect, starting with the primary server, and trying each redundant server in turn. Note that this is different than a true round-robin search."

#### Configure Polarion for LDAP user synchronization

If you just want to have Subversion (and Polarion) authenticate user credentials against LDAP instead of a password file then you can skip this section. However, if you want to configure Polarion to auto-create users from LDAP (as described in Configuring Auto-creation of Users) or to synchronize the users defined in LDAP with Polarion users (as described in Synchronization of Polarion users with LDAP), then you need to configure Polarion as well.

The **LDAP Configuration** page of Administration enables you to activate LDAP synchronization and configure Polarion to work with your LDAP server. When LDAP is enabled and configured, you can also explicitly invoke synchronization from this page.

#### To access the configuration page:

- 1. Log in with administrator permissions for the repository. If you have a clustered multi-server environment, you need these permissions for the repository of any server instance you want to configure for LDAP synchronization.
- 2. Go to Global Administration if you're in a project, or Administration if you're working in the Default Repository.
- 3. In Navigation, expand **User Management** and select **LDAP Configuration**.

# Edit the configuration:

The **LDAP Configuration** page has several sections. The settings for configuring Polarion to work with your LDAP server are in the "*LDAP Server Connection Settings*" section. The settings are disabled until you check the **Enable LDAP Synchronization** box in the **User Synchronization with LDAP** section.

The rest of the settings are information about the LDAP server... the host URL, etc. If you are not the administrator for your organizations LDAP server, you may need to consult with that administrator to obtain the information needed for these settings. Help text explaining the purpose of each of the settings is embedded in the **LDAP Configuration** page.

## **Explicitly Invoking Synchronization:**

After configuring Polarion to work with your LDAP server, you can explicitly invoke synchronization. The LDAP Configuration page has several sections. The controls for explicit synchronization are at the top of the page. There are 2 options for synchronization:

- Create New Users: When checked, the synchronization operation will create new Polarion user
  accounts for new LDAP-registered users added since the last synchronization operation. The
  LDAP configuration specifies parameters controlling which groups, etc. are included in the
  operation.
- Update Existing Users: When checked, the Polarion accounts of existing users will be updated
  with any changes for the same users on the LDAP server. For example, if a user's email address
  has changed in the user's LDAP record, the user's Polarion account will be updated with the new
  address during the synchronization.

After selection the desired option(s), invoke the synchronization using the **Synchronize** button, which is enabled when any of the options are checked.

#### **Configure the Default License:**

If more than one license type is present on your Polarion server, you may want to configure the license to be assigned to new users. Specify the license in the <code>licenseForNewUserAccount</code> property in the polarion.properties file. (Windows: <code>polarion/configuration/polarion.properties</code>, Linux: <code>%POLARION\_HOME%/etc/polarion.properties</code>). See the comments for this property in the configuration file for more information. Note that if concurrent license groups are defined in the global configuration, it is possible to specify a concurrent license group in the <code>licenseForNewUserAccount</code> property rather

than a license. For information on these groups, see the embedded "Quick Help" in the License topic of global Administration.

#### Configure the Default user Role(s):

New users created via LDAP synchronization are assigned a default Polarion user role according to the setting in the system property rolesForNewUserAccount. The default role is user, which has minimal permissions and allows a new user to log in. You can decide to specify a different default role, or multiple default roles. Comments for this property in the system configuration file polarion, properties provide details about how to set the value.

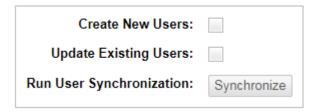
The polarion.properties file resides on the server's file system (see System Properties File Location). You can open and edit it using a text editor application.

### Synchronization of Polarion users with LDAP

When LDAP-only authentication is set up you cannot create new users using Polarion. Instead, you should create new users in the LDAP server and then synchronize the Polarion user scheme with LDAP. However you can use the user synchronization action regardless of what Subversion authentication type is used; it is just required that the synchronization is enabled and properly configured (as described in Configure Polarion for LDAP User Synchronization).

## To begin the synchronization process:

1. Navigate to Administration → **Solution Solution Solution Solution** The following Synchronize Userld with LDAP options appear:



- 2. Tick the Create New Users box to synchronize users on the configured LDAP server with the Polarion user scheme.
- 3. If you tick the **Update Existing Users** box, then existing user fields will also be updated, and existing user data will be overwritten by LDAP data.

Note	
The list of user fields is taken from the ldap-config.xml file.	

4. Click **Synchronize**. An **Info** dialog box displays the synchronization results.

The Auto-create feature is independent of use of an LDAP server. (It can be used with ordinary Subversion authentication as well.) However if LDAP user synchronization is enabled (see Configure Polarion for LDAP User Synchronization), then a newly created user is synchronized with information provided by the LDAP server.

#### To access the auto-create configuration file:

- 1. Click Administration.
- 2. Select the **Repository** scope.
- 3. Use the link in the **Configuration** section (Content Pane) to download a local copy of the autocreate-config.xml file.
- 4. Click User Management Auto Create.
- 5. Click on Navigate to Repository Browser on the right to access the ldap-config.xml file.
- 6. Click (Download icon) to download a local copy of the ldap-config.xml file, and then open it in a text editor.
- 7. If <auto-config> <enabled> is set to true, then Auto-create is enabled.

```
Info
                      Date: 2018-10-17 09:41 Author: Filip Hyr
  Revision: 1 211 872
                                                               Size: 190
  Comment: Polarion commit Wed Oct 17 09:41:03 CEST 2018 [03e4b5a0622a420388026cfbe000c168]
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2
  <autocreate-config>
3
       <enabled>true</enabled>
4
       ≺globalRolesX
5
           <role name="user"/>
       </globalRoles>
6
7
8 </autocreate-config>
```

- 8. (Optional) You may also want to adjust the list of roles that are assigned to auso-created users in the <role> elements contained within the <global-roles> element.
- 9. When you've finished configuring the file, save it and click (Upload icon) to commit your updated file the the repository.

# **Network-authenticated Users (SSO)**

Polarion supports single sign-on (SSO) for users authenticated on the network. For information, please see: Configure Single Sign-on (SSO).

# **Configure the User Accounts Vault**

Some features need to have a user name and a password specified in the configuration in order for the feature to work properly. For example, a scheduled job that imports automated test results needs credentials of a user on whose behalf it creates the Test Cases and Test Runs. For security reasons, administrators may not want certain user credentials stored in the underlying configuration file in the file system.

The User Accounts Vault provides a secure way to store user credentials, and enables administrators to reference them in a secure way in configurations. For each user account that needs to have its credentials specified in configurations, you create a Key and associate the user name and password

with it. Then in configurations, instead of supplying the concrete user name and password, you supply the Key value from the User Account Vault. Polarion automatically resolves Key references when necessary.

The User Account Vault can be configured in the global (Repository) scope only. To access the configuration:

- 1. Log in with administrator permissions for the repository.
- 2. Open the Repository and enter Administration.
- 3. Expand User Management and select User Account Vault.

The User Account Vault page presents a table of currently configured Keys. You can add new Keys, edit existing Keys (e.g. change the password), or delete existing Keys. Before deleting any Key, be sure it is not used in any configuration, especially in scheduled jobs, which will fail if a nonexistent Key is specified in the job configuration.

### Tip

It is not recommended to create Keys for credentials people who use Polarion. Rather, keys should refer to special user accounts created for automated data access.

For example there is a user "polarion" which is the system user. This user needs to be specified for jobs like the one cited above for test result imports. You should create a Key for this user in the vault and specify that Key in all configurations requiring system user credentials, rather than the actual user name and password.

# **Create and manage projects**

# Create a new project

In order to create new projects you must be granted that permission by your administrator. There are two basic scenarios for creating a project:

- Create a completely new project from a Project Template. This creates a new folder in the repository as the project root.
- Mark an existing repository folder as a project. In this way you can begin managing existing sources and other content with Polarion. This option requires permission to access global Administration.

#### Create a new Project from a template

Procedures for this are covered in: Create a New Project From a Template.

## Tip

If you are in Administration, open the default repository, then in Navigation select **Projects**, and use the **New Project** button on the Projects page of Administration to launch the new project dialog. The rest of the process is the same as described in the User Guide.

# Mark an existing repository folder as a Project

If you have an existing folder structure in your Subversion repository that you want to start managing with Polarion, you can do so by specifying the root folder and telling Polarion to "mark" it as a Polarion project.

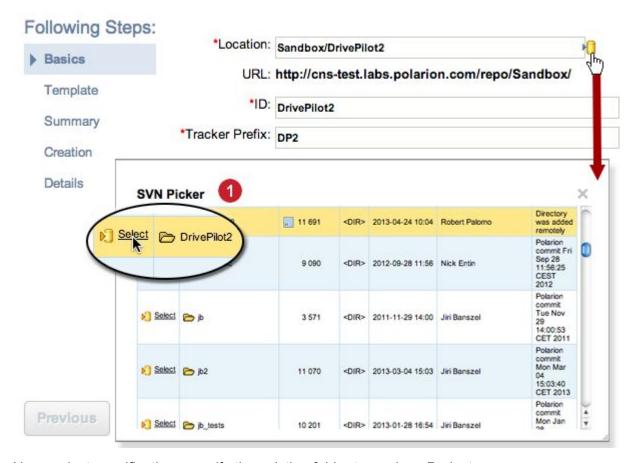
To create a new Project from an existing repository folder tree:

- 1. Log into Polarion and click the **Administration** link in the tool view of Navigation. Recall that you must have global Administrator permissions to access the Administration interface.)
- 2. In the Open Project or Project Group dialog, click Repository.
- 3. In the **Navigation** pane, select the **Projects** topic. A table of currently defined Projects loads in the Content Pane.
- 4. Click the Mark Project button located on the toolbar above the table of projects.

The new project wizard appears titled Mark existing location as Project.

# Mark existing location as Project





New project specification: specify the existing folder to mark as Project.

- ( Existing repository folder.)
- 5. The repository root URL appears beneath the **Location** field. In the edit box, type the repository path to the folder you want to mark as being a Project folder. Use forward slashes (/) to separate subfolders.

Alternatively, you can click the icon next to the edit field and use the SVN Picker dialog to browse the repository and select the folder that is to be marked as the new project root folder.

# Warning

- If you're re-marking an existing project, don't change the Project ID or you'll lose its History.
- You can't use special characters (/\@:\*?"\|;`~'# etc.) in the **Location** field.

 After selecting an existing SVN folder as a project root, you can proceed with the rest of the wizard steps to create and configure the new project. For information on the steps see: Create New Project Dialog.

# **Modify existing Project properties**

You may sometimes need to review existing projects and make changes to a project's properties. For example, if a project is no longer active, you may want to set its status to <code>Inactive</code>. Or if your organization utilizes Work Records, you may need to periodically set a date for locking Work Records created in a project.

#### **Accessing the Projects Table**

In the Administration interface, you can easily access a table listing all the Projects in Polarion's Subversion repository. You can select any Project in this table to see its properties, and you can modify those properties for which modification is allowed.

To access the Projects table:

- 1. Click on the **Repository** node of the **Open Project or Project Group** dialog.
- 2. In the **Navigation** pane and click the **Projects** node.

Table is empty if there are no projects in the repository.

# **Modify Project Details**

When you select a Project in the Projects table, its details appear in read-only mode in the lower pane of the Content Pane. You can edit all detail fields except **ID** and **SVN URL**(Subversion repository URL) after clicking the **Edit** button in the editor pane toolbar.

# **Add and Remove Projects from Polarion management**

You can have projects in your repository that are not currently managed by Polarion: projects not currently under active development, for example. Removing such projects from management by Polarion does not remove any artifacts from the repository. If you want to restore a removed project (if development resumes, for example), you can do so.

#### Removing a Project

To remove a Project from Polarion management:

- 1. Log in to Polarion with global administrator permissions
- 2. In the **Open Project or Project Group** dialog, select the **Repository** node.

(See: Access Projects.)

3. In the **Navigation** pane, select **Projects**.

The **Projects** table appears in the Projects page listing all projects currently being managed with Polarion.

4. In the **Projects** table, select the Project you want to stop managing with Polarion.

5. Click the **Unmark** and respond affirmatively to the confirmation prompt. This will stop the project from appearing in the portal, but files remain in the repository.

Alternatively, you can click the **Delete** button which will remove the project from the portal and delete the files from the repository.

# **Moving Projects**

There may be time when you want to move the location of a project within the repository. For example, the project might be finished, and you want to archive it and keep it out of the list of projects still in progress.

Polarion provides the possibility to move projects to another location in the repository. Moving preserves the project history. There are a couple of caveats to keep in mind when moving projects for any reason:

- The target location must not have been occupied by any other project in the past.
- Future new projects cannot reuse the project ID of any previously existing project in the repository, even if the location has been unmarked as a project.

To move a project to a different location:

- 1. Log in to Polarion with global administrator permissions
- 2. In the Open Project or Project Group dialog, select the Repository node. (See: Access Projects.)
- 3. In the Navigation pane, select Projects.

The **Projects** table appears in the Projects page listing all projects currently being managed with Polarion.

- 4. In the **Projects** table, select the Project you want to move.
- 5. Click the **Move**. The **Move existing project to new location** dialog appears.
- 6. In the **Location** field, enter the target location. You can click the icon next to the field to launch the SVN Picker dialog and use it to navigate to and select the target location where you want the project to be moved.
- 7. Click **Next** to begin the move operation. The **Moving** screen appears showing a progress indicator, and a message appears when the move is finished.

#### **Note**

- The settings for the project in the Subversion access file are not updated by the process. You will need to be sure to change these manually.
- The project will be unavailable to users during the move. If the project is large, the
  move could take several minutes. It is advisable to alert users about the move ahead
  of time.
- You are prevented from renaming a project's ID when moving it, so that its History is not lost.
- You can't use special characters (\\@:\*?"\`|;`~'# etc.) in the **Location** field.

# **Customize Project templates**

Projects are based on a project template which provides configuration settings and/or content targeted a some specific needs or requirements. Each Polarion product comes with one or more project templates from which administrators instantiate new projects. However, you can create additional templates using the default templates as a starting point. This section provides some necessary background and explains the basic process of creating a custom project template.

#### Tip

You can find more project templates on the Polarion ALM extensions portal at <a href="http://extensions.polarion.com">http://extensions.polarion.com</a>. You can create customized project templates based on these as well.

#### Why create Project Templates?

Like other kinds of templates you are familiar with, project templates help save time creating and configuring projects that are fundamentally the same. A template makes sense whenever it is beneficial to have some pre-defined configurations and/or content that can be reused for multiple new projects. For example, you might create a project template to be used for all new projects created for a specific team in which all project users and user roles are already defined in the template. Or you might create projects that already contain base requirements which are then added to in individual projects.

## What's in a Project Template?

A project template can contain any content that a project can contain. Here are some common things it could make sense to include:

- Project repository structure (for example, trunk/branches/tags)
- Project-level configuration: custom fields, categories, enumerations including Work Item types, workflows, notifications scheme, wiki pages, Documents, even user project roles if specific users should be assigned the same roles in all instantiated projects.

Any general resources or their templates. For example, additional empty folders, outlines and/or structures of Documents or wiki pages, etc. It is also possible to include Work Items in a project template, but that could potentially cause problems with conflicting IDs as a template gets used for creating multiple projects.

### **Legacy Project Templates**

Prior to version 3.3.0 of the Polarion platform, default project templates shipped with Polarion were located in /.polarion/projects/templates/ in the repository, and all subdirectories in this location were considered as individual project templates. As of version 3.3.0, Polarion project templates are part of the binary distribution as plugins. Any custom project templates you may have created in prior versions remain in the repository in the same location and users can still instantiate projects based on them. If you want to create new project templates based on these, however, it is recommended to port them to the newer template implementation.

#### **Access Project Templates**

The Administration interface provides access to the project template(s) available with the license(s) for your Polarion portal. The administration page is available in the repository scope only. Via the administration page you can:

- View the Polarion project templates available on your system, and any custom project templates that have been developed and committed to the repository. You can filter the view to include all available templates, only **Polarion Templates**, only or Custom Templates. The Polarion Templates link lets you view the Project Templates page on Polarion's extensions portal (in a new browser instance).
- View the properties of any project template, including the ID, name, description, and license(s) required for use.
- Create a duplicate of any project template listed on the page (either Polarion Templates or Custom Templates).
- Download a selected project template to your local drive for customization.
- Upload a customized project template to the repository, making a new template available under Custom Templates.

To access Project Templates administration:

- 1. Log in with administrator permissions for the repository and enter the Administration interface using the **Administration** link in the tool view of Navigation.
- 2. Open the repository in which you want to access project templates (see: Access Projects).
- 3. In Navigation, select Project Templates.

#### **View Template Properties**

You can view the properties of any project in the Project Templates administration page. This is useful if you are planning to create custom templates because each project template in the system must have a unique ID. By viewing properties of existing project templates, you can ensure that you do not give any custom project template a duplicate ID.

To view project template properties:

- 1. Access the Project Templates page in Administration.
- 2. Select a project template on the Project Templates administration page.
- 3. In the page toolbar, click the **Properties** button. The **Project Template Properties** dialog appears, which displays the template ID as well as other properties.

#### **Duplicate a Project Template**

You can easily create a duplicate of any existing project template. This is one approach to creating a customized version of an existing template: create a duplicate, then download and customize the duplicate.

To duplicate a project template:

- 1. Access the Project Templates page in Administration.
- 2. On the Project Templates administration page, select the project template you want to duplicate. After selecting, check the properties and note the template's ID.
- 3. On the page toolbar, click the Duplicate button. The **Duplicate Project Template** dialog appears.
- 4. Enter a value in the **ID** field. Note that the ID must be unique among all project template IDs. If you enter a value that already exists as the ID for another project template, the duplication operation will fail with an error message.
- 5. Enter a value in the **Name** field and click **OK**. A new icon for the new duplicate appears in the **Custom Templates** section of the page.

Immediately after duplication, the duplicate project template appears in the New Project wizard can be used by any administrator to instantiate new projects. Of course, it will provided exactly the same configuration and content as the original template until it has been customized and committed back to the repository. Assuming you are creating a duplicate project template preparatory to customizing it, you should make sure all administrators know not to use the duplicate project template until customization is complete.

## **Download a Project Template**

To create a customized version of a project template, you need to download it to your local computer, modify it, and upload the modified version to the portal. You can download any template on the Project Templates administration page.

To download a project template for customization:

- 1. Access the Project Templates page in Administration.
- 2. On the Project Templates administration page, select the project template you want to download. After selecting, it's a good idea to check the properties and note the template's ID.
- 3. On the page toolbar, click the **Download** button. The file save dialog of your operating system appears.

4. Save the project template to any location on your local system. The template is downloaded as a Zip archive.

#### **Create a custom Project Template**

You can use any of the project templates shipped with your Polarion product, or available on the Polarion Extensions portal as the basis for custom project templates.

#### Tip

To familiarize yourself with what each of the default templates provides in terms of configuration, Documents, Pages, etc. create a Sandbox project from each of the project templates listed in the Create New Project wizard and explore them. (You can delete these projects later on once you know about them.)

When you decide which of your existing project templates you want as the basis for a custom project template, can either:

- Download an existing project template, customize the downloaded template, and upload the modified template overwriting the original. OR...
- Create a duplicate of a project template, download and customize the duplicate, and upload the modified duplicate.

After downloading a project template, you can perform customizations by editing various files on your local system. You will need a text editor application at a minimum. A editor for HTML and XML is recommended.

Each project template may contain a file named template.properties in its root. Use of this file is not required, but is recommended. The file has a standard Java properties file format. For a list of the properties that can be set, see: Project Template Properties.

#### Note

If your custom project template is large, you may find that you cannot upload it due to the attachment size limit currently set in the system configuration. In such cases, your Polarion administrator can temporarily increase the limit set in the maxAttachmentSize system property.

You can potentially use some existing project as the basis for a project template. If you do that the project should not be referencing any specific revisions - baselines, frozen Work Items or Test Runs, for example, unless you are sure that the template will always be used with the same repository. Such references can cause problems if the template is used on a different repository. Best practice is to avoid creating a template from a project that contains specific revision references.

#### Approaches to template development

There are 2 basic ways you can approach template customization:

- Edit downloaded configuration and content files belonging to a project template using a text editor, and upload a modified archive to the portal. (This option is more for advanced users who are experienced with project template structure and customization.)
- Customize an existing project in the portal a Sandbox project created for the purpose, for
  example then save a local copy of changed and added files, and upload an archive of these to
  the project template on the portal that you want to customize. (This option is probably better if
  you are just getting started with customizing project templates.)

#### **Customize local files:**

- 1. Download a copy of the project template you want to customize from the Project Templates administration page and unpack the archive.
- 2. Edit the XML files in the local copy of the project template, creating new xml files (e.g. for new Documents, wiki pages, etc.) as needed.
- 3. Edit the template.properties file, specifying a unique ID for the customized template, and an appropriate name and description.
- 4. Create a Zip archive containing all the files belonging to the template, including any files which you did not change. If you downloaded a "stock" template provided by Polarion, or a duplicate template created for the purpose of customizing it, and you want your changed version to overwrite the original version on the server, then name the Zip archive with the same filename as the archive of the template you downloaded.
  - If you want to preserve the original version of a project template you downloaded from the server and have your customizations appear as a new template, give the zip archive a filename different from that of the downloaded template.
- 5. Upload the Zip archive to the portal using the **Upload** button on the Project Templates administration page.

#### **Customize using the Portal and SVN:**

- 1. In the Polarion portal, instantiate a sandbox project based on the project template you want to customize.
- 2. Then proceed to customize that project in the portal, modifying configurations and creating Documents, Pages, etc. as needed.
- 3. Use the Repository Browser to save the project folders containing modified configuration and content files to some location on your local system. Generally, the folders involved include: .polarion, wiki, modules. You may include the template.properties file as well.
- 4. Create a Zip archive of the downloaded folders and files. Include the template.properties file, and give the archive a filename the same as the template name. The template.properties file and all other template files and folders must be placed in a folder the name of which is same as the template ID.
- 5. Upload the Zip archive containing the modified template files to the portal using the **Upload** button on the Project Templates administration page.

#### **Replace Work Item Prefix:**

There may be cases where you want your project template to provide some pre-created Work Items in projects based on it. In this case, you will probably develop your template as a project in Polarion, and create a project template from that, as previously described. When you create your template development project in Polarion, you specify a prefix for new Work Items. For example, you might specify "MYTMPL", in which case new Work Items in your template project will have IDs like MYTMPL-100, MYTMPL-101, etc.

When your project template is finished and deployed, and when users create a project based on your template, they will be able to specify a prefix for new Work Items in their new project. For example, suppose the project creator specifies "FSPECS" as the Work Item prefix for a project based on your project template. New Work Items created by project users will have IDs like FSPECS-100, but the "default" Work Items provided by your project template will not have this prefix. Rather, they will retain the original "MYTMPL" prefix. You need to make sure to replace your Work Item prefix with the user-specified prefix in user-created projects. Because not all project templates need to provide "default" Work Items, this prefix replacement is not enabled by default in custom templates.

#### Note

#### If custom templates are used:

To have Polarion replace the Work Item Prefix from template-provided Work Items, add the following lines to your template.properties file: trackerPrefixToReplace=PREFIX, where PREFIX is the Work Item prefix specified in the project template's development project. For example, if your development project's Work Item prefix is "MYTEMPL", then you should specify trackerPrefixToReplace=MYTEMPL in the properties file.

```
process=.polarion/tracker/fields/workitem-type-enum.xml
```

These lines are added automatically in templates that ship with Polarion.

#### **Control access to instantiated Projects**

You can optionally create and configure a file named access.template which, when created in the root folder of a project template, controls low-level access permissions for new projects instantiated on the project template. Rather than being copied into the instantiated project, the file content is inserted into the Subversion access file if one is configured and accessible (which is normally the case).

The following listing provides an example of an access.template file:

```
[groups]
{\project id}-project admin = {\project user}
{\project_id}-project_user =
[{$project location}]
@admin = rw
polarion = r
@{$project id}-project admin = rw
@{$project id}-project_user = r
[{$project location}/.polarion]
@{$project id}-project admin = rw
```

```
@{$project_id}-project_user = r
[{$project_location}/.polarion/tracker/workitems]
@{$project_id}-project_admin = rw
```

Pay particular attention to the references to user roles such as project\_admin and project\_user. The role identifiers in the above example are defaults in some project templates delivered with Polarion products. If custom roles have been configured in your the project template, you should reference the identifiers of those roles when configuring the access.template file.

Be aware that if no access.template is defined in a project template, Polarion will add the content of the default access.template to the access file when a new project is instantiated from the template.

### **Use external repositories (GIT, SVN)**

Polarion supports revision links with external **Subversion** or **Git** repositories in addition to the "default" Subversion repository installed and/or configured during Polarion installation. If you have existing repositories that you want to use with your Polarion installation, you will need to configure your portal or projects to work with them. You can do this in the **Repositories** topic of **Administration** in the global repository or project-specific scope.

For information and procedures, see: Configure External Repositories (Git, SVN).

# **Configure Work Items**

### **Customize Work Items and related features**

**Work Items** are a major focal point of data and activity managed by Polarion. There is of course no single definition or configuration of Work Items that would be applicable to all needs everywhere, so Polarion provides extensive possibilities to customize Work Items and functionality related to Work Items. This chapter covers these capabilities.

The **Work Items** topic of the **Administration** interface provides access to all Work Item related configurations. The sub-topics in this section describe the various Work Item configurations and settings. In online Help, use the Contents panel to navigate to specific topics.

# **Configure Work Item types**

Work Items represent various artifacts - requirements, tasks, change requests, for example. Each artifact type is represented in Polarion by a **Work Item type** defined in the global configuration, the project configuration, or both. (For projects, the default types are usually pre-defined in the project template on which the project is based.) If the default types do not meet your specific needs in terms of semantics or artifacts represented, you can customize the relevant configuration.

For example, suppose one default type for a project is **Defect**, but your internal terminology is **Issue**. You can change the name of the existing type to conform to your standard. Or suppose a given project needs to track an artifact "Supplier Part" but the project configuration has no analogous Work Item type. You can easily define a new type to represent such an artifact. If the need for the customized type(s) will extend beyond a single project, it is advisable to create or customize a project template and define the desired type(s) there. (For information see Customize Project Templates.)

You can customize the Work Item type configuration in the **Types** page of the Work Items Topic of Administration.

#### To configure Work Item types:

- 1. Log in with Administrator permissions for the scope you want to configure (Global or project).
- 2. If configuring a project, open the project. Otherwise, open the Repository.
- 3. Enter Administration, expand Work Items and select Type.

The configuration is stored in the underlying configuration file workitem-type-enum.xml. The Types page provides a visual interface for the configuration. On the page you can:

- Modify properties of existing Work Item types.
- Delete existing Work Item types.
- Define new Work Item types.

For new custom types, you can upload custom icon images. You can also specify an existing Work Item as a template for all new items of your new type. For more information see Define Template Work Items.

The workitem-type-enum.xml file is one of a number of "enumeration" configurations. It is therefore alternatively accessible via the Enumerations page ( Administration Work Items **Enumerations**: workitem-type-enum.xml). You need to access the file through this path if you want to remove an existing Work Item type configuration. For more information see Configure Enumerations.

# **Configure Time Points**

A Time Point is essentially a named milestone that has a date when it occurs. For example, you might have a Time Point named "Beta 1" that falls on some date prior to the actual end date of the project. Work Items can be associated with a Time Point so that the developers can quickly understand where a Work Item fits in the larger time picture. Time Points also appear in the project Live Plan (in products which have the feature) so that managers and others can see what items are planned to each one.

Time Points can be configured globally for the Repository, for project groups, and individual projects.

#### **Create a new Time Point**

- 1. Log in with administrative rights for the scope you want to configure.
- 2. Enter the **Administration** interface.
- 3. Open the project or repository in which you want to configure Time Points. (See: Access Projects).
- 4. In the **Navigation** panel, expand the **Work Items** node and select **Time Points**.

A table appears in the top portion of the Time Points page listing all Time Points currently defined for the current scope (if any). The Detail pane shows the properties of the current selection in the table.

- 5. Click **Create New Time Point**. A form appears where you can specify the properties of the new Time Point.
- 6. Fill in the properties of the new Time Point and click **Create**.

#### Note

#### **Time Point IDs:**

It is possible to specify the same ID for Time Points defined in different scopes. For example, you could configure a Time Point ID BETA1 for a project group ("parent"), and configure a Time Point with the same ID but a different date in a project in that same group ("child"). The Time Points in the project would then override those defined in the project group. However, **this is not recommended practice**.

If a parent Time Point is removed, or a child Time Point is created, it is necessary to clear the system caches.

#### **Edit an existing Time Point**

You may wish to edit the properties of an existing Time Point, to extend the time or mark the Time Point closed, for example.

- 1. In the \*Administration interface, open the scope in which the Time Point is configured (Repository, project group, project) and select the Time Point in the Time Points table as described in Creating a New Time Point above.
- 2. In the Time Point detail pane, change any field in place (except ID) or click **Edit** to put all fields into Edit mode.
- 3. Click **Save** to save your changes.

#### **Delete a Time Point**

You can only delete a Time Point if no Work Items are associated with it in their Time Point field.

A button labeled **Delete** appears in the Time Point page detail pane. If any Work Items are associated with the Time Point, the **Delete** button is *disabled* and you cannot delete the Time Point.

If there were ever actually a reason to delete a Time Point after Work Items are associated (and this seems unlikely), you would need to execute a query to locate all the affected Work Items, and then use **Bulk Edit** to remove the Time Point setting from all of them. You could then return to the **Administration** interface, select the Time Point as has been described, and delete it by clicking the **Delete** button which should then be enabled.

## **Configure Prioritization**

Several things relating to the prioritization of Work Items can be customized from the system defaults. These include:

- Priority Ranks
- Priority Increment

#### **Priority Ranks**

When the standard Priority field of Work Items is used as the field for prioritizing Work Items, the priority of a Work Item is a float value between 0.0 and 100.0. The higher the number, the higher the item's priority. The value is used by the LivePlan engine along with other factors to calculate where Work Items fall in the project plan.

Polarion's default configuration provides priority "ranks", which are simply textual labels representing a priority range. The default values are:

- Highest (minimum value 90)
- High (minimum value 70)
- Medium (minimum value 50)
- Low (minimum value 30)
- Lowest (minimum value 10)

Thus, an item with a Priority value of 75 has a priority rank of **High**, and an item with Priority value 49 is ranked Low, and so on.

You can change the total number of ranks (add or delete ranks), the rank labels, and the rank Minimum Value. For example, you might decide you don't need the rank Lowest, so you would delete that row in the configuration page and adjust the Minimum Value field of the other ranks.

End users see the result of this configuration in the **Priority** field of Work Items, and in the **Prioritization** sidebar of the Work Items table (used to re-prioritize multiple Work Items at once).

#### **Priority Increment**

The Priority Increment value is used by Polarion when the position of an item in the table is changed by using the Up or Down buttons in the Prioritization sidebar, by drag-and-drop, and the item ends up at the top or bottom position in the table. Such an item is assigned the numeric priority value of the item that formerly occupied the top/bottom position, plus or minus the configured increment value. The default increment value is 10. Thus, if a Work Item is moved to first position in the table, and the item that was in that position before has a priority value of 50, the moved item would get a new priority value of 60. If a different increment value is desired for such operations, system configuration can be modified to use a different Priority Increment value.

To change the default increment value, add the system property prioritizationModeIncrement to the system configuration file polarion.properties (follow link if you need the location). Set the property to an integer value. (As with any change to the system properties, the Polarion server must be restarted before the change takes effect.)

**Examples**: prioritizationModeIncrement=5 **Of** prioritizationModeIncrement=20

### **Configure Categories**

#### **Configure Categories overview**

By default, Work Items have a field called **Categories**. By assigning a Work Item to one or more Categories, you add yet another parameter for queries, reporting, and dashboards.

Categories are purely arbitrary- you can define any scheme that's appropriate to your needs. For example, your might define a **User Interface** or **Look & Feel** category for all Work Items that relate to your application's user interface. You would then be able to query for this Category, and dashboards will be able to report information such as unresolved items for the Category.

In order for Work Items to be assigned to Categories, you must first define Categories. This configuration is available only in the Project scope, as Categories only apply to individual projects and not to all projects globally.

#### **Create a new Category**

- 1. Log in with administrative rights for the Project you want to configure.
- 2. Enter Administration.
- 3. Open the project or repository you want to configure. (See: Access Projects.)
- 4. In the **Navigation** pane, expand the **Work Items** node and select **Categories**. (Note that this item only appears when working in a project.)
  - A table appears in the top section of the page listing all Categories currently defined for the Project (if any). The lower section shows the properties of the current selection in the table.
- 5. In the Content Pane toolbar, click **Create New Category**. A form appears in which you can specify the properties of the new Category.
- 6. Fill in the properties of the new Category and click **Create**.

#### **Edit an existing Category**

You may wish to edit the properties of an existing Category, to expand the description or make the name more descriptive, for example.

#### To edit an existing Time Point:

- 1. Navigate to the Project and select the Category in the Categories table as described in Create a New Category.
- 2. In the Category's detail pane, click **Edit**. The data in the Detail pane becomes editable.
- 3. Change the properties as desired. (You can edit all except ID.)
- 4. Click **Save** to save your changes.

#### **Delete a Category**

You can only delete a Category if no Work Items are associated with it in their Categories field.

A **Delete** button appears beside the **Edit** button described above. If any Work Items are associated with the Category, the **Delete** button is *disabled* and you cannot delete the Category.

If there were ever actually a reason to delete a Category after Work Items are associated (and this should happen very rarely if ever), you would need to execute a query to locate all the affected Work Items, and then use Bulk Edit to remove the particular Category from all of them. You could then

return to the \*Administration interface, select the Category as has already been described, and delete it by clicking the **Delete** button which should then be enabled.

### **Configure Custom fields**

### **Configure Custom Fields overview**

In addition to the default Work Item fields, you can configure custom fields of different types. For information on supported field types see: Custom Field Types. Custom fields can be used to store whatever data is needed to support your development process, and they can also serve as a parameter for queries, providing an additional way to search for subsets of Work Items. You can define custom fields in the global (repository) scope, or for specific projects. Within each scope, you can define custom fields that apply to all Work Item types in the selected scope, or to one particular type of Work Item... a Defect or Change Request for example.

If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface. Custom fields are defined in one of several possible configuration files, depending on whether you are defining custom fields applicable to all Work Item types, or to a specific type. The configuration files are accessible in the **Custom Fields** topic of Administration under Work Items. The configuration files can be edited online using the graphical web interface (recommended), or downloaded and edited offline using an XML editor (recommended for advanced administrators only).

Custom field definitions applicable to all Work Item types are defined in custom-fields.xml. Custom fields applicable to a specific Work Item type must be defined in a file of the same name prefixed with the ID of the applicable type. For example, a configuration file for Requirement type Work Items must be named requirement-custom-fields.xml (requirement being the type ID), for Defect type items defect-custom-fields.xml and so on.

#### Warning

#### If a new \*Required custom field is added to an existing Work Item Type:

If an existing Work Item type is amended so that a new custom field is \*Required, then Work Items of that type created BEFORE the new field was added can still be amended without entering a value for the new field. (But you won't be able to clear a value in the field if one is already there.)

If you define a custom field of the Enum: (enumeration) type, it needs to refer to an existing enumeration which provides the values that appear in the field's pick list. Before configuring this type of custom field, you should first create the enumeration to which it will refer. For example, if you want a project-scope custom field Coffeefrom which users can select from among the values espresso, americano, and latte, you would first need to create a project-scope enumeration containing these values before creating the Coffee custom field in the project configuration. See Configure Enumerations and especially the section Create Enumerations for Custom Fields.

#### Warning

Multi-value custom fields must be of the enum type.

#### Understanding the different configuration files

The <code>custom-fields.xml</code> file in the global (Repository) scope defines custom fields that are the global default for all Work Item types in all new projects. You then have the possibility to define custom fields for different Work Item types (you should define all Work Items types before defining custom fields). Configuring the different types in different scopes can seem a bit daunting for new administrators, but once you grasp the concept it's not too difficult. Here are some common scenarios, and what you would need to do in each case.

#### Scenarios:

- 1. All Work Item types, global scope
- 2. All Work Item types, project scope
- 3. Specific Work Item type, global scope
- 4. Specific Work Item type, project scope

#### All Work Item types - Global scope

You need to work with the custom-fields.xml file in global Administration.

#### To access the custom-fields.xml configuration file:

- 1. Log in with global (repository) administrator permissions.
- 2. Enter **Administration** if your are not there after login.
- 3. Select Repository in the Open Project or Project Group dialog (see: Access Projects).
- 4. In the **Navigation** panel, expand **Work Items** and select **Custom Fields**.
- 5. In the row labeled **All Types** -, click the **Edit** link in the **Actions** column to launch the **Custom Fields** Editor.
- 6. Fill in the properties of the custom field you want to create.
  - If you want to create a multi-value field in which users can select multiple values in a drop-down list of values, select **Enum**: in the Type column, and check **Multi**.
  - Check **Required** if the new custom field should be mandatory. (End users cannot save Work Items until all required fields are filled.)
- 7. To create additional custom fields, click the icon in the **Actions** column of the Custom Fields Designer.
- 8. Click **Save** to save the custom field definition(s) to the global configuration file.

#### Tip

#### Tip for advanced administrators:

You may optionally use the custom-fields.xml link in the Custom Field Definitions section of the page to download the global configuration file to your local system, and edit it with an XML editor to create the desired custom fields definitions.

After editing the file locally, you use the controls in the Upload New Global Custom Field **Definition** section to upload the modified file back to the Polarion repository.

#### All Work Item types - Project scope

Projects inherit the global configuration of custom fields. A copy of the global configuration file custom-fields.xml is created in the project repository at the time the project is created. This file defines custom fields applicable to all Work Item types. In any project, you can create a new project-scope configuration defining custom fields for all Work Item types that will apply only to the specific project. This configuration overrides the global configuration of custom fields for all Work Item types.

Once a custom fields configuration for the project has been created, it can be modified to change field types, required status, default value, etc.

#### To create a new project-specific custom fields configuration for all Work Item types:

- 1. Log in with administrator permissions for the Project you want to configure.
- 2. Click Administration in the tool view of Navigation.
- 3. Open the project you want to configure.
- 4. In the **Navigation** panel, expand **Work Items** and select **Custom Fields**.

A link "custom-fields.xml" appears in the - All Types - row of the Custom Field Definitions table. Depending on the project template used to create the project, the Type column will say Global configuration and the Actions column will be empty, or the Type column will say Project configuration and the Actions column will contain links Edit (leading to a project-specific configuration) and Remove (which removes the project-scope custom fields configuration for all Work Item types).

In the latter case, you can modify the project configuration or remove it and start over. For the purposes of this scenario, let us assume there is no project-specific configuration for custom-fields.xml.

- 5. Click Create New Configuration. The Create New Custom Field dialog appears.
- 6. In the Work Item Type list, choose All Types and click Next. The Custom Fields Designer page displays. The file name of the configuration file you are working on appears in the header of a table which provides data entry controls for each property of the new custom field you are defining.
- 7. Fill in the properties of the custom field you want to create.

If you want to create a multi-value field, select Enum: in the Type column, and check Multi.

Check **Required** if the new custom field should be mandatory. (End users cannot save Work Items until all required fields are filled.)

- 8. To create additional custom fields, click the icon in the **Actions** column of the Custom Fields Designer.
- 9. Click **Save** to save the new custom field definition(s).

#### Specific Work Item type - Global or Project scopes

In the global scope, you can configure custom fields that apply to a single Work Item type. As mentioned previously in this chapter, the global custom fields configuration is inherited by new projects.

As an example, you might want to create a field **Fixed in Build** that appears in all new projects, but only in "Defect" type Work Items.

#### To create custom fields applicable to a single Work Item type in the global scope:

- 1. Log in with administrator permissions for the repository.
- 2. Click Administration in the tool view of Navigation.
- 3. Select Repository in the Open Project or Project Group dialog (see: Access Projects).
- 4. In the **Navigation** pane, expand **Work Items** and select **Custom Fields**. The Custom Fields administration page displays a table that lists all existing custom field configurations.
  - In a new Polarion installation, there will be no Work Item type-specific custom field definitions in the global/repository configuration.
- 5. Click Create new configuration. The Create New Custom Field dialog appears.
- 6. In the **Work Item Type list**, choose the Work Item type for which to create new custom field(s), and click **Next**.
  - The Custom Fields Designer appears. You are now creating a global custom field for the Work Item type you selected in the dialog. The header of the table of custom field definitions displays a file name like [WORK-ITEM-TYPE-ID]-custom-fields.xml (where [WORK-ITEM-TYPE-ID] is the ID of the Work Item type you selected in the dialog... **defect-custom-fields.xml**, for example).
- 7. Fill in the properties of the custom field you want to create. For the Fixed in Build example, enter an ID like **fixinbuild**, and **Fixed in Build** in the **Name** column.
  - If you want to create a multi-value field, select **Enum**: in the Type column, and check **Multi**.
  - Check **Required** if the new custom field should be mandatory. (End users cannot save Work Items until all required fields are filled.)
- 8. To create additional custom fields, click the icon in the **Actions** column of the Custom Fields Designer.
- 9. Click **Save** to create the new custom field definition(s).

#### Specific Work Item type - Project scope

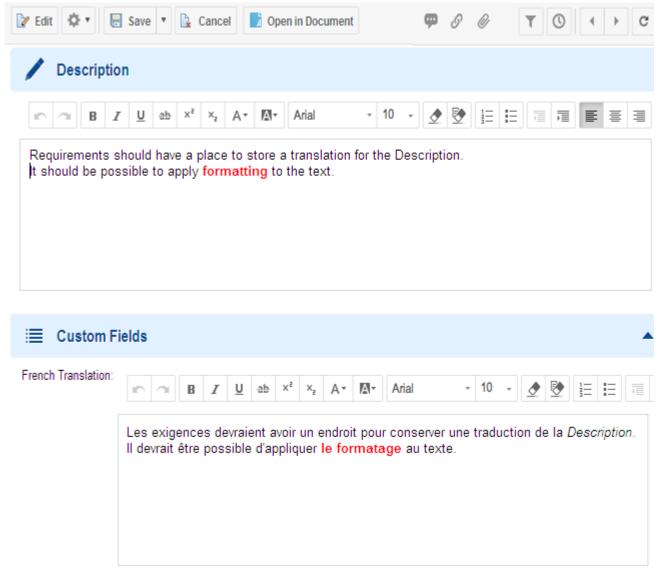
The procedure for defining custom fields applicable to a single Work Item type for a single project is identical to that just described in the previous section for a single type in the global scope, except that you should open the project you want to configure, as opposed to working in the Repository scope.

Projects based on a project template will probably have some pre-configured custom field definitions for the Work Item types configured for the project in the template. You may find that you need to modify an existing custom field(s) configuration for some Work Item type rather than create a new one.

#### **Rich-text Custom fields**

The **text/html** field type is provided for custom fields. When you configure a custom field with this type using the Rich Text (multi-line) option in the field designer, the field appears in the Custom Fields section of the Work Item form as a multi-line text field with a toolbar enabling rich-text formatting of the field content (in exactly the same way as the standard **Description** field). You can configure rich-text fields in either global or project configuration, and you can limit the configuration to a specific Work Item type.

A use case for rich text fields is to provide storage for a translation of the content of the **Description** field to another (human) language. This might be especially applicable to Requirement type Work Items. Consider an example project configuration for this use case.



#### Result for end user of rich-text custom field configuration

If the Work Item being editing is one that is stored in a Document, the Table button is disabled in the rich text custom field toolbar and the user cannot insert any tables in the rich text field.

When working in the Multi Edit view in the Work Items topic, users can show the editor for rich text custom under the editor for the **description** field by clicking the **Edit** in the Work Item's row in the display.

#### **Custom fields with dependent enumerations**

Consider this scenario: you have a custom field for Defect type Work Items named "Affected OS", of type *enum*, with enumeration values *Linux*, *Windows*, and *OS-X*. You have another custom field "OS Version", also of type *enum*, which holds different versions of the operating systems listed in the "Affected OS" enumeration.

By default, when your users choose *Linux* they would have to pick the value from a list that includes not only Linux versions, but also Windows and OS-X versions, because all these are in the "OS

Version" enumeration. The list could be large, so it would be better for users if the enumeration values shown to them in the "OS Version" field were dependent upon the user's selection in the "Affected OS" field. That is, when users pick Linux, they should only be shown values like SuSE, Ubuntuetc. in the "OS Version" field. Polarion provides a feature "Dependent enum Fields" that enables administrators to easily set up this kind of usability for end users. Note that it is only for custom fields.

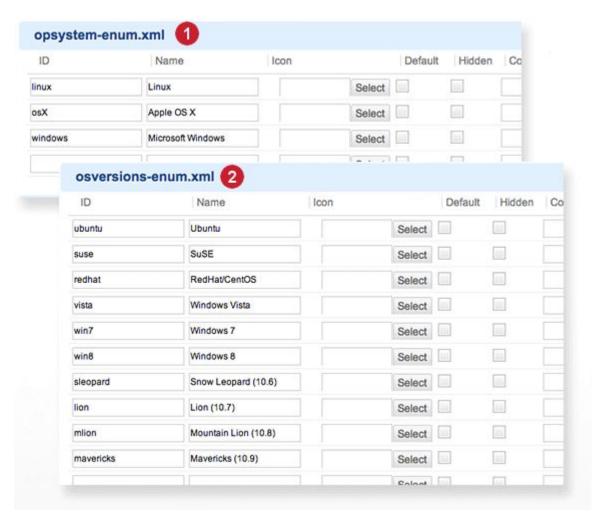
You can configure Dependent enum Fields both globally, and per-project. The process has 2 main steps:

- 1. Configure enumerations and map value dependency between them.
- 2. Configure custom fields of the enumeration types and map the selection dependency between them.

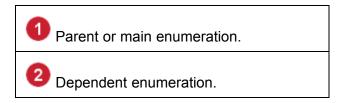
#### **Configure Dependent Enumerations**

The first step is to configure the necessary enumerations in Administration. For the example scenario above, you would define one enumeration containing the names of operating systems (let us say, opsystem-enum.xml), and another containing the versions (osversions-enum.xml). The objective here is to create a value dependency between them. The values shown to users from the latter will depend on the value selected in the former.

- 1. Open the Work Items Enumerations topic of Administration.
- 2. Define 2 enumerations and populate them with the desired values, remembering that the value selection in one is to filter the display of values from the other,



Value selection in "parent/main" filters values in "dependent"

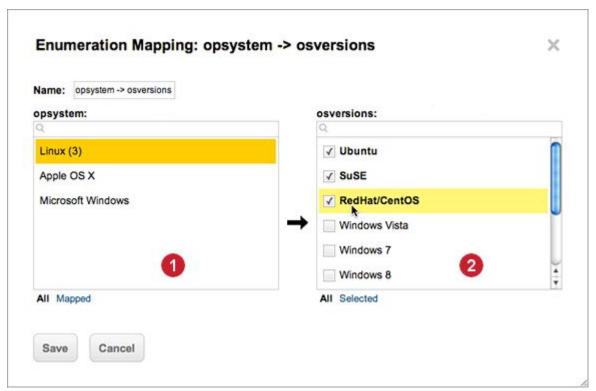


3. Map the dependency between the enumerations. Open either of the enumerations in the editor, and use one of the 2 dependency selectors to select the other enumeration, and create a mapping of values.

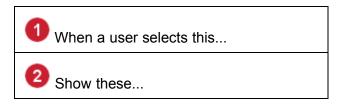
This may be easiest at first if you open the enumeration whose selection filters the values from the other one. In our example, this would be the <code>opsystem-enum.xml</code> enumeration (because when a user selects, for example "Linux", that selection should result in showing only Linux versions from the <code>osversions-enum.xml</code> enumeration). Therefore, you would go to the **Enumerations depending on this one** section and select **osversions** in the list.

4. Now click the **Add Mapping** link. In the mapping dialog, select the first value of the current enumeration in the left-hand list, and in the right-hand list select all of the dependent values that should be listed for the selected value. For example, if Linux is selected on the left, you would

select Ubuntu, SuSE and CentOS on the right, as they are all valid values for Linux. Leave other values unchecked.



Mapping values to show users in an enumerated list of OS versions



5. Repeat this mapping for all the other values in the current enumeration, selecting only valid dependent values in the right-hand list.

After creating the necessary enumerations and mapping dependency between them, you must still create custom fields and associate them with the enumerations.

#### **Create Dependent Custom Fields**

This section explains how to create custom fields that use the enumerations described in the previous section.

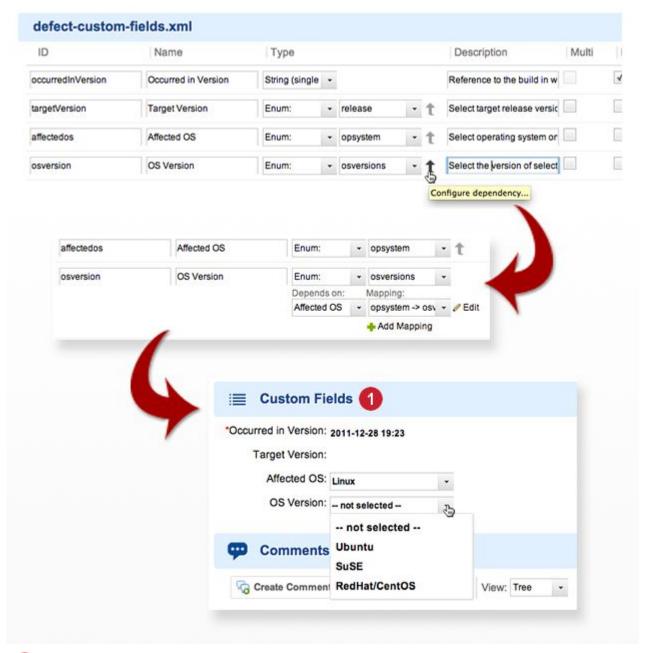
1. Configure two custom fields for the desired Work Item type, or all types is that is what you need (Navigation: Work Items > Custom Fields). Name them in such a way that the first suggests that it is the parent or main value on which the second depends.

In our example scenario, you would create a custom field "Affected OS", of type enum, selecting the opsystem enumeration. Then you create another custom field "OS Version", also of type enum, selecting the osversions enumeration.

#### Tip

If you do not see the list for selecting the enumeration, enlarge the Type column by dragging its header to the right. Enlarge until you see the enumerations list and the (Configure Dependency) icon.

- 2. Click the 1 (Configure Dependency) icon on the row of the dependent field (the field whose values you want filtered for the user). The **Depends on** field appears. In that field, select name of the "parent" field.
  - In the example scenario, you would click  $\hat{\mathbf{1}}$  on the "OS Version" field, and in the **Depends On** field, choose **Affected OS**.
- 3. When you select a value in the **Depends On** field, the **Mapping** filed appears. Select the mapping you specified when configuring the enumerations. (It is possible to have more than one mapping for an enumeration pair. If there are multiple mappings, select the one you want to use for this field pair. You can optionally create a new enumeration mapping now, using the **Add Mapping** link, or modify the selected mapping using the **Edit** link.)



Result for end users in the UI.)

#### **Remove or change Custom fields**

You may remove a custom field from the configuration, or change its type. In the case of a field of type Enum (enumeration), the field can be changed from single-value to multi-value. If you change custom field type, existing values in Work Items may become incompatible with the new type or simply invalid. You can check for such problems by running a query in the Table view of the Work Items topic when a project is open:

Search: All and containsIncompatibleCustomFields:true in Project

### **Configure Enumerations**

#### **Overview - Configure Enumerations**

Enumerations are basically lists of options that can be selected and applied as a value to some field of a Work Item. Typically, enumeration values appear as choices in a drop-down list. Take for example the **Status** field. Defaults for this field are specified in the project template. Typical options for a workflow configuration might be:

- 1. Open
- 2. Accepted
- 3. In Progress
- 4. Reopened
- 5. Resolved
- 6. Closed

#### **Enumeration files and conventions**

Enumeration values for are defined in XML files pertaining to specific Work Item fields. For example, values for the Priority field are defined in priority-enum.xml, for the Status field in status-enum.xml, and so on. If you define custom fields of the enumeration type, you would create an enumeration configuration file named similarly: [fieldID]-enum.xml where [fieldID] is the ID of the custom field. If you have a custom enumeration field named "Coffee" whose ID is "coffee", the enumeration configuration file would be coffee-enum.xml.

#### Tip

The default enumeration configuration files for the Repository scope are listed in: Default Enumeration Configuration Files

In either scope, enumerations apply to all Work Item types by default. However, enumerations can be defined to apply only to a single Work Item type - a Requirement, Task, or Defect for example. Suppose you have a set of values defined for Severity. By default, the values specified in severity-enum.xml are used for all Work Item types. Now, suppose you want a different set of values for Defects. To override the configuration for all types and have different values just for Defect type items, you would need to create a type-specific configuration for Defect. The file would be named defect-severity-enum.xml where defect is the ID of the configured Work Item type "Defect".

#### **Editing Options for Enumerations**

There are two options for editing enumeration configuration files:

- Online in the portal graphical interface.
- Offline, editing locally in an XML editor. (Advanced).

The online option provides graphical controls in the browser enabling you to define new enumerations or edit existing enumerations. All work is done online using your web browser, and changes are written to the underlying XML files. This is the recommended approach for most administrators, and the rest of this topic is based on this approach.

The offline option involves downloading a local copy of some existing XML configuration file, modifying it locally using an XML editor, and uploading an appropriately named copy of the file back to the portal. This is recommended only for advanced administrators who have knowledge of XML and the XML schema involved.

#### Overview of the configuration process

The configuration process is essentially the same for both global and project scopes. Just remember that projects inherit the global configuration, so it's usually a good practice to configure the global defaults before customizing any projects. The process is essentially as follows:

- 1. Log in with administrator permissions for the scope you want to work with.
- 2. Open the repository or project you want to configure, enter the **Administration** interface.

(Click Administration in the tool view of Navigation, and select Work Items Enumerations in Navigation.

The Enumerations page presents a table of the existing enumerations configurations for the current repository or project. On this page you can do the following:

- View the underlying XML code of any configuration file (click on the file name).
- Create a new enumeration applicable to all Work Item types or to a specific type (click Create **new configuration** at the bottom of the page).
- Modify an existing enumeration configuration (click the Edit link in the Actions column of the enumeration you want to change).
- Remove an existing enumeration configuration (click the **Remove** link in the **Actions** column of the enumeration you want to remove).

#### **Common configuration scenarios**

This section covers several common enumeration configuration scenarios:

- Edit an existing global enumeration.
- Create a new type-specific enumeration.
- Override a global enumeration in a Project.
- Use custom icons in an enumeration configuration.
- Create enumerations for Custom Fields.

#### Edit an existing global enumeration

Your Polarion installation has global default configurations predefined for the following Work Item fields:

- Hyperlink Role
- Priority
- Resolution
- Resolution
- Severity
- Status
- Linked Work Item Role
- Work Item Type
- Work Record Type

These enumerations apply to all Work Item Types. By default, there are no type-specific enumerations defined in the global scope. You can create type-specific configurations once you have the global defaults defined as you want them. You should review the default definitions for each global enumeration and make any changes before creating type-specific enumerations that override any of them. Always keep scope in mind. The global enumeration definitions will be the default for all subsequent new projects.

#### To edit an existing global enumeration:

- 1. Make sure you are working in the global (Repository) scope.
- 2. In the Enumerations table, find the row for the enumeration you want to edit and click the **Edit** link in the **Actions** column. The graphical enumerations editor appears, with the name of the configuration file in the header.
- 3. The editor presents a table, each row of which represents a value that appears in the pick-list for the relevant enumeration type Work Item field. Change any existing values on any row, or delete any rows with values you don't want, or add rows for additional values you need. Note that the Default column enables you to specify a value as the default selection in the pick-list.
- 4. Click **Save** to save changes to the underlying XML configuration file and return to the **Enumerations** page.

#### Create a new type-specific enumeration

In either scope (global or project) you can create enumerations that apply to only one Work Item type. Before doing this, you should configure the enumeration that defines Work Item types (workitem-type-enum.xml in the scope for which you want to configure type-specific enumerations.

1. Be sure you are working in the scope you want to configure (Repository or Project).

- 2. On the Enumerations page (\*Administration \*Work Items \*Enumerations), click Create new configuration at the bottom of the page. The Create New Enumeration dialog appears.
- 3. In the **Work Item Type** list, select the type to which this new enumeration will apply (Requirement or Defect, for example).
- 4. In the **Enumeration** list, select one of the already-defined enumerations, or select Custom and enter an ID for a new enumeration to apply to this definition, then click **Next**. If you selected an existing Enumeration, the enumeration editor appears with the values from the all-types configuration. If you selected Custom, the enumeration editor appears with a single new empty row where you can define a new enumeration for the Work Item type you are configuring.
- 5. Edit the enumeration definition in the enumeration editor, adding, removing, or changing values on the various table rows as desired, and click **Save** to create a new project-scope configuration file in the repository with the values you defined.

As mentioned earlier, projects inherit the globally defined enumerations. Project administrators can override the global enumeration configuration in the project scope in any or all of the following ways:

- Modify inherited global enumerations. For example, the global link roles might not be suitable for the project, so the values could be modified in workitem-link-role-enum.xml in the project scope.
- Create type-specific enumerations for the project. For example, the global settings might be
  oriented toward development projects, and in a requirements project, the values in the global
  enumerations for such things as Status, Resolution, or Severity might not be suitable, so a
  type-specific enumeration just for Requirement type Work Items might be created for the project.
- Remove enumerations. For example, the global configuration might define enumerations for Test Case type Work Items, but the project doesn't use this type of Work Item, so a type-specific enumeration might be removed for the project.

#### Override global enumerations in a project

As mentioned earlier, projects inherit the globally defined enumerations. Project administrators can override the global enumeration configuration in the project scope in any or all of the following ways:

- Modify inherited global enumerations. For example, the global link roles might not be suitable for the project, so the values could be modified in workitem-link-role-enum.xml in the project scope.
  - In the Enumerations administration page for a project, any enumeration inherited from the global configuration shows <code>Global Configuration</code> in the **Scope** column. If no project-scope enumeration exists, there is a row for it in the table and the Create link appears in the **Actions** column. Clicking that link launches the graphical enumeration editor where you can specify the enumeration values for the project scope. The editor contains rows with values from the global configuration. You can modify these values, remove them, or add rows for new values.
- Create type-specific enumerations for the project. For example, the global settings might be
  oriented toward development projects, and in a requirements project, the values in the global
  enumerations for such things as Status, Resolution, or Severity might not be suitable, so a
  type-specific enumeration just for Requirement type Work Items might be created for the project.

To create a type-specific enumeration, use the **Create new configuration** button at the bottom of the Enumerations administration page. The **Create New Enumeration** dialog enables you to select a Work Item type and apply an existing enumeration to it, or create a custom enumeration for it.

 Remove enumerations. For example, the global configuration might define enumerations for Test Case type Work Items, but the project doesn't use that type, so the inherited type-specific enumeration might be removed for the project.

#### Create enumerations for Custom Fields

Custom fields can be of the enumeration type. If so, they must have pick-list values defined in an enumeration. Custom enumeration fields can use any existing enumeration. In fact the enumeration must be defined before a field can use it.

- 1. Be sure you are working in the correct scope (<sup>€</sup>Repository or <sup>□</sup>Project) for the configuration.
- 2. Create a new enumeration using the **Create new configuration** button.
- 3. In the **Create New Enumeration** dialog, select which type of Work Item the enumeration is for, or choose  **unspecific –** to make it applicable to all types. Then in the **Enumeration** field, choose **Custom** and enter a file name prefix for the new enumeration's XML file.
- 4. Click **Next**. The Enumeration Designer appears with an empty row, ready to add values to the new enumeration.
- 5. Create the enumeration values and click **Save**. You now have the enumeration defined and ready for your custom field to use.
- 6. Navigate to Work Items Custom Fields.
- 7. In the Custom Fields page, click the Edit link on the row of the custom field definition file that contains, or will contain the custom field. The Custom Fields Designer appears.
- 8. If the custom field does not yet exist, define it in the last row of the table. In the **Type** list, choose **Enum**: Another drop-down list appears containing the IDs of existing enumerations. Choose the custom enumeration you created.

Alternatively, you could change the Type of any existing custom field to the Enum: type and specify your custom field enumeration. You should only do this in the initial stages before any Work Items are created. You should not change the field type to Enum: when there are existing Work Items in the project with field values of a different type.

#### **Use custom icons in Enumerations**

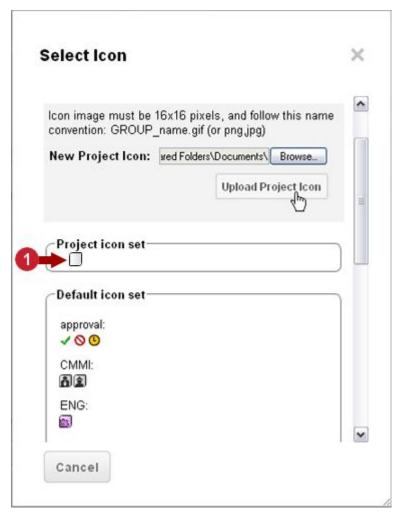
Polarion comes with a set of icons for enumerations and other configurations. You can add your own icon images and use these in new enumerations, or to replace the default icons in existing enumerations.

Icon images must be 16 x 16 pixels. Supported formats are: GIF, PNG, and JPG. Custom images are saved in the **Others** group of the Project icon set or the Repository icon set, depending on the scope you are working in when you upload your custom image(s). Within the icon sets you can save icons in custom groups by prefacing the file name with the name of a group that it belongs to.

For example, if you want to save an icon named myIcon1.png in a group named Widgets, rename the file locally to Widgets myIcon1.png and upload the renamed file.

To upload a new custom image in an enumeration:

- 1. Access the Enumeration Designer for the enumeration in the scope you want to configure.
- 2. In the row for the value you want to have a custom icon, click **Select** in the **Icon** column. The Select Icon dialog appears.



Upload local image files for use in configurations

- Where custom icons appear.)
- 3. Click Choose File and select the desired icon image on your local disk. The name of the selected file now appears in the dialog.
- 4. Click **Upload Repository Icon** or **Upload Project Icon** (depending on the scope you are working in), and wait for the icon image to upload to the portal. The image appears in the scope-specific icon set.

5. In the icon set where the new icon image displays, click on the icon image. The dialog closes and the selected icon appears in the Icon column of the Enumeration Editor.

#### Hide obsolete values

Over time, the values specified in some enumerations may become obsolete. You can delete obsolete values, but if such values have already been used in Work Items, you may still want the values to appear in the items for your historical record for traceability and governance reasons. In this case, simply deleting a value from the configuration to keep it out of the list is not the answer.

Instead, you can check the **Hidden** column on the value's row in the configuration page. When this column is checked, the value still appears in the Work Items where it was used, but end users will no longer see it in the field's select list, and consequently won't be able to specify it for any new or existing Work Items.

#### **Common enumeration configurations**

This section highlights enumerations that are frequently customized.

#### **Work Item Types**

This is probably the most common customization. Work Items can represent literally anything. If you have a project in which you need to deal with things you call "Supply Chain Risks" you can create a Work Item type "Supply Chain Risk" with whatever attributes you need for it to have.

This enumeration is so frequently customized that Polarion's Administration provides a special Navigation topic to access it: Work Items Types. (The same settings are also accessible in Work Items Enumerations in workitem-type-enum.xml.)

#### **Priority Ranks**

By default, the Priority field of Work Items can be set by selecting one of several configured "ranks" such as High, Medium, Low, which correspond to a configured minimum numeric value. Many teams prioritize Work Items according to different ranking - more or fewer ranks than the Polarion default - and will therefore need to reconfigure the prioritization.

For information, see: Configure Prioritization.

#### **Dependent Enumerations**

You can create custom enumeration pairs with a value dependency between them. When used as the type for a pair of custom fields, the selection of a value on one enumerated list causes some filtering of the values in another, limiting the choice of values for the end user. It is possible to create multiple dependency mappings between pairs of enumerations for use with different scenarios. The administrator can choose which mapping to use with any pair of custom fields associated with the dependent enumerations.

For information on how to configure this functionality, see Custom Fields with Dependent Enumerations.

#### **Note**

#### Query Enumerations in a Database:

See the Advanced Administration section to populate an enum table so that it can be used with an external reporting tool.

### **Configure Bulk Edit**

The Bulk Edit feature enables users to select multiple Work Items and edit, move, or delete them as a unit. Administrators can set a limit on the number of Work Items selected before the restricted Bulk Edit mode is activated. Restricted Bulk Edit loads less data and therefore minimizes the load on the server when a user selects a large number of Work Items for bulk editing.

You can set the limit with the system property com.polarion.ui.bulkEditRestrictedLimit, in the polarion.properties file.

The value is an integer representing the number of Work Items users can select in the Work Items table before restricted Build Editing is activated.

**Example**: com.polarion.ui.bulkEditRestrictedLimit=100

### **Define Template Work Items**

A template Work Item is a Work Item that provides default content such as Description, and default field values for a specific Work Item type. For example, the template item for a Defect or a Test Case might have boilerplate text in the Description field, and/or some specific values set for fields such as Assignee, Priority, etc. or custom fields.

Before creating a template Work Item, its type must already be defined in the project configuration. In other words, before you can create a template item for a Defect, the Defect type must be defined in the enumeration for Work Item types in the project configuration.

Once the Work Item type exists, there are 2 steps to create a template Work Item:

- Create a Work Item of the desired type in the tracker and define default content and field values.
- 2. Specify the ID of the template item in the definition of the Work Item type in the project configuration.

#### Create a template Work Item:

- 1. Log into a project with administrator permissions.
- 2. In Navigation, select **Work Items** to load the Table view of Work Items.
- 3. Create a new Work Item of the type for which you want to create a template. For example, create a new Defect item to be the template for all new Defect items.
- 4. Enter a title such as Defect Template.
- 5. In the **Description** field, enter any boilerplate text you want users to see when the create a new item of this type. For example, the Description field might contain instructions on what information

- must be entered in the Description. (The boilerplate text will appear in the Description field of new Defect items created by users, and users will be able to replace it with their own text.)
- 6. Set default values for other fields if desired. Keep in mind that default value for Status is set by the workflow, and Severity can be defined in the Severity enumeration in the project configuration. Your focus here is most likely to be on any custom fields for which a default should be set for new items of the type.
- 7. Save the new item. Make a note of the new item's ID, as you will need it in the next process step, which is to specify the ID of the template item in the Work Item type configuration.

#### To specify the template item in the Work Item type configuration:

- 1. Enter **Administration** and in Navigation, expand **Work Items** and select **Enumerations**.
- 2. On the Enumerations page, locate the row for the workitem-type-enum.xml configuration file and click **Edit** on that row.
- 3. In the configuration editor, locate the row for the type of the template item you created. If that was a Defect type item, for example, you want the row for the Defect type.
- 4. In the **Template** column on the row, enter the ID of the template item you created.
  - (You can optionally click the Wicon and query for and select the template item in the Work Item Picker dialog.)
- 5. Click Save to finish.

You need to repeat this entire process for each Work Item type for which you want to have a template Work Item.

#### Field values not copied

When a user creates a new Work Item based on a template Work Item, values in the following fields of the template item are *not* copied to the new item:

- approvals
- assignee
- attachments
- author
- categories
- comments
- created
- hyperlinks
- linkedWorkItems

- resolution
- resolvedOn
- status
- title
- votes
- watches
- workRecords

Note that images are technically attachments, which currently are not copied to new items. Therefore, images appearing in the **Description** field and rich-text custom fields of the template item do not appear in these fields of new items.

### **Customize export templates**

Polarion provides templates which are applied by default to exported documents when users export Work Items to various supported formats. The default templates can be customized, or you can create one or more customized copies of the default templates (recommended).

Export Templates are initially in the Repository scope and used for new projects. If you customize Export Templates in the repository scope, the customizations are the default in new projects. Export Templates can be customized in projects (and project templates), and the project-scope template overrides the Repository-scope version of the same template in the project (or in projects based on a project template in which Export Templates have been customized).

The Administration interface provides access to Export Templates in the Repository or projects. You can download any existing template, upload custom templates, or delete existing templates. (Note that a Repository-scope Export Template cannot be deleted from the project-scope Administration.)

#### To access Export Templates administration:

- 1. Log in with administrator permissions for the scope you want to access. If your login does not automatically take you into Administration, enter Administration using the link in the tools view of Navigation.
- 2. Open a repository (to access global repository templates) or a project.
- 3. In Navigation, expand Administration Work Items and select Export Templates.

The **Export Templates** page lists available export formats in the left-hand column. When you select one of these, icons for the existing Export Templates for the selected format appear in the right-hand column.

To customize an existing Export Template:

- 1. Select the template in the **Export Templates** page.
- 2. In the page toolbar, click **Download**, and save the template file to your local file system.

 Edit the downloaded Export Template file in a compatible application and save the changes. For example, if it is a template for Microsoft Word documents, edit the downloaded template in Microsoft Word.

If you want to create a custom copy of the template in Polarion, save the modified file with a different name.

4. Return to the **Export Templates** page in Polarion, click **Upload**, select the locally modified Export Template file, and upload it to the portal.

### Tip

You can optionally use the Repository Browser to upload modified Export Template files to the <code>export\_templates</code> folder of the project's repository (<code>[PROJECT\_NAME]/.polarion/tracker/export\_templates</code>).

For additional information, see: Customize XML Export Templates and Customize Excel Export Templates.

#### **Customize field filters**

Administrators can customize which Plans appear in the Planned In list using a velocity script.

They can create custom queries with the following variables:

- \$\sigma\_wi The Work Item variable. Can filter for results based on Work Item fields. (Status, Severity, Priority, Type, Assignee, Author, Custom Fields etc.)
- \$user The User variable. Can filter for results that are only relevant to the current user. (Name, E-mail, Role etc.)

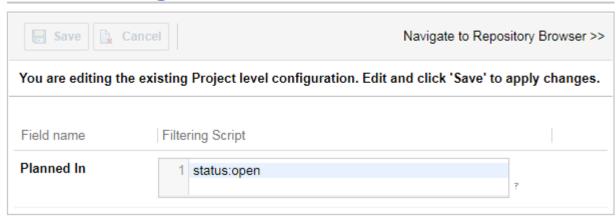
#### Note

Only Lucene queries are supported.

### Add a custom field filter velocity script:

- 1. Log in with administrator permissions and enter **Administration**.
- 2. In Navigation, select \* Work Items Field Filtering
- 3. Enter a script in the **Filtering Script** column of the **Filters** section.

# Field Filtering



(See the example scripts below.)

4. Click Save.

Polarion provides a query as the default filter for the items listed in the **Planned In** field.

For example, if your project uses the E-Library template, the query is:

```
((project.id:elibrary AND NOT HAS VALUE:projectSpan) OR projectSpan:(elibrary)) AND
 (allowedTypes:userstory OR allowedTypes:######NULL) AND NOT HAS VALUE:finishedOn
```

This basic condition is always applied.

Any additional query that's added, is appended to it.

### **Examples:**

Only select Plans with an *Open* status:

```
status:open
```

Display all Plans that are Open are In Progress.

```
template.id:iteration AND
#if ($wi.getStatus().id == "draft")
status:open
#else
NOT HAS VALUE: finishedOn
#end
```

Lists Plans with a status other than *Done*. Only the Work Item Assignee(s) can view/change them.

```
#if ($wi.getAssignees().contains($user) )
template.id:iteration AND (NOT status:done)
#else
NOT status:(open inprogress done)
#end
```

### **Configure calculated fields**

You can define fields which derive their value from a calculation based on values of other fields in the same Work Item, or in child Work Items. Here are several key concepts to keep in mind:

- Aggregation of the values is always done according to a parent-child hierarchy of linked Work
  ltems defined by links that have a role which has the attribute parent set to true in roles
  configuration file workitem-link-role-enum.xml (found in the Work Items Enumerations
  topic.)
- A calculated field's type must be the same as that of the fields from which it is calculated. Supported types are: integer, float, currency, duration.
- Calculated field values are calculated asynchronously and incrementally after each commit.
- Values of calculated fields are not committed to the underlying repository, and are therefore not present in the Work Item history.
- Calculated fields appear as read-only fields in the portal user interface. I.e., they are not modifiable by end users.
- Calculated fields can be configured in these scopes: global, project and Work Item type.

The following operations are available for calculated fields:

- sum: Calculates and renders the sum of two or more values.
- **multiply**: Multiplies several values and renders the resulting value. Risk value \* User Story size value, for example.
- **average**: Computes the average of several values and renders the resulting value. Calculate average time spent on Work Items, for example.
- min: Computes a minimum value. For example, compute the minimum time of several builds.
- max: Computes a maximum value. for example, compute the maximum duration of a test execution.

#### **Example use case: Work Item cost**

A common use case for calculated fields is the calculation of the cost of a set of Work Items. For example, a project could have a custom field **Cost** configured for Task type Work Items, and a **Total Cost** field configured for Requirement type Work Items. The Requirements would be linked to the tasks that implement them with a "parent" link role (appropriately defined in the project link roles configuration). Alternatively, this could be done with just Tasks. Both custom fields could be

configured for Task type items, and the tasks linked so that sub-tasks are children of a main task. The aggregated cost of the sub-tasks is then reflected in the Total Cost field of the linked tasks. Using the latter example of Task-to-Task, the steps to realize this use case are as follows:

- Check that the **parent** role has the parent flag set in link roles configuration.
- Define custom fields to become calculated fields, in this case Cost and Total Cost, making sure they are of the same data type required for the <sum> calculation.
- Configure calculated fields that reference the custom fields.

#### Define the custom fields

For this use case, you would need to configure 2 custom fields (probably in the project scope). In the \*Administration \*\* \*\* Work Items \*\* \*\* Custom Fields topic, you would enter the following using the Custom Fields Designer:

- ID: cost
- Name: Cost.
- Type: float
- Description: Cost of a Task

Create a new row in the table for the next custom field and enter the following:

- ID: totalCost
- Name: Total Cost
- Type: float
- Description: Aggregate cost of all linked tasks

#### Define the calculated fields

You now need to reference the IDs of the custom fields appropriately in the Calculated Fields topic in the appropriate scope of calculated-fields.xml:

- 1. In Navigation, select Administration Work Items Custom Fields.
- 2. On the Calculated Fields page, click the Create New Calculated Fields button. The Create New Calculated Fields dialog appears.
- 3. In the dialog, select the Work Item type from the list of types and click OK. (For the example being discussed here, you would select Task). The Calculated Fields Configuration page loads.
- 4. In the Calculated Field ID column, enter the ID of field that will store the calculation result. In the example this would be the totalCost custom field.
- 5. In the Source Field IDs column, enter the ID(s) of the field(s) to be calculated. In a sum calculation, this would be the ID(s) of the field(s) to be summed. In the example, it would be the cost field.

- 6. In the **Field From** column, select the value specifying whether the field in **Source Field ID(s)** is a field in the parent or the child Work Item.
  - If more than one field will be a source for calculation, you can add fields by clicking the icon next to the Field From value.
- 7. To define another calculated field for the same Work Item type, click the icon in the **Actions** column and repeat the above steps to define another calculated field.
- 8. Click **Save** to save changes and create the new calculated field(s).

The value of the calculated field **totalCost** for a Work Item will be a sum of the value of the **cost** field in that Work Item plus the sum of all values of the **totalCost** field from all child Work Items.

For example, if Work Item B is linked to Work Item A with role parent (that has parent flag set), and there are no other linked work items, and Work Item B has cost value of 10, and Work Item A has cost value of 20, then the calculated values in Work Item B will be **totalCost** = 10, and in Work Item A **totalCost** = 30.

Other use cases can make use of several of the "standard fields". For example, you could set up a calculated field **Total Remaining Time** that would aggregate the values of the remainingEstimate fields in a set of parent-child linked Work Items. Or at the beginning of a new project you could create a Total Estimated Time field that sums the values of the **initialEstimate** field of a set of Work Items. Remember that calculated fields can be included in exported Work Item reports. Values will be current as of the date-time of the export.

#### Modify calculated fields

If you need to modify an existing calculated field configuration, you can do so. Be sure you are making changes in the right scope.

- 1. In the Administration interface, select the scope of the configuration: Open Project or Project Group dialog. (See: Access Projects.)
- 2. Expand the Work Items topic and select Calculated Fields.
- 3. In the Calculated Fields page, click the **Edit** button in the Actions column of the calculated field you want to modify, and make the desired changes in the Calculated Fields Configuration page.
- 4. After saving the changes, click the **Recalculate Fields** at the top left of the Calculated Fields page.

# **Configure Work Item Workflow**

#### **Note**

This topic covers workflow configuration for **Work Items**. There is a separate workflow configuration for Documents.

Workflow controls a development process to ensure that no process steps are missed or skipped over. Workflows for Work Items in Polarion can be mapped to the steps in almost any process, whether it's defined by some methodology (Scrum, V-Model, etc.), completely by customer, or a hybrid of both.

#### Warning

Workflow definitions and status enumeration work in tandem.

Type-specific and generic enumerations should not be mixed together.

(This also applies to *Project* and *Global Definitions*.)

A workflow consists of a set of named statuses and status transitions, transition conditions and dependencies that a Work Item passes through in its life cycle. For example, consider the following set of status definitions:

- 1. Open (which might be the initial status for new items)
- 2. Accepted
- 3. In Progress
- 4. Resolved
- Closed
- 6. Reopened

If a user changes the Work Item's status from Open to In Progress, or Open to Resolved, this invokes an action, which triggers a status transition, which is automatically tracked in the Work Item's history. Actions may trigger underlying system functions... creating a linked Work Item, or clearing some field's value, for example. Workflow actions can be conditional... that is, some condition must be checked and fulfilled before the user can execute the action.

You can customize Work Item workflows and transitions in several scopes: globally (for all projects), project-specific for individual Projects, and/or Type-specific, which applies only to a specific type of Work Item in a project. (Type-specific can be configured both globally and in projects.) Polarion looks for the most specific workflow definition first and proceeds toward the most generic in the following search sequence:

- 1. Project-specific and Work Item type-specific
- 2. Global and Work Item type-specific
- 3. Project-specific
- 4. Global

#### **Configuration process summary**

Generally you will perform the following operations in the order listed:

1. **Configure Work Item custom fields:** define any that will be needed for Work Items in the scope. If your workflow will not have any functions and conditions that reference Work Item custom fields, you can skip this operation.

- 2. **Configure Statuses:** Determine the statuses a Work Item can have at various stages of its lifecycle (e.g. "Draft", "Implemented", etc.) and create a Status definition for each one in the **Statuses** section of the Workflow Designer.
- 3. **Configure Actions:** Determine what actions are needed to transition a Work Item from one status to another throughout your development process, and create an Action definition for each one in the **Actions** section of the Workflow Designer.

Action names appear in the drop-down list of a Document's **Status** field in Document Properties. The name indicates to the user what transition will take place as a result of invoking the action. For example, a "Mark as Implemented" action might transition a Work Item to an "Implemented" status.

- 4. **Configure Functions and Conditions:** This is optional, depending on whether or not any system functions should execute when an action is invoked by users, and if the execution should be conditional.
- 5. **Configure Transitions:** Now you can specify how the Work Item **transitions** from one status to another in the Transitions section of the Workflow Designer page (see Using the Transition Matrix).

### **Workflow Designer**

A graphical Workflow Designer tool is provided in the web interface. It presents a visualization of workflow configurations and related enumerations such as status definitions. The Workflow Designer is available in all of the available scopes.

#### **Access the Workflow Designer for Work Items**

To access the Workflow Designer tool, you need to go into Administration for the scope you want to customize (global or project).

#### **Access for Global configurations**

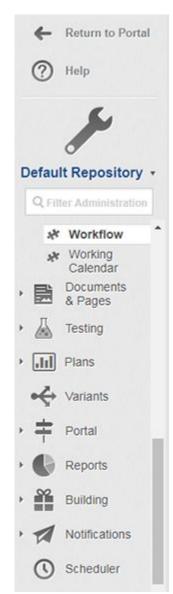
If you want to configure Global workflows, select Repository in the **Open Project or Project Group** dialog. Keep in mind that changes will apply not only to new projects, but to all existing projects unless a project-scope configuration has been defined for them individually.

#### **Access for Project configurations**

To configure workflow for a specific project, select the project in the **Open Project or Project Group** dialog after entering Administration.

#### The Workflow topic

Once you are in the desired Administration scope, go to the **Navigation** pane, expand the **Work Items** node, and select **Workflow**. The Work Item Workflow page opens and displays a table of existing and available workflow configurations.



# Work Item Workflow

Info: The workflow configuration of the Work Item enables you to specify workflow transitions from start state to end state and perform a specific action for each transition.

Work Item Type	Scope	Actions
All Types	Global Configuration	Edit
✓ Task		Create
Change Request	Global Configuration	Edit Delete
Requirement	Global Configuration	Edit Delete
P Defect		Create
Test Case	Global Configuration	Edit Delete
Nový Globální Typ		Create

# **Quick Help**

The table above shows the workflow configurations currently defined for Work Items. In general, configurations can be Global or project in their scope. The Scope column indicates the scope of the configuration currently in use. A Work Item workflow configuration can also be specific to one or more Work Item types.

There are two simple rules that define the order in which configurations are used:

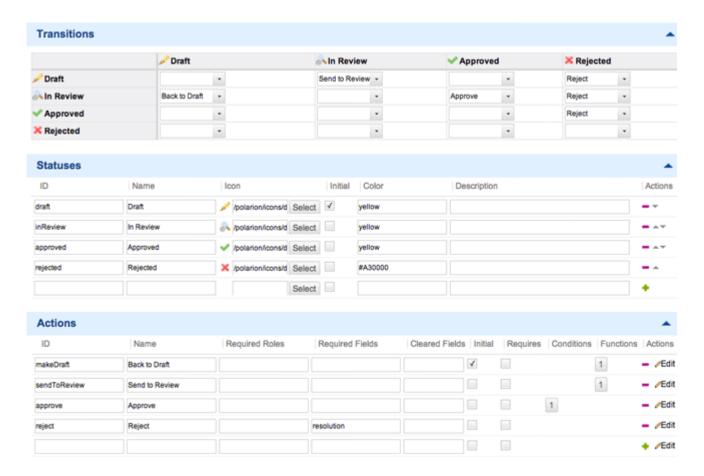
- Configuration in the project scope always overrides the Global configuration.
- Type-specific configuration always overrides the All Types configuration.

Workflow configurations table shows existing configurations, and provides access to the Workflow Designer.

The first row of the table is labeled **– All Types –**. This is where you can define a default workflow for all Work Item types in the scope that do not have an overriding type-specific workflow configuration defined for them. The **Type** column indicates whether or not a type-specific workflow is configured for that type in the current scope. If a cell in that column is empty, it means no workflow is configured for the Work Item type shown on that row, and the All Types workflow will be used.

## **Using the Workflow Designer**

Please refer to the following figure while reading the information in this section.



## Workflow Designer for a Work Item

The **Transitions** section at the top of the Workflow Designer is a matrix of workflow transitions between Work Item statuses. The direction is from row to columns. For example, the value where the row **Open** intersects the column **Resolved** represents the transition from the "Open" status to the "Resolved" status. At each intersection where a transition is possible, there is a drop-down list of actions. The Actions are defined in the **Actions** section of the designer, an they occur when the transition is triggered by the user changing the Work Item status. For more information, see **Using** the **Transition Matrix**.

The **Statuses** section is an enumeration editor for defining workflow status definitions for the scope in which you are working.

Below the **Statuses** section is the **Actions** section. This is an editor for transition Action definitions. Action names that appear in the drop-down lists in the **Transitions** matrix are defined here. The **Required roles**, **Required fields**, and **Cleared fields** columns are comma-separated lists of role IDs and field IDs. Only simple field types can be specified in these columns (i.e., not collection or structure types).

The **Initial** column in each Action row enables you to specify one action as the first action that must be executed in order for the Work Item to enter the status defined as the initial status when a new Work Item of the applicable type(s) is created. (Defining an **Initial** status is *required*.) Specifying an initial action is mainly important if some fields must be marked as required, and/or some field(s) must have value(s) cleared, and/or some workflow function must be executed before the defined initial status ("Draft", for example) is applied to a new Work Item. If no initial action is specified, then

no fields are changed and no workflow function is executed when a new item is created, even if the Action has these defined. Field changes and workflow functions only take place when the action is invoked manually by changing the Work Item's status.

The Conditions and Functions columns in each Action row display indicate if any workflow conditions and functions are configured for the action. (Indicator is a button labeled with the number of conditions or functions currently configured. Clicking the button opens the respective dialog where the conditions/functions are specified.)

The **Actions** column in this section of the designer contains 2 icons. The / icon invokes the Action Details dialog, which provides tools for defining workflow functions and conditions to be executed when the Action is invoked.

## Required roles

In the Required Roles field you can enter a comma-separated list of roles that a user has to have in order to be able to perform the Action. These roles may be either project or global user roles (e.g. "project\_admin" or "admin") or two special roles:

- workitem.assignee only assignee of the Work Item has this role
- workitem.author only author of the Work Item has this role

Valid role names are all global and project roles (admin, user, etc.) in the same form as in User Management- Roles. There is no need to prefix them with a project ID, because they are always resolved against given Work Item's project. The only permitted logical role is workitem.assignee. Roles have OR logic - one match is enough for the condition to succeed.

#### Available conditions and functions

For a listing of the available conditions for this configuration and functions that can be specified, see: Workflow Conditions and Functions.

#### **Action Names**

It can be helpful to your end users if you specify the Name property of an action as a verb or verb phrase. For example, suppose you have w statuses: Implemented and Verified, and you want to define Actions for the transition from some other status(es) to these statuses.

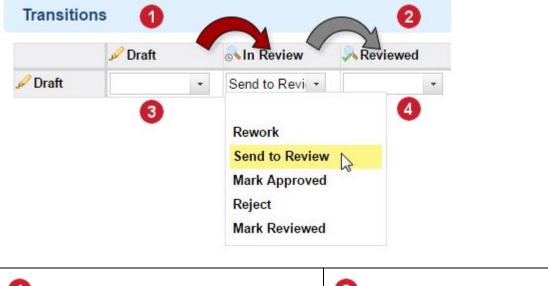
For transitions to the Implemented status, a good choice for the Action name would be Mark as Implemented as opposed to just Implemented. The same idea applies to the Verified status. The transition Action name might be Verify Implementation Of Mark as Verified.

## **Using the Transitions Matrix**

When defining transitions using the Workflow Designer's Transitions matrix, it's important to keep two things in mind:

- You are defining what transitions are allowed between any 2 statuses
- Therefore, you don't need to specify an action for every transition... just those transitions you want to allow to occur between 2 statuses

Consider an example of a workflow for a Requirement. Assume that, among others, the statuses Draft and Awaiting Review are defined.





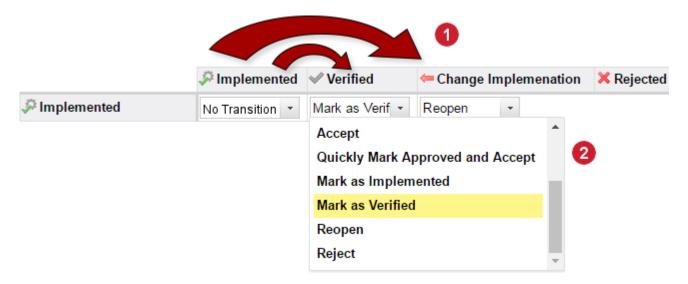
Allow and disallow transition to take place between 2 Work Item statuses

It makes sense to allow transition from the <code>Draft</code> status to the <code>Awaiting Review</code> status, so at the intersection of these two, you specify the <code>send to Review</code> action (which has already been defined in <code>Actions</code>).

It would not make sense for the process to allow transition directly from the Draft status to the Reviewed status (assume the team process does not allow this), so no action is specified where Draft intersects Reviewed in the matrix.

For some status, you may want to have more than one possible transition. For example, suppose you have the status definitions Implemented, Verified, and Needs Changes. For Work Items having the Implemented status, let's say there are two possibilities in the process: QA can pass the item and give it the Verified status, or QA can reject the item and send it back for changes, giving it the Needs Changes status. The workflow can be configured to support this process but allowing 2 transitions on the Implemented status.

- Transition from Implemented Status to Verified Status via the Mark as Verified action.
- Transition from Implemented status to Needs Changes status via the Reopen action.



Where it makes sense for your process, enable more than one transition for a status



## **Workflow for Approvals**

Polarion's standard project templates are pre-configured to support a formal approval process. Even if the overall process as configured meets your needs, there may still be some things you want to adjust in the approvals workflow. For example, the default configuration includes a workflow function on some Actions that automatically adds users to the Approving Users table of the **Actions** field of Work Items, resulting in notifications to those users. However, the users so added by the default settings may not be users you actually want, or there might be other users you want included that are not.

Basically, for each Work Item type you should review the Actions that relate to the approval process in the Workflow Designer. For example, for a requirement, click Editfor the Send to approve action. By default, this Action invokes a workflow function AddDefaultApprovals, which automatically adds some users to the Approvals field of Work Items when the **Send to approve** action is invoked.

By default, this function uses the approvals.roles parameter with a single value project approver. This means that when the Action is invoked, all users whose user profile includes the project approver role will be added to Approving Users table in the item's **Approvals** field. If this is not what you want, you can either change the role, or add more roles, or use the approvals.users parameter instead. (For more information, see Workflow Functions.)

Another Action to review might be Back to Draft. By default, this Action invokes the workflow function ResetApprovalStatus. Perhaps you want to invoke some other function after that... SetDate, for example.

## **Require Signatures for Workflow Actions**

You can require end users to electronically sign when invoking a workflow action to transition to a new status.

In the **Actions** section of the Workflow Designer, each action has a check-box option, Requires signature. Checking this option means that a user invoking the action is requested to provide an electronic signature, and the workflow transition will not take place until a signature has been logged.

To configure this functionality, you need to edit the following properties in the system properties file polarion.properties (follow link for location):

• secure.approvals: The value is a Boolean true or false (the default value is false). If set to true, then when an end user executes an approval or disapproval action, the password dialog is presented as described above.

**Example**: secure.approvals=true **Or** secure.approvals=false

• secure.dialog.title: The value of this property is a string of characters. The string appears as the title of the password dialog presented when an end user invokes one of the configured "secure" workflow actions. You only need to set this property if you want to override the system default title for the dialog (shown in the example below).

**Example**: secure.dialog.title=Enter Password

• secure.dialog.message: The value of this property is a string of characters. The string appears as a message in the password dialog presented when an end user invokes one of the configured "secure" workflow actions. You only need to set this property if you want to override the system default message for the dialog (shown in the example below).

**Example**: secure.dialog.message=By entering my password I signify that I have reviewed and approved the content of this individual Work Item, or group of Work Items, just as I would have by applying my own handwritten signature.

## Tip

For security, it is highly recommended to use the https protocol for access to Polarion when your system is configured to use electronic signatures.

# **Enable Electronic Signatures Support**

You can enable secure approvals that require electronic signatures to satisfy compliance or governance requirements.

Use the following system properties in polarion.properties:

- secure.approvals=true: Enables electronic signatures for approvals.
- secure.approvals.comment=put your custom comment text here: The custom text is used as the text of a new comment created when a user enters a password to confirm a secure approval.

An example value would be: Item was e-signed.

 secure.approvals.comment.onBehalfOf=put your custom comment text here: Similar to the above property, but also supports a single user who can confirm a secure approval on behalf of another user.

Supports the following variables:

O \$user

Will insert the name of the user currently logged in.

O \$otheruser

Inserts the name of the user that the current user is doing the approval on behalf of.

0 \$verdict

Inserts the approval verdict.

O \$img

Adds an approval verdict image.

- O An example value would be: Approval verdict: \$img \$verdict (by \$user on behalf of \$otheruser).
- secure.approvals.comment.text=put your custom comment text here: Displays the state of the approval.

## Supports the following variables:

- O \$img
- O \$verdict
- O \$user
- O An example value would be: \$img \$verdict by \$user

# Localize the Approval messages:

You can override the signature approval messages for the below properties directly in the polarion.properties file.

You don't have to dig through and change the localization messages file.

```
(For example, the
```

C:\Polarion\polarion\plugins\com.polarion.psvn.translations.en\_3.18.2\META-INF\messages\_en.properties file for English.)

## Tip

Non-ASCII languages should be converted to Unicode.

- Shutdown the Polarion server.
- 2. Open the polarion.properties file in your favorite text editor.

(**Default location**: C:\Polarion\polarion\configuration.)

3. Uncomment the secure.approvals=true property.

This will enable secure approval comments using the default (English) messages. (If it's not there, just add it.)

4. Add the following properties with your custom messages .

(Customize the default messages after the =)

• secure.approvals.comment=Your custom message

Overrides the dialog.secure.comment=Item was e-signed value in the localization messages file.

secure.approvals.comment.text=Your custom message

Overrides the dialog.secure.comment.text=Approval verdict: \$img {1} value in the localization messages file.

(Where {1} is the verdict text.)

• secure.approvals.comment.onBehalfOf=Your custom message

Overrides the dialog.secure.comment.onBehalfOf=Approval verdict: \$img {1} (on behalf of {2}) value in the localization messages file.

(Where  $\{1\}$  is verdict text and  $\{2\}$  is the user that the approval was done on behalf of.)

- 5. Save your changes and close the polarion.properties file.
- 6. Start the Polarion server.

# **Configure Work Records**

If your Polarion product license supports the Work Records feature, and if this feature is enabled in the project, an administrator can configure values that represent Work Record types, and also whether or not the **Type** and **Comment** fields of Work Records appear to end users as required fields. You can also configure a lock date for Work Records to prevent any dated earlier from being modified.

# **Configure Work Record types**

In Work Records, the **Type** field is a drop-down list of values that define Work Record type. The values appearing in the list are customizable in the <code>work-record-type-enum.xml</code> configuration file in both global and project scopes. This file is accessible in the Enumerations topic ( **Administration** interface: **Navigation** pane: **Work Items Enumerations**). You modify this configuration in the same was as any other enumeration. For more information see: Configure Enumerations.

## Make Type and Comment fields required

This configuration is available in both global (repository) scope and project scope. In both scopes, the fields are not required by default and you will need to create the necessary configuration to change this. Configuration is performed entirely in the Administration interface of the portal (i.e. you do not need to edit any XML configuration file).

To change the default and cause the **Type** and **Comment** fields of Work Records to be required fields:

- 1. Enter the Administration interface with permissions for the scope you wish to configure (global or project).
- 2. In the Open Project or Project Group dialog, select Repository if performing the configuration in the global scope, or select the project you want to configure.
- 3. By default, no configuration exists and you will see configuration does not exist yet in the Configuration section together with the Create button. Click this button to create the configuration for the selected scope. Two check boxes appear in the section: Type Required and Comment Required.
- 4. To make the **Type** field required in Work Records, check the **Type Required** box. To make the **Comment** field required in Work Records, check the **Comment Required** box.
- 5. Click the **Save** at the of the page to save your changes. If you decide not to save changes, navigate away from the page by selecting a different topic in the **Navigation** pane.

## **Lock Work Records**

In Polarion products that support Work Records, you can configure a locking date for Work Records in each project. Work records dated on or before the locking date cannot be added, removed, or updated for any Work Items in the project. This enables project managers to "close" work record reporting by teams as of the specified date. Note that this configuration may need to be changed fairly often - monthly or even weekly. The principal Polarion administrator may find it easier to grant a project manager or other user project administrator permissions so that this configuration task can be delegated.

To set a locking date for Work Records:

- 1. Log in with administrator permissions for the project that is to have the locking date set or modified.
- 2. Enter the Administration interface and select the project you want to configure in the **Open** Project or Project Group dialog (see: Access Projects).
- 3. If not already selected, select the **Project** topic in the **Navigation** pane. The project properties appear in the Content Pane.
- 4. Specify a date in the Lock Work Records Date field. This field can be edited "in place" without clicking the Edit button. When you hover your pointer above the field, you can invoke a calendar picker in which you can select a date. The selected date will be the date when Work Records in the project will be locked.
- 5. After changing the date, click the **Save** button to effect the change.

# **Enable decimal hours in Work Records**

By default, reporting of time values (e.g. Initial Estimate, Time Spent, etc.) uses a fractional convention. For example one half hour is reported as 1/2 h. It is possible to change reporting of hours to decimal format (.50, 2.5 for example). This requires a change to the system properties stored in the system properties file polarion.properties. For information on this setting, see System Tuning for Advanced Administrators, property useDecimalHoursDurationFormat.

# **Configure the Work Items Table**

You can configure the data columns that display in the Table view of the Work Items page. You can customize the column display either globally or for a specific project. If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface.

Note that individual users can reconfigure the Work Items table to suit their own needs and preference via the Table Configuration dialog available in the Polarion web interface. Their changes are saved to their respective user profile. For information, see: Customize the Work Items Table.

## The Configuration File

The Table customization is done by editing the table-settings.xml file for the scope you want to customize, either online in the Table Configuration topic of Administration, or offline by downloading the file from that topic, editing offline, and then uploading the modified file back to the Polarion server in the same Administration topic.

The configuration file contains <column> tags for all the available data fields that can be displayed in the Work Items Table. The tags for the columns that appear by default are uncommented. Other available tags are placed inside a comment. If you want to remove one of the default columns so that it does not appear in the Table, cut it and move it into the comment that contains the other available column tags. To make one of the other available columns appear in the Table, cut its tag from the comment that contains the other available columns and paste it into the block of uncommented column tags in the position you want it to appear.

For example, if you wanted to add **resolution** as the third column in the table, cut the tag <column id="resolution" ...> from the block of commented tags and paste it as the fifth tag in the block of uncommented <column> tags.

## Accessing the configuration files

- 1. Log in with the necessary administrative permissions and enter the **Administration** interface.
- 2. Open the scope you want to configure: Repository, or a project. (Click the current repository or project name in Navigation, just under the Polarion logo image.)
- 3. In the Navigation pane, expand Work Items and select Table Configuration.

You can now download the relevant configuration file. If you are making a project configuration and no configuration file for the project scope exists, you can use the controls in the **Edit Project Configuration** section to create a project-scope copy of the global configuration file in the text editor, where you can modify and save it.

## **Note**

Administrators with global rights can optionally change the table configuration using the Table Configuration dialog as described in: Customize the Work Items Table. To change the global configuration, be sure you have selected the Repository the **Open Project or Project Group** dialog before invoking the Table Configuration dialog.

## Limit the loading of Work Items

User queries on Work Items can sometimes find very large numbers of Work Items. However, it is not likely that all of the items are actually needed. So for performance reasons, you may want to set a limit on the number of Work Items that can be loaded in tables of Work Items. You can do this with the system configuration property loadAllLimit (in the polarion properties file.) By default, the limit is 3000 Work Items.

Note that setting this limit in the configuration does not mean that is the number of items that will always be loaded. It reflects only the maximum number of items that are allowed to be loaded in the table. When the number of items found by a guery exceeds the configured limit, users are presented with a link Load first NNNN where NNNN is the configured load limit (for example: "Load first 3000"). When a user clicks the link to start the loading, items are loaded in blocks of 50. The count of loaded items is updated after each block loads, so the user can estimate when the loading will be finished.

## **Note**

The configured limit applies in Users, Roles, Projects, Time Points, Categories, Builds and Jobs in addition to the Table and Multi-Edit views in the Work Items topic.

## Fields displayed as table columns

In a new, clean installation the fields available to display as table columns are listed in the configuration file in comments. However, you may have an upgrade installation which has a version of the configuration file without such comments. Therefore, it may be useful to have a complete list of fields that you can optionally configure for display in the Work Items Table. The following table lists the field IDs of fields that you can display as columns in the Work Items table, along with which ones are shown by default in new installations.

Field ID	Shown by Default
type	YES
title	YES
priority	YES
severity	YES
status	YES
created	YES

Field ID	Shown by Default
assignee	YES
author	YES
plannedStart	YES
timePoint	YES
description	NO
categories	NO
resolution	NO
previousStatus	NO
dueDate	NO
initialEstimate	NO
timeSpent	NO
remainingEstimate	NO
attachments	NO
updated	NO
plannedEnd	NO
approvals	NO
planningConstraints	NO
linkedWorkItems	NO
hyperlinks	NO
linkedRevisions	NO
comments	NO

# Note

Custom fields may also be used.

# **Configure queries**

You can configure some named queries than can be entered in the Query field of the Work Items topic. Users can simply enter the query name and the associated query syntax will be executed.

This configuration is available for both global (repository) and Project scope. If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface. The configuration is specified in the queries.xml configuration file. You will find comments with a customization example in this file.

## To access the queries.xml configuration file:

- Log in with administrator permissions for the scope you want to configure (repository and/or Project).
- 2. Enter the **Administration** interface.
- 3. If making a global configuration, select **Repository** in the **Open Project or Project Group** dialog. If making a project configuration, navigate to and select the Project in the Open Project or Project Group dialog. (See: Access Projects )
- 4. In the **Navigation** pane, expand the **Work Items** node and select **Queries**.
- 5. You can edit the configuration file on line or download it for off-line editing and later re-upload. If you are editing a project for which no project-scope configuration exists, you can click the Paste global configuration button to load the XML from the global configuration into the online XML editor, where you can proceed to modify it.
- 6. If you edit the file off-line, use the controls in the Upload New [Project][Global] Configuration section to upload the modified file back to the Polarion repository.

The default global configuration contains several pre-configured named queries, each contained in a <query> element. The user-friendly name is specified in the name attribute, and the Lucene query syntax, which will execute when the user invokes the name in the Query field, specified in the query attribute of the element. For example...

```
<query name="Unscheduled" query="NOT HAS VALUE:timePoint.id"/>
```

...retrieves Work Items having no value specified in the Time Point field.

## Change the system default query

When users access the Work Items table in the Work Items topic, Polarion runs a default query that simply queries for all unresolved Work Items: NOT HAS VALUE:resolution. This default query applies to the Repository and all projects.

Administrators can change this default query in the system property com.polarion.ui.defaultQuery in the

polarion.properties system configuration file. The above query is the default value for this property, and you can change it to any valid Lucene query that is preferred as the system default.

Users can override the query specified in this property by saving any query in the Visual Query Builder as their personal default query for the Work Items Table.

# Configure the global working calendar



The Working Calendar feature enables the Polarion project planning engine to calculate the amount of actual working time available to projects. The Working Calendar defines the working days comprising a normal work week, and working hours within each normal working day. Exceptions to the normal working and non-working days/times, including temporary exceptions, can also be configured.

Working Calendar can be configured in 2 scopes: global (repository), and User. The global Working Calendar, configured in the Administration interface, affects all Projects in the repository. In this scope it is possible to allow for known non-working time, such as national holidays, across the entire organization.

The user, or personal Working Calendar enables specification of vacations, personal holidays, and other personal time off from work, or fewer workdays or working hours in the case of part-time people, for individual users. This configuration can be performed either by an administrator, or by individual users.

The aggregate of both configuration scopes then affects the overall Live Plan of each Project. User calendars affect only the plan of projects for which they have a role. The Live Plan visually represents non-working time configured for both scopes and applicable to the project plan being rendered.

This section describes the procedures for accessing and modifying the global Working Calendar, and for accessing a user's Working Calendar for administrators working in the Administration interface. The user calendar contains additional user interface controls. The modification procedure is the same whether the user calendar is accessed by an administrator or by the individual user.

## Access the working calendar

This section describes how to access the global Working Calendar, and a user's Working Calendar in the Administration interface. Administrator permissions are necessary for access. (Non-administrators wishing to edit their personal Working Calendar should consult Configure your working calendar.)

# To access the global Working Calendar configuration:

- 1. Enter the **Administration** interface.
- 2. Select Repository in the Open Project or Project Group dialog.
- 3. In the Navigation pane, expand the Work Items node and click on Working Calendar.

To access a user configuration:

- 1. Enter the **Administration** interface.
- 2. Select Repository in the Open Project or Project Group dialog.
- 3. In the Navigation pane, expand the User Management node and click on Users.
- 4. Use the Search feature to locate the user whose Working Calendar you want to configure.
- 5. In the user account preview pane (bottom of the screen), scroll downwards, locate the **Working Calendar** button and click it to display the selected user's personal Working Calendar.

# Modify the global working calendar

The Working Calendar page consists of the following 2 sections:

- Regular Schedule
- Schedule Exceptions

## Regular schedule

This section defines the normal work schedule for your organization. In a new installation, the defaults are set as follows:

**Start Time**: 09:00

End Time: 17:00

- Days with working time specified: Monday Friday
- Days with NO working time configured: Saturday, Sunday

Days that have Start Time and End Time values are "Working" days. Days having no start/end time values are "Not-Working" days.

## To change the start or end time for a Working day:

- 1. Change the **Start Time** and/or **End Time** value(s) as desired.
- 2. Click **Save** in the toolbar above the section.

To quickly change a Working day (one with Start/End Time values specified) to a Not-Working day, you can click the icon that appears to the right of the day. (alternatively, you can just clear the values in the **Start Time** and **End Time** fields.)

Tip

Click **Revert** to cancel the changes.

# Schedule exceptions

In the Schedule Exceptions section you can define exceptions to the regular schedule. Exceptions can be permanent, or temporary for a specific period of time. For example, holidays that occur on the same date every year can be set up as permanent exceptions to the normal work schedule. Holidays that fall on a different date each year can be set up as temporary exceptions.

## Schedule Exceptions can be either of two types:

- Time Off the exception is time that will not be available for work in addition to any non-working time configured in the Regular Schedule.
- Time Working the exception is time that will be available for work in addition to the working time configured in the Regular Schedule.

## Add a Schedule Exception:

- 1. Enter a meaningful title for the exception in the Title field. For example: New Years Day Holiday.
- Select the exception type by choosing a value in the **Type** drop-down list. For example: **Time Off**.

- 3. Specify the date when the schedule exception is to begin in the Date From field in this format: yyyy-mm-dd. Alternatively, click the calendar icon next to the field and choose a date in the pop-up calendar. For example: 2008-01-01.
- 4. If the exception you are defining is temporary, specify the date when it should end in the **Date To** field, in the same format as mentioned in the previous step. For a one-day exception like our New Years Day example, the value should be the *same date* as in the **Date From** field. If the exception is ongoing, leave the **Date To** field empty.
- 5. If the exception you are defining occurs only on one specific weekday in the time period defined, choose that day in the drop-down list in the **Weekday** column. Otherwise, leave the default value All days.
- 6. If the exception you are defining is of the type "Time Working", set appropriate time values in the **Time From** and **Time To** fields, which are enabled when this type is selected. **Time From** is the time when the additional working hours start, and **Time To** is the time when the additional working hours finish.
- 7. If you want to add another Schedule Exception, click the icon in the **Actions** column and repeat the above steps on the new row that appears.
- 8. When you are finished defining your Schedule Exception(s), click **Save** in the Working Calendar toolbar. To cancel all changes, click the **Revert** button.

## Remove a Schedule Exception:

- 1. Click the in the **Actions** column of the exception you want to remove.
- 2. Click **Save** in the Working Calendar toolbar. To cancel the remove (before any Save), click **Revert**.

# **Configure Planning**

# ALM

## You can configure certain defaults for the Live Plan project planning engine:

- Default initial time estimate value
- Query that determines which Work Items are processed by Live Plan
- Default roles for dependency and parent relationships for linked Work Items
- Work Item splitting
- Default behavior of Due Date

Note that the value for the number of hours per work day is read-only in this Administration page. This value is set in the system property polarion.durationHoursPerDay. See Advanced System Tuning for information on this property.

This configuration is available for both global (repository) and Project scope. If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface.

## Access the configuration

- 1. Log in with administrator permissions for the scope you want to configure (repository or project).
- 2. Enter the Administration interface (click Administration in the tool view of Navigation).
- 3. If making a global configuration, select Repository in the Open Project or Project Group dialog. If making a project configuration, navigate to and select the Project in the Open Project or Project Group dialog. (See: Access Projects.)
- 4. In the Navigation pane, expand Work Items and select Live Plan. The Planning page loads in your browser.
- 5. Set the field values as desired and click Save to save your changes. Refer to the Quick Help text on the page for information about the fields.

## Split Work Items across time and dependencies

There is a per-project option in the Planning configuration, Enable Work Item Splitting, that defines whether Work Items in a project can be split for project planning purposes. The option is enabled by default.

If enabled, then when Polarion searches for a space for a Work Item in the project plan, it may decide to split the Work Item into two or more pieces and thus make use of periods of time before the Work Item is fixed or postponed by a planning constraint. The resulting plan is more optimal in terms of priorities. This approach ensures that dependencies are handled.

The Live Plan visualization shows both split and nested Work Items.

## Due date planning option

The Enable Due Date Planning option controls the way in which the project planning engine processes and renders the due date of Work Items in the Live Plan. The option is disabled by default, and is available in the global scope only. Enabling the option has the following effect:

- Due date will be considered a suggestion for planning, not rule, meaning that it is considered by the planning engine when determining the order of items in the plan, but the produced plan can still violate the due date constraint – which will always be indicated in the Live Plan visualization and **Due Date** column in the **Table** view of **Work Items**.
- When determining a Work Item's best order based on due dates, any item that has a due date defined (including having a Time Point - see note below) will take precedence before any Work Item that does not have such date or time point defined. This implies that good practice is providing due dates (or time points) for all items when this option is turned on. Items without a due date are automatically penalized regardless of their priority.
- Items without a due date specified will inherit the due date from their parent, if it can be determined. As such this makes an exception/correction to point 2 above. A due date defined at the child item always takes precedence, regardless of whether it is later or earlier than the parent due date.

## **Note**

Wherever **due date** is mentioned, it means **time point or due date**, as time points are internally interpreted as due dates. All items above apply for items from projects that have the **Enable Due Date Planning** option turned on.

# **Configure Work Item linking**

There are three main configurations that affect the linking of Work Items:

- Work Item Link Roles and Rules
- Default state of the Suspect attribute of Work Items
- External Linking

#### Link Roles and Rules

You can configure **link roles** that define the relationship between linked Work Items. For example, a link role **depends on** can be specified for a link between Work Items to indicate that the source item depends on the target item. When you define such a link role, you also define its opposite role. For example, the **depends on** link role could have an opposite of is **depended on by**. This link role will be seen by anyone accessing the target item.

Link roles can be configured globally for any repository, and for individual projects to override the global configuration. You can define as many link roles as you need for your process. Project templates may predefine project-scope link roles.

Each link role definition can have one or more *rules*. Rules control which type or types of Work Items can be linked to which other type or types with the selected role. For example, a link role **verifies** might have a rule that in effect says that the role can only be applied from a Test Case type Work Item to a Requirement type Work Item.

## Tip

Defining rules for link roles is extremely important because they are reflected in the user interface elements in Work Items for creating links. Rules prevent users who may not understand the various link roles and the directionality of linking from creating inappropriate links which result in erroneous data in the system that is then reflected in dashboards and reports.

## Access the configuration:

Link roles are an enumeration. The configuration settings are stored in the workitem-link-role-enum.xml configuration files. A graphical editor is provided for editing both link roles, and rules for each defined link role.

- 1. Log in to the portal with administrator permissions for the scope you want to configure.
- 2. Enter the **Administration** interface (if not already there from a previous login)
- 3. To configure the global scope, select **Repository** in the **Open Project or Project Group** dialog.

To configure a project, select the project to configure in the same dialog.

(See: Access Projects.)

- 4. In the **Navigation** pane, expand **Work Items** and select **Enumerations**. The Enumerations page loads in the Content Pane.
- 5. In the Enumerations table, locate the row with workitem-link-role-enum.xml in the Name column.
- 6. If configuring the Repository scope, click the **Edit** link in the **Actions** column. If configuring a project, what you click depends on whether or not a project configuration already exists.
  - If a project configuration does exist, the Edit link appears in the Actions column, and you should click it to access the Enumeration Designer.
  - If no project configuration exists, the Create link appears in the Actions column, and you should click that to access the Enumeration Designer.

#### **Define Link Roles:**

The Enumeration Designer presents two sections: one for the link role definitions (the top section) and one for rules. In the definitions section, you can:

- Modify the properties of any existing link role.
- Remove any existing link role (click the in the **Actions** column).
- Add a new link role (click the icon in the **Actions** column.

## Tip

- **ID**, **Name**, and **Opposite Name** are required. Spaces are allowed.
- Do not use spaces in the **ID** and limit characters to ASCII upper and lower case letters, and numbers, and underscore.
- Make sure the value of **Opposite Name** is meaningful. Look at the default global configuration to see how the concept should work. For example if you were to define a custom link role named amends, a logical opposite name would be is amended by.
- **Default**, if checked, will result in that link role appearing first in the list of of roles in the Linked Work Items section of the Work Item form.

(If this remains unchecked for all roles, then the first role defined appears first in that list.)

## **Define Rules:**

The Rules for link role section of the Enumeration Designer is where you define rules for the various link roles. Here you can:

Define new rules.

- Modify existing rules.
- Delete existing rules.

# Warning

New and modified rules are not applied to existing links between Work Items. If you have links that predate the rules feature, or which existed before rules were configured on your system, the system will not generate any errors if any of these links violate link role rules, and the system will still contain links which now violate rules.

To access the rules for a link role, click inside one of the fields in its row in the top section of the page. The row is highlighted and any existing rules appear in the rules section. Any new rules you define are applied to the selected link role. A single link role may have multiple rules.

A rule says, in essence, that a link with the selected role is allowed from one or more Work Item types to one or more Work Item types. Accordingly, each rule has a property From Types and To Types. Both are multi-valued, since a rules may affect multiple Work Item types. A rule may permit links from a single type to multiple types, from multiple types to a single type, from multiple types to multiple types, or from a single type to the same type only. (The **Same Type** check box can be used for the latter).



Example of rules for a derived from link role.

One type to same type.	One type to any other.
One to multiple specified types.	One type to one type.

#### **Note**

You can create a rule for links that go from a Work Item type in the current project to a Work Item type defined in another project or group. To do that, select **other** in the **To Types** field and enter the ID of the Work Item type in the desired project or group. A typical scenario for this is where e.g. a Requirement type is defined in the current project and a Test Case type is defined in a different project.

# **Configure Work Item linking menus:**

The linking menus of Work Items can be configured to define menu items for creating new linking/linked Work Items while working with an existing Work Item. Among other things, the role of created links can be defined, so that when the end user creates a new Work Item using the menus, only links with correct roles can be created.

The menus are located on the toolbar of the Work Item form in the Work Item Viewer/Editor pane in the Table view of the Work Items topic.

## To access the configuration:

- 1. Log in with administrator permissions for the scope you want to configure (repository or project).
- 2. Open the repository or project you want to configure and enter the \*Administration interface if you are not already in it.
- 3. In Navigation, expand Work Items and select Form Menus. The Form Menus configuration page loads in your browser.

At this point you can create a new configuration for Linked Menus and Linking Menus, or update or delete existing configurations, if any.

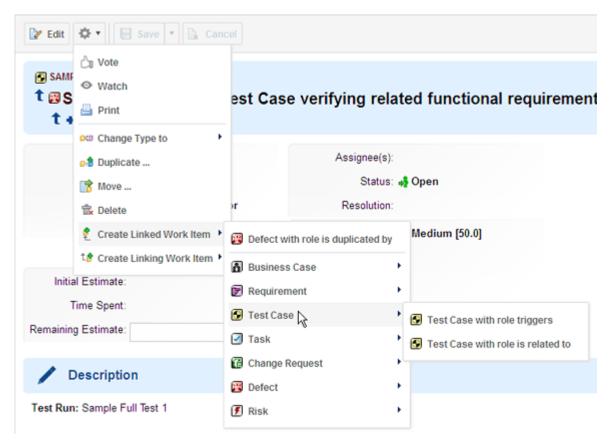
When Creating a new configuration, you select the Interface in which the configured menu will appear, and the type of Work Item that can have the menu item. Unspecific means the menu item is available for any type.

Menu configuration is done using XML in an online editor. For new configurations, an example is provided in comments. Some help is embedded on the menu configuration pages.

Reference information for this configuration is provided in embedded Quick Help on the administration pages in the portal.

#### Rules in the Work Item interface:

Link role rules control the items reflected in the Action menus in the Work Item form. Users creating links to other Work Items can only select valid link roles because, thanks to the rule definitions, only valid link roles are presented in the menus.



Users can only select a link role that is valid for the current Work Item.

## Link with the opposite Role:

End users can link Work Items with an applicable Link Role or its opposite, so as not to have to switch between different Work Items in order to link with the desired relationship. Backlinks from the same project can be added or removed. When adding backlinks, the Role list shows forward link roles first, followed by appropriately filtered backlink roles from the current project. If the picker dialog is used, it presets the correct query based on the current project's configuration.

When linking with the opposite role, there are some limitations:

- Opposite link can only be created/editing in an individual Work Item (that is, not using Bulk Edit or Multi Edit).
- Opposite links cannot be created in the Document Editor. User must open Document-contained items in the Table view of Work Items.
- Backlinks from different projects cannot be edited.

## Suspect default state

You can configure the default state of the Suspect attribute of Work Items.

This configuration is available for both global (repository) and Project scope. If you are not familiar with the basics of the different scopes, you may want to review Administration Basics: The Administration Interface.

## To access the configuration option:

- 1. Log in with administrative permissions for the scope you want to configure (repository and/or Project).
- 2. Enter the Administration interface and open the repository or project you want to configure. (See: Access Projects.)
- 3. In the **Navigation** pane, expand the **Work Items** node and select **Linking**. The Linking administration page appears.
- To have the suspect attribute automatically set to true for all new Work Items, check the Auto Suspect box. Clear the check box if it is checked and you do not want new Work Items to automatically have the Suspect attribute automatically set.
- 5. Click the **Save** button to save your change.

Remember that projects inherit the global configuration. If you want to override the global Auto-suspect setting for a project, click the **Create** button on the Linking page toolbar when you are working in a project, then set the options as you want them for the project. If you want to revert from an existing project configuration back to the global repository configuration, click the **Delete** button to delete the project-scope configuration.

## **Enabling linking across servers**

In a clustered server environment with multiple Polarion servers, if any of the servers host their own Polarion repository, it is possible to link a Work Item hosted on one server's repository to another Work Item hosted on a different server's repository. For more information, see Link to Remote Work Items.

# **Configure Auto-assignment**

You can configure Polarion to automatically assign new or existing Work Items to one or more users based on some condition(s). This is an important feature that can save many hours of work over time. It is recommended that you take the time to become familiar with it, and then set up a default configuration for your system, and specific configurations for projects based on the their individual characteristics.

You can configure one or more Auto-assignment Rules which specify the conditions a Work Item must meet in order for it to be automatically assigned, and the user or users to which it is assigned if the conditions are met. Rules can be defined which apply to all Work Item types, or to one specific type. For example, you might define a rule that says any type of Work Item with some value of "usability" in the Category field is automatically assigned to some user in charge of usability issues.

You can optionally select current user to have new items assigned to the currently logged-in user. By default, current assignee would be applied only if the user is among the possible assignees for the current context. You might want to constrain this further with one or more rule conditions.

Auto-assignment is not active by default. This configuration is available for both global (repository) and Project scope. If you are not familiar with the basics of the different scopes, you may want to review Administration Basics: The Administration Interface.

# Access the configuration

1. Log in with administrative rights for the scope you want to configure (repository or Project).

- 2. Enter Administration .
- 3. If making a global configuration, select **Repository** in the **Open Project or Project Group** dialog. If making a project configuration, navigate to and select the Project in the Open **Project or Project Group** dialog.

(See: Access Projects.)

4. In the **Navigation** pane, expand the **Work Items** node and select **Auto-assignment**. The Auto-assignment page displays a table of the existing auto-assignment rules, if any.

## Creat a new rule

- Access the configuration in the desired scope, as described in the previous section.
- If no rules exist for the current scope, click the **Add New Rule** button to add an auto-assignment rule. If one or more rules exist, click the **Add Rule Below** in the row below which the new rule should appear in the table of rules. (Rules are processed in the order listed in the table).

The **Auto-assignment Configuration** page appears.

- In the **Conditions** section, specify the conditions which must be met in order for Work Items to be automatically assigned. Refer to Quick Help embedded in the page.
- In the **Assignees** section, specify one or more users to whom Work Items will be assigned which the conditions specified in **Conditions**.
- Click the Save button. If you want to create or edit another rule, click the link Return to Auto-assignment.

## Modify an existing rule

You can modify existing rules in either the repository or project configuration. A common reason for changing a rule is that the assignee(s) need to be changed due to changes in personnel, responsibilities, etc.

- 1. Access the Auto-assignment topic in Administration as described earlier. Be sure you have opened it in the scope you want to configure.
- 2. In the table of existing Auto-assignment rules, click the Edit button. The **Auto-assignment Configuration page** loads.
- 3. Edit the Conditions and or Assignees as needed.
- 4. Click the **Save** button. If you want to create or edit another rule, click the link **Return to Auto-assignment**.

## Remove a rule

1. Access the Auto-assignment topic in Administration as described earlier. Be sure you have opened it in the scope you want to configure.

2. In the table of existing Auto-assignment rules, click the **Remove** button and respond to the confirmation prompt.

## Warning

If you have administration permissions for the Repository, be sure you are removing a rule in the correct scope. For example, don't remove a rule in the Repository scope thinking you are doing so for a project.

# **Configure Voting**

You can configure the Work Item Voting feature to enable or disable it (it is enabled by default) and/or set the per-user limit of votes on Work Items in projects.

This configuration is available for both global (repository) and Project scope. If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface. The configuration is specified in the voting.xml configuration file. You will find comments in this file that explain the elements and their usage.

To access the voting.xml configuration file:

- 1. Log in with administrative rights for the scope you want to configure (repository and/or Project).
- Enter Administration.
- 3. If making a global configuration, select **Repository** in the **Open Project or Project Group** dialog. If making a project configuration, navigate to and select the Project in the Open Project or Project Group dialog.

(See: Access Projects.)

- 4. In the **Navigation** pane, expand the **Work Items** node and select **Voting**.
- 5. Use the appropriate link in the **Configuration** section to download the configuration file to your local system, or edit online with the text editor provided.
- 6. If you edit the file offline, use the controls in the **Upload New [Project] Configuration** section to upload the modified file back to the Polarion repository.

# **Configure Multi-language support**

An administrator can optionally configure Polarion to support multiple human language translations of Work Items. When correctly configured, a Title and Description field for one or more additional languages appear in the Work Item form in tracker views, and users can switch languages in Documents to view translated Work Items.

For information on this configuration, please see: Configure Languages.

See also: View Work Item Translations.

# **Configure Testing**

# **Configure Testing overview**

# ALM REQ QA

Administrators can configure Test Management features both globally for the repository to create a default configuration, and in any project to override global defaults and create a custom configuration for the project.

This configuration specifies mapping of test cases to a configured Work Item type, parameters for automated testing, and how Polarion imports results of external testing tools. It also specifies the values for several enumerations which appear in various lists in different configuration fields. You can configure:

- The general testing configuration
- Values that describe different test results
- Templates for Test Run export to Microsoft Excel.
- Custom fields for Test Runs
- Test Run types

# Access the configuration

- 1. Log in with administrator permissions for the scope you want to configure (repository/global or project).
- 2. Enter Administration (if your login does not take you into Administration).
- 3. Open the repository or project you wish to configure (see: Access Projects).
- 4. In Navigation, expand the **Testing** node to expose the testing configuration topics.

# **Configure Types and Enumerations**

## **Configure Types and Enumerations overview**

# ALM QA

Before performing the main testing configuration, you should review the settings in the enumeration topics and make any changes. The values you specify in these topics are used in lists in the main testing configuration topic.

## **Test results**

The **Test Results** topic enables you to specify a set of values (called and "enumeration") that define different outcomes or results of testing. For example, you specify if the result of a successful Test Run execution should be labeled "Passed", "Succeeded" or something else. You should define a value for every possible result of your test execution process. An example of these values might be:

- Passed (result value for tests which succeed)
- Failed (result value for tests which fail)
- Error (result value for tests which succeed but with errors)
- Blocked (result value for tests which could be run for some reason)

You can specify an existing icon for any value, or upload a custom icon by clicking the **Select** button on the relevant row.

The Create Defect column, if checked, means that a Work Item representing a defect will be created when a execution of a Test Run terminates with the result defined on the row. For example, you would probably check this option for a Test Result representing test failure.

You can add a new Test Result value to the table by filling in the last row. You can add more Test Result values to the table by adding a new empty row by clicking the • icon. You can remove a Test Result value by clicking the **-** icon on the relevant row.

Be sure to click **Save** when finished, before navigating to some other topic.

## **Test Run custom fields**

This topic enables you to define custom fields which should be added to Test Runs to store information from external testing tool results or manual testing results. For example, if your tool returns a datum "error code" that you want to track in testing results, you can add a custom field e.g. Error code.

You can add as many fields as are needed, one field per row of the table. Add more rows using the • icon.

The Multi field, when checked, enables multiple values to be entered into the field. This option is only enabled when the field type selected in **Type** supports multiple values.

The **Require** field, if checked, means that the field is required and changes to the Test Run cannot be saved unless a value is supplied. This is more applicable when Test Runs are executed manually, or when results of automated tests are logged manually.

## **Test Run Statuses**

This topic enables you to define the different status values which Test Runs can have, including a standard or custom icon image to represent the status value in the user interface.

The Terminal option marks that the status is a terminal one, meaning that testing activity is complete. If you set a Test Run to a status marked Terminal, the Finished on field value will be filled in with the current date-time.

See also: Customize Test Run Templates.

# **Test Run Types**

This topic enables you to specify values that define the type of a new Test Run. Typical values include Manual and Automatic. You can change the semantics or add more values to cover all the types of Test Runs you need to create to support your testing process. You can optionally require that users log an electronic signature for executions of Test Runs of any or all of the configured types.

# Set the testing configuration

# Set testing configuration overview

# ALM QA

The Configuration topic ( Administration Testing Configuration) is where you specify how Polarion's Test Case Management feature should work. For field-specific help, please refer to the Quick Help text at the bottom of the Configuration page. This configuration is available in the Global (repository) scope, and in projects. Project settings override the global settings.

# The main artifacts of the testing process are:

• **Test Cases**: These Work Items define a specific test or set of test steps. Test Case items can be linked to e.g. the requirement type item(s) they verify. For test management purposes, the global configuration and each project configuration should define a Work Item type for test cases. Some Polarion project templates are pre-configured to provide a type named **Test Case**.

In project templates (and projects based on them) Test Case type is the type for automated tests. It is used to define the link roles and custom fields. The assumption is that all the test-case-like types have the same custom fields used in the testing configuration and the same link role type between a Defect and Test Case.

You may need to define other Work Item types for different types of test cases: acceptance test case, functional test case, etc.

You can do this globally or in any project in Administration Work Items Enumerations Workitem-type-enum.xml.

• **Test Runs**: These Work Items represent an instance of execution of one Test Case, or more usually, a set of Test Cases, and the result of the execution... test(s) passed, failed, etc.

The Polarion project template for test case management projects is pre-configured to include the Test Run type.

Defects: These Work Items record and describe a defect. Defects may be created as the result of
a Test Run, either manually by testers executing tests manually, or automatically by a test result
import job configured to import the results of external automated tests. For test management
purposes, the global configuration and each project should define a Work Item type for defects.
(Most Polarion project templates are pre-configured to provide a type named "Defect".)

# The Testing Configuration page contains settings that specify a number of properties for Test Runs:

- You need to specify which configured Work Item type Test Runs should link to as test cases.
   For example, a global configuration might contain several Work Items types that are analogous to different types of test cases that are run in different types of projects. A project configuration needs to specify which type should be used by project Test Runs.
- You can specify templates for several purposes including default for Test Runs, Test Cases, and Defects. Generally you select an existing Test Run, Test Case, or Defect to serve as the template of others that get created during the testing process.

- You can specify how Test Runs map passed tests, failed tests, and errors to values in the Test Results enumeration.
- Defects are created as a result of failed tests. You can specify properties of such Defects including which Work Item type to create to represent a defect, the link role between Defects and Test Cases, field data from Test Cases and Test Runs copied to Defects, etc. Again, refer to the embedded Quick Help text for details.

## **Migrate Test Steps button**

#### **Note**

This toolbar button is present only if the system property com.polarion.hideMigrationTestStepsButton is set to false.

(polarion.properties file location: Windows - polarion/configuration/polarion.properties, Linux - %POLARION HOME%/etc/polarion.properties)

In earlier implementations of the Test Steps feature, manual test steps were created and stored in the **Description** field of Work Items. In later implementations (beginning with version 2012-SR3) test steps are created and stored in a dedicated custom field of type Test Steps. The Migrate Test Steps button enables an administrator to move existing test steps data from the Description field to the new Test Steps field of Work Items. Migration is available only in the project scope and is performed on Work Items of the type specified by the administrator in the Migrate Test Steps dialog. (This would normally be the Test Case type.)

Before running the migration utility in a project having existing Test Steps data in Description fields, you must create a corresponding Test Steps configuration in (\*Administration \* Testing Test Steps). Names of the Test Step Columns in the configuration must be the same as the column labels in existing Test Steps tables (in the **Description** field of Work Items). You must also configure a custom field of type Test Steps for the Test Case type Work Item. For complete information please download the Polarion Test Steps Migration Guide.

## Automated testing

This section defines parameters used for importing results of automated tests (thereby enabling tracking of those results).

## **Test Case type**

Defines which of the configured Work Item types corresponds to a test case.

(Work Item Types are configured in **Administration** Work Items Enumerations.)

## **Default Test Run template**

Defines which Test Run to use as a template for creating new Test Runs when importing results of automated tests. (May be overridden in a job or build configuration.) Enter the ID of an existing Test Run from the current project.

## **Default Test Case template**

Defines which Work Item to use as a template for creating new Test Case Work Items while importing results of automated tests. (May be overridden in job or build configuration.) Enter the ID of a Work Item from current project or projectID/workItemID for Work Items from other project.

## **Custom field for Test Case ID**

Defines which string-type Custom Field (of Test Case type Work Items) to use for storing the ID string of an automated test, the results of which are being imported. It is needed for subsequent imports of the tests in order to recognize Test Case type Work Items created by previous import(s). The value for Java JUnit tests is the name of a specific test class and method.

(Work Item Types are configured in **Administration** → **Work Items** → **Custom Fields**.)

# **Create Summary Defect**

Defines the conditions under which a single "Summary" defect item is created and linked to a Test Run instead of multiple Defects (one for each failed Test Case). It is useful for situations when automated tests fail because of some environmental problem not directly related to the concrete Test Cases, which causes many test failures.

# Map results of automated test to Enumerations

This section allows mapping of automated test results to configured Test Record Result and Test Run Status values in Polarion.

This configuration also specifies for which Test Result(s) a linked Defect item should be created.

Before configuring this mapping, you need to configure the two enumerations: **Test Run Statuses** and **Test Results**, in the corresponding administration topics under **ATesting**.

#### **Note**

When there are multiple types of test results, the Test Run status will be mapped as follows:

- If there are any errors, the Error status is will be set.
- If there are any failures (and no errors), the **Failed** status will be set.
- Otherwise, the Passed status will be set.

## Manual testing

This section provides options and setting applicable to manual testing and manual type Test Runs.

# **Default planning query**

Here you can specify Lucene query syntax for a query that retrieves some subset of all Work Items when a user launches manual selection of Test Cases for a Test Run. For example, such a query might retrieve all Test Case type items excluding all other types, or some subset of Test Case items such as Test Cases of manual type and not in Draft status. The purpose of the query is to limit the

amount of items in the Work Items table to some reasonable starting point for a user who is beginning manual selection of Test Cases for a manual Test Run. The user is not restricted to the results of this query. They can formulate and run any query in the Work Items Table view any time during Test Case selection. However, if users select Test Cases that are outside the scope of the results of this default query, they are notified that they have selected such items and prompted to confirm their selection.

You can use the Select Test Cases item on the Actions menu of a Test Run to select test cases for it if the Select Test Cases field is set to Manual, Query on Create or Document on Create. In a Test Run Template, you can use the action only if the Select Test Cases field is set to Manual.

#### Allow retest of manual Test Case

Defines the conditions under which a single "Summary" defect item is created and linked to a Test Run instead of multiple Defects (one for each failed Test Case). It is useful for situations when automated tests fail because of some environmental problem not directly related to the concrete Test Cases, which causes many test failures.

# **Defect configuration**

## **Create Defect in Project**

Defines the project in which Defects (resulting from failed tests) should be created.

## **Defect Type**

Defines which of the configured Work Item types is used to represent Defects resulting from test

(Work Item Types are configured in **Administration** Work Items Enumerations.)

## **Defect Template**

Defines which Work Item is used as a template for creating a new Defect type Work Item as a result of test failure while importing results of automated tests, or when a manual Test Case is executed. It is also used as the template for the "Summary Defect" (see Create Summary Defect). Enter the ID of a Work Item from current project or projectID/workItemID for Work Items from a different project.

## Link Role

Defines which Link Role is used for linking Defect Work Items (created when test results contain failed tests) to Test Case Work Items.

(Link Roles are configured in Administration Work Items Enumerations.)

#### **Reuse Defects**

Defines when the defect created when a test case fails is reused.

## **Reuse Defect Options:**

## Never:

When set to **Never**, if the current Test Record does not have a linked defect, a new defect is always created. While re-testing, the defect that is already linked to the test record can be reused.

## If failed in group / If failed in previous:

- 1. First **If failed in group** and **If failed in previous** check iterations of the Test Case in the current Test Run. If there is a resolvable, unresolved, not summary defect, it is reused.
- 2. Then they check Test Runs with the same Group as the current Test Run and in the same Project as the current Test Run. If there is a resolvable, unresolved, not summary defect, then it is reused.
- 3. Then, the **if failed in previous** strategy checks Test Runs in the previous group and in the same project as the current test run. (The previous group is determined by searching last Test Run by creation time with a different group.) If there is a resolvable, unresolved, not summary defect, then it is reused.
- 4. Otherwise no defect is reused.

## Always:

- 1. **Always** finds the top 20 Test Records in the Project of the current Test Run, for the same Test Case that have a defect set.
  - (Sorted by execution time in descending order.)
- 2. It then iterates over these defects and finds the first one that is resolvable and not a summary defect of its Test Run.
- 3. If such a defect is found and has no resolution, it is reused.
- 4. Otherwise no defect is reused.

## **Perform Auto-assignment**

Defines if auto-assignment should be executed to automatically fill the Assignee field in new Defect items created automatically for a failed Test Case.

( Administration → Work Items → Auto-assignment.)

## **Allow Retest of Manual Test Case**

Defines whether re-running of manually-executed Test Cases is allowed.

## **Copy Test Case or Test Run fields to Defect**

## **Copy Test Case fields to Defect**

Defines which fields are copied from the Test Case Work Item to automatically created Defect Work Item(s).

## **Copy Test Run fields to Defect**

Defines which fields are copied from Test Run to an automatically created Defect Work Item. When the source field is single-enum field and the target is multi-enum field, then the value is added, otherwise the value in the target field is overwritten.

## To Existing:

When checked then this mapping is used also when existing Defect is linked to new failed Test Record

## **Automated cleaup of Test Runs**

There is a job polarion.jobs.testruns.delete that can be run to clean up old Test Runs. The job runs in the global scope, but it uses configurations from project scopes.

## **Note**

Enabling this option in the global configuration enables it in all projects that do not define their own configuration for automated cleanup of old Test Runs. Test Runs in such projects will be deleted according to the settings in the global configuration, which may or may not be what is wanted. It is advisable make sure that projects subject to regulatory compliance are configured so that no Test Runs important for regulatory reporting are deleted.

You can optionally define a Lucene query in the field Query used by job to select Test Runs for deletion, that returns the set of Test Runs to be deleted by the job. This filtering query is automatically extended by Polarion so that the job never deletes Test Run templates and Test Runs that have the **Keep In History flag** set.

When Test Runs are deleted by the job, no email notifications are generated, and nothing is shown in the Activity stream.

# **Example query**

type:automated AND finishedOn:[19700101 TO \$today-30d\$]

## **Macros for Test Run Pages**

Polarion provides the following macros which can be used to incorporate export-import functionality into Test Run pages:

1. {export-tests} - Renders a link which launches a dialog enabling a user to export of a set of Test Cases to Microsoft Excel. Use the query to retrieve the test cases to be exported. Example: {export-tests:query=status:active|sortby=id}

Excel round-trip export prefers the current context's export template when searching for a default one. (It takes first export template in lexicographical order from the project configuration, then the first in lexicographical order from the global configuration.)

2. {import-test-results} - Renders a link which launches a dialog enabling a user to import a Microsoft Excel file containing test results.

For complete syntax information, see the Syntax Help provided when editing a Test Run page.

## **Configure Test Run ID prefix**

You can configure any Test Run Template to add a prefix at the beginning of the IDs of new Test Runs created from the template. The prefix will be added to both manually specified and generated Test Run IDs.

1. Review the Test Run Templates, and decide which of them you want to have add a prefix to new Test Run IDs.

2. In each template you choose, edit the template Properties and enter a prefix value in the **Test Run ID Prefix** field.

For example, in a template "Software Design Verification Test", you might specify a prefix like "SWDVT" or "SDV".

When manually creating a Test Run, users can select the Test Run Template. If an ID prefix is configured in the template, and the project is not configured to generate Test Run IDs automatically, the prefix is shown as read-only in the **Create Test Run** dialog, before the text box in the ID field.

#### **Generate Test Run IDs**

You can set up the testing configuration to automatically assign IDs to new Test Runs.

When configured, users who manually create new Test Runs do not need to specify an ID, and the naming convention will be consistent for all Test Run IDs.

- 1. Enter Administration for the project you want to configure.
- 2. In Navigation, select Later Testing → Configuration.
- 3. In the Testing Configuration page, check the option **Enable Generated Test Run IDs**.

#### **Note**

This option setting is project-specific. You will need to set it in every project in which you want automatic generation of Test Run IDs.

## **Generated ID Format**

The ID of new Test Runs will begin with the value configured in the *Test Run ID Prefix* field of the Test Run Template (if one has been specified), followed by a dash (-), followed by a unique ID generated by the system. This ID has the following form:

PREFIX-YYYYMMDD-HHMM with a uniqueness suffix (generated only if necessary) in the form of NUM

For example: MYPREFIX-20150721-1032 or MYPREFIX-20150721-1032\_2 - where MYPREFIX is an ID prefix value that has been configured in the Test Run template (not present if no prefix has been configured), and the trailing 2 in the second example is a uniqueness suffix.

When users manually create a new Test Run, if generated IDs are configured, the ID field is hidden in the Create Test Run dialog. The generated ID appears in the new Test Run after it is created, and the ID cannot be changed.

## **Limit notifications for imports**

The first time you import automated test cases (jUnit tests, for example) to Polarion, it is unlikely that you would want notifications sent about newly created Test Case type Work Items. Also, it is probably not necessary to show all the resulting Create activities in the Activity Stream. This is especially true if you are importing a large number of test cases.

An automated test case is a Work Item of the type configured as Test Case, with a non-empty value in the custom field defined as the Test Case ID (see configuration page in \*Administration \*\*

\*\*Esting\*:

There is a system property that defines a threshold above which notifications are not sent, and the Activity Stream is not updated when new automated test case Work Items are created. The default value is 50, meaning that if more than 50 new Test Case items are created, notifications and activity are not triggered.

## If you want to change the default threshold value:

- 1. Using any text editor, open the system properties file polarion.properties (follow link for location).
- 2. Search for com.polarion.maxNotificationsAboutCreatedTestCases. It is not present in the file by default, so you probably need to add it on a new line.
- 3. Set the threshold value as follows: com.polarion.maxNotificationsAboutCreatedTestCases=[VALUE] ...
  - ...where [VALUE] is a positive or negative integer, or zero.

4.

- If [VALUE] is positive, notifications and activity will be suppressed if the number of imported automated tests exceeds [VALUE].
- If [VALUE] is 0 (zero), no notifications will be sent and no activity will be streamed regardless of the number of automated tests imported.
- If [VALUE] is negative, no threshold is applied and all notifications are sent and all activity is streamed.

## Warning

Setting no threshold can result in significant temporary performance degradation of Polarion, your SMTP server, and your network if large numbers of automated tests are imported and no threshold limit is set, to say nothing of user irritation if large numbers of notification emails arrive in their in-boxes as a result of some test case import.

## **Limit Test Run Records in History**

If you have very large manual Test Runs, the default system configuration can result in excessive numbers of Test Run records in the historical database. In such cases, an administrator can set a system property to reduce the number of records generated. See Prevent Excessive Test Run Records for information.

# **Configure Test Run workflow**

Configure Test Run workflow overview



Polarion can support formal testing review and sign-off process for Test Runs, using secure electronic signatures compliant with U.S. 21 CFR Part 11. An administrator can map the organization's review and sign-off process to the Test Run Workflow configuration. Each step in the process can optionally be set up to require an electronic signature before a Test Run can move to the next stage in the workflow

- Initial setup for existing installations
- Use the Test Run workflow designer

## Initial setup for existing installations

If you have updated an existing Polarion system from a previous version that did not support workflow for Test Runs (version 2015 or earlier), you will need to enable the feature in Administration before it can be used with existing projects. This section provides details about the procedure.

You should enable Test Run workflow in project configurations first, then in the global scope. This ensures that existing projects do not inherit the global configuration incorrectly. For example, some projects might have Status definitions that are incompatible with the initial global workflow configuration. During the setup process, Polarion creates an initial workflow configuration for all Test Run types in the scope based on the current configuration of Test Run statuses.

# Warning

Polarion does not support having some Test Run types with workflow, and some types without workflow in the same project. Polarion sets up a default workflow configuration applicable to all Test Run types, which can be overridden by defining workflow for specific types. To avoid problems, do not remove the **All Types** workflow configuration in any scope. You can modify it to be as generic or a broad as you want, and then focus on type-specific configurations.

# **Enable workflow for a Test Runs in a project**

- 1. In project Administration's Navigation, expand Testing and select Test Run Statuses.

  (Only present if the workflow feature is not already set up.)
- 2. If you need to modify the Statuses configuration, do that now. You can still make changes after Test Run workflow is enabled. However, the initial setup is based on the current statuses, so it is preferable to fully define all the Test Run statuses the project needs before proceeding.
- 3. On the toolbar of the Test Runs Statuses topic, click **Set Up Workflow**.
- 4. Follow the messages in the dialog to complete the setup operation.

## **Enable workflow for Test Runs globally**

- 1. Log in with administrator permissions for the **Repository** scope.
- 2. Enter global Administration, and in Navigation expand area and select Test Run Statuses (only present if the workflow feature is not already set up).

- 3. On the toolbar of the Test Runs Statuses topic, click the **Set Up Workflow** button.
- 4. Follow the messages in the dialog to complete the setup operation.

After the setup is complete, the **Test Run Statuses** node in Navigation becomes hidden, and is replaced with the **Test Run Workflow** node, and the Test Run Workflow page opens. On this page you can view and edit the default workflow configuration that Polarion created for all Test Run types, and you can define type-specific workflow for any of the existing Test Run types, which appear listed in the table. If you need to define new types, you can do so.

#### **Alternatives for Advanced Users**

You will get a simple all-allowed workflow upon using the Set Up Workflow action described above. You can optionally install the new default from a project template instead, either by picking it up from a new project created from a corresponding template, or by mining it directly from the plugins and then uploading it to your own project's Subversion repository.

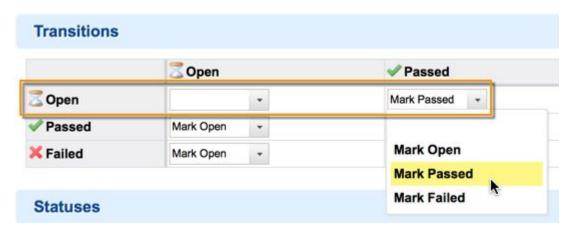
#### **Using the Test Run Workflow Designer**

If you have previously configured workflow for Work Items or Documents, you already know most of what you need to be able to configure it for Test Runs. As with these other workflows.

- In the Statuses section you define every status a Test Run in the scope can potentially have during its lifecycle.
- In the **Actions** section, you define the actions that users invoke to trigger the transition of a Test Run from one Status to a different Status. This is where you, in essence, map your specific process for running tests into Polarion's workflow control.
  - Programmatic workflow functions can be specified and their input parameter values configured. The configured workflow functions are then executed along with the respective transition actions. The execution can be conditional, dependent upon one or more specified workflow Conditions. All conditions must be satisfied in order for the related function to be executed.
- In the **Transitions** section, for each Status in the left column you specify to which other Statuses it can transition, and which of the defined Actions triggers each transition.

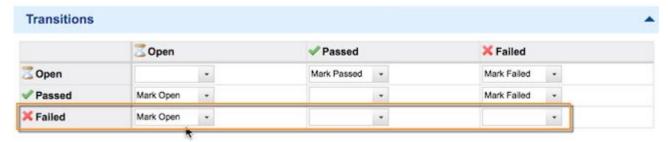
A brief example can help to grasp the concept. Suppose there are only 3 possible Statuses a Test Run can have in your process: Open, Passed, and Failed. The Actions to transition from one Status to another might be specified as: Mark Open, Mark Passed, and Mark Failed. With these defined, you only need to decide to which other Status(es) a Test Run can transition from any given Status.

A Test Run with the Open status cannot transition to that same status. Can it transition from Open to Passed? It obviously can, so in the cell where Open intersects Passed, you select the Mark Passed action. From Open status, a Test Run should also be able to transition to the Failed status, so you would select the Mark Failed action where Open intersects Failed.



Setting workflow transition for a Test Run

The end result for users is that they can only execute valid transitions, and they cannot bypass or skip any process steps. For example, you might decide a Test Run should never directly transition from Failed to Passed, but only from Failed to Open, you would set an action only in the **Open** column:



Transition from Failed is limited to the Open status

Users cannot then bypass the *Open* status and set *Passed*. There can be some important reasons for this beyond mere policy. Transitions can be configured so that certain things happen when they are invoked: field values are set or cleared, some programmatic function executes (such as prompting for electronic signature), and so on.

#### **Defining Test Run Statuses**

Testing managers should analyze their needs and process to determine the statuses a Test Run can have at the various stages of its lifecycle (e.g. "Open", "Planned", "Failed", etc.) Then the administrator can create a Status definition for each one in the Statuses section of the Workflow Designer. Keep in mind that you can have workflow configurations for different Test Run types, which means that different types can have different sets of statuses, if that is what makes sense for the context. For example, Test Runs for quick, experimental testing might have one set of status definitions, while Test Runs for formal regulatory documentation might have a quite different set.

#### **Defining Workflow Actions**

It is in the Actions section of the workflow configuration that you exert the greatest control over the workflow. You can:

- Specify the role(s) that users must be assigned in order to be able to invoke the Action.
- Require that some field(s) be filled before the Action can be executed by specifying field ID(s) in the Required Fields column.

- Specify fields whose values is cleared when the Action executes, by specifying the relevant field ID(s) in the Cleared Fields column.
- Mark an Action as the initial one in the process. Remember that the Initial action, will execute along with its required and cleared fields, workflow conditions, and workflow functions when a new Test Run is created.
- Require an electronic signature from the user invoking the Action before the Action is executed.
- Specify one or more workflow Functions to execute when the Action executes to set a date or run some custom script, for example.
- Specify one or more workflow Conditions to control whether or not the workflow action can be executed. If any of the conditions is not satisfied, users are informed that the corresponding action as unavailable, and they are blocked from executing it. For example, a Condition might check that a specific field is not empty, and the action is not available to users unless that Condition is satisfied.

For reference information about workflow Conditions and Functions, see Workflow Conditions and Functions (Test Runs).

### **Workflow and Imported Test Results**

After Test Run workflow is set up and configured, when test results generated by external testing tools are imported, the **xUnitFileImport** job processes the results according to the workflow configuration. including conditions and functions for workflow actions, if workflow is configured for the project and the Test Run type involved.

# **Configure Test Run Signatures**

#### **Configure Tetst Run Signatures overview**

# ALM QA

Polarion features US FDA-compliant electronic signatures, which can be applied to different artifacts, including Test Runs. When configured, Polarion automatically maintains e-signatures as part of the overall artifact history. It is then easy to retrieve data to prove who signed what and when. Administrators can set configuration to require electronic signatures from users when they:

- Change the Status of a Test Run via the workflow.
- Execute the Test Cases of a Test Run (including importing of results of Test Cases executed externally to Polarion).

#### Warning

For security, it is highly recommended to use the https protocol for access to Polarion when your system is configured to use electronic signatures.

Signatures for Test Run workflow actions

- Marking Signatures as obsolete
- Signatures for Test Case execution

#### Signatures for Test Run workflow actions

The Test Run Workflow page of Administration has a section *Actions*, which defines the set of actions users can invoke to transition a Test Run from one Status to any other allowed Status. The Actions table has a column Requires Signature. When you check this option for any Action, users invoking that Action are prompted to enter their username and password in lieu of their handwritten signature. (The dialog requires two steps in order to require explicit action by the user, preventing browsers from supplying all credentials.)



Requiring e-signature for Test Run actions

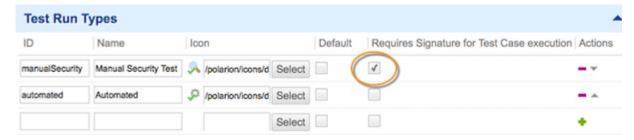
#### Mark Signatures as obsolete

It can happen that after some workflow action has been executed, that another action is invoked that renders existing signature(s) irrelevant. For example, a test manager might decide that a Test Run with failed Test Cases should be executed again, and so sets the status back to the awaiting execution state. In this case, any existing signatures previously applied to actions such as "mark passed" aren't really relevant anymore.

To handle this situation, you can specify the workflow function *MarkWorkflowSignaturesAsObsolete* for the relevant workflow action. In the example above, when the test manager invokes the action (which could be configured as something like "Mark as Reopened"), all signatures from previous Test Run workflow actions are marked as obsolete.

## **Signatures for Test Case execution**

For each Test Run type defined, it is possible to require electronic signature when executing Test Cases in Test Runs of the type. For example, if you have a type, **Manual Security Test**, you can configure it to require a signature when executing Test Runs of that type:

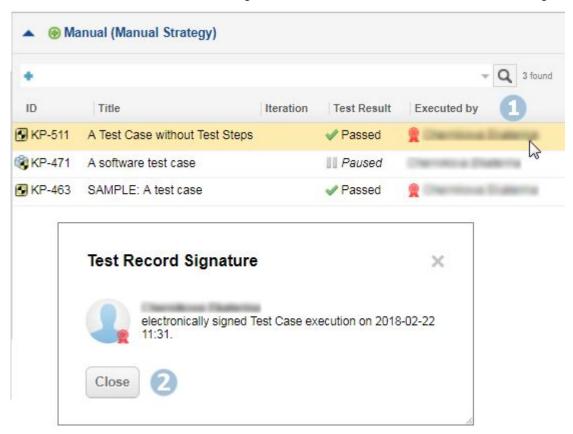


Requiring e-signature for execution of Test Cases in Test Runs

# To require signature for Test Case execution:

- 1. Open the scope you want to configure: Repository or a project.
- 2. Enter Administration, and then in Navigation select Testing Test Run Types.
- 3. In the **Test Run Types** page, check the **Requires Signature for Test Case execution** option for every type for which you want to require Test Case execution signatures.

When this option is enabled, users executing Test Runs of the type configured are prompted to electronically sign when they log the verdict for each Test Case executed during the Test Run. The Test Records show an icon indicating that the user who executed the Test Case signed.

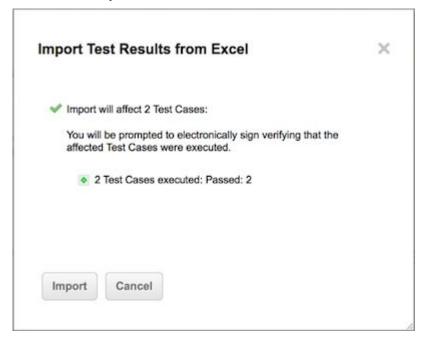


Test Case execution records show who signed.

#### Signatures for Externally Executed Test Cases

When Test Cases are exported to Excel for execution externally to Polarion, it is possible to require electronic signature when a user imports Test Case results from Excel (using the "Import Manual Test Results" action on the Test Run overview page.) Before importing Test Cases from Excel, configure the Test Run type for your external tests to require electronic signature for execution of Test Cases, as described in the previous section. Once configured, you can create Test Runs and export the Test

Cases to Excel as usual. When a user re-imports the Excel file to Polarion, that user is presented with a summary of the executed Test Cases:



When e-signature is required for imported results of external testing.

If the user proceeds with the import, s/he is prompted for electronic signature. The testing results are imported, and the Test Run is updated only after the user completes the signature.

# **Configure Test Export Templates**

# ALM QA

Polarion provides a default template for exporting Test Cases to Microsoft Excel. Administrators can download the default template file which can then be used as the basis for one or more custom templates. An administrator can upload customized template files which users can subsequently select as templates for Excel files generated by exporting Test Cases to Excel. Custom templates can be uploaded in the global scope, to serve as the default for new projects, or in specific projects.

#### Access the test export templates

- 1. Log in with administrator permissions for the scope you want to access (global or project).
- 2. Open the repository if you want to access the global templates, or open the relevant project. (See: Accessing Projects).
- In Navigation, expand are ting and select Test Export Templates.
   The Test Export Templates page loads in your browser displaying the existing templates for the current scope (if any).

#### Download an existing template file

1. On the **Test Export Templates** page, click on the template you want to download.

2. Click the **Download** button on the page header.

#### Upload a custom export template file

- 1. Be sure you are working in the desired scope. For example, if you downloaded a global template and want to use a modified version for a project, be sure to open the project. If you forget that, you will overwrite the global template file with the modified file.
- 2. Open the Test Export Templates page (Navigation: & Testing → 18 Test Export Templates).
- 3. On the page header, click the **Upload** button. In the Upload Export Template dialog, select the template file on your local file system.

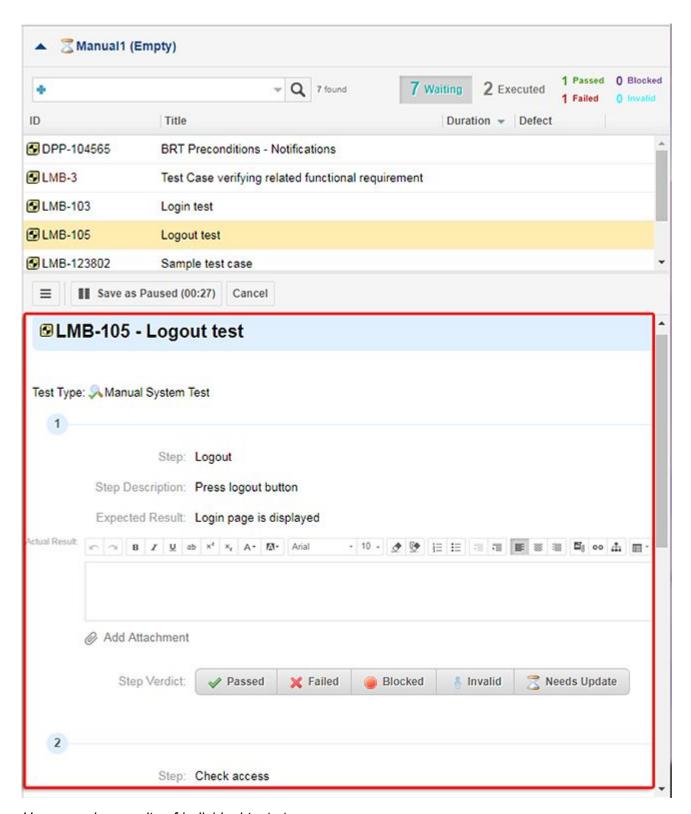
If you want to upload to overwrite an existing template, check this option in the Upload Export Template dialog.

# **Configure manual Test Steps**

**Configure manual Test Steps overview** 

# ALM PRO QA

For manual testing, it is possible for testers to track not only the overall results of a Test Case (passed/failed), but also the results of each test step in a Test Case.



Users can log results of individual test steps

There is a special Work Item field **Test Steps**, to be used in Test Case type Work Items. It is rendered as a table, with each row representing a discrete manual test step. Columns of the table are

user-configurable. You may need to enable test steps in your project configuration if the feature is not already enabled by the project template. This section discusses these configurations.

When correctly configured, a table of test steps appears in the **Custom Fields** section of new Test Case items in which testers can specify each test step (unless an administrator has configured the field to appear separately on the Work Item detail form). This table is rendered as executable test steps in a panel below the Test Run's table of Test Cases. (It has an Execute Test Case heading when Classic execution view is enabled.

#### Tip

- Some project templates delivered with some products may have steps already configured. You can tell by creating a new Test Case in the tracker and checking whether a Test Steps table is present. (You can then cancel the Create Test Case operation without saving.) Even if Test Steps are pre-configured, you may still want to customize the Test Steps table as mentioned in step 4 above.
- You can optionally display Test Steps as a separate field in Test Case items, rather than in the Custom Fields section.

```
To do this, you must edit the form layout for the Test Case type ( Administration
→ Work Items → Form Configuration).
```

If no Form Layout exists for the Test Case type, you can create one. (See: Configure Work Item Form Layout).

In the layout configuration, add the Test Steps field (for example: <field id="testSteps"/>), and remove the Test Steps field from the configured custom fields to be displayed:

```
<panel description="Custom Fields">
    <field id="@allCustomFields,-testType,-testSteps"/>
</panel>
```

#### Migrate existing Test Step data

In earlier implementations of the Test Steps feature, manual test steps were created and stored in the Description field of Test Case type Work Items. In later implementations test steps are created and stored in the dedicated Test Steps field described above. If you have projects in which test steps data are still stored in Description fields, an administrator can migrate the existing data to the **Test Steps** field.

Polarion provides a special migration guide document, which explains how to perform this migration. The guide is available on the Polarion ALM product web site at https://www.polarion.com/hubfs/Docs/Guides and Manuals/Polarion-Test-Steps-Migration-Guide.pdf.

## **Enable Test Steps support**

You can enable the Test Steps feature either globally or per project. You can also add this configuration to any existing custom project templates that for projects that include testing support. This process step requires editing XML configuration code.

## **Configure a Custom Field for Test Steps**

1. In project Administration, open Work Items → Custom Fields.

Add a new custom field to the Test Case Work Item type, with ID testSteps, name Test Steps, and specify the **Test Steps** type. The custom field type "Test Steps" renders a table of test steps in new Test Case items. Note that you can add only one custom field of this type to the configuration.

#### **Enable the Test Records section**

- 1. Log in with administrator permissions for the scope you want to configure (repository or project) and enter Administration.
- 2. In **Navigation**, expand Work Items, select **Form Configuration**, and scroll the page to show the **Form Layouts** section.
- 3. If there is no layout for the Test Case Work Item type, click **Create New Form Layout** and specify the layout as described in **Configure Interface Views**.
  - If the table of layouts does contain a layout for the Test Case type (or whatever custom type of yours corresponds to a test case), click the Edit button on its row in the table.
- 4. If the following element is not already present, add it below the execute-test extension element:

```
<extension counts="1,5,10" id="test-records" label="Test Records" />
```

5. Save the configuration. Note that you need to repeat this configuration in every Interface View that has a form layout for Test Cases, and in which you want the Test Steps feature enabled for Test Cases.

#### **Note**

For Projects created prior to Polarion 12.3:

Contact Technical Support (GTAC), if you have projects created prior to Polarion 12.3 that you want to Execute a Test Run Using Polarion for.

#### **Configure Test Step columns**

The table of the Test Steps field contains some columns controlled by Polarion (**Actual Result**, for example) and some columns that you can modify. The user-modifiable columns (only) are accessible in a dedicated Administration page. There, you can customize the table, adding additional columns, and adjusting headings of the user-modifiable columns to use your own semantics.

### Access the Test Steps columns configuration

- 1. Log in to the project with administrator permissions and enter **Administration**.
- 2. In **Navigation**, expand **Lesting** and select **Lest Steps**.

Refer to the embedded Quick Help text on the configuration page for information about how to perform the configuration.

#### Tip

If your users write up test case specifications in LiveDoc documents, it will be necessary for them to set the Document's Work Item Presentation to show the Test Steps field in Document-based Work Items. For more information, see Edit Test Steps in Documents.

### Control time tracking for test execution

By default, Polarion tracks time expended on executing manual tests and includes it in Test Records, which are visible to all users with the necessary view permissions. There are some jurisdictions that prohibit the visibility of tracking of individuals' time. If you need to stop the tracking and display of time spent executing manual tests in order to comply with local laws, there is a special system property you can use: com.polarion.alm.ui.forms.extensions.disableTimeSpentRecording. If set to "true", then time spent during manual test execution is not stored within test records. (Automated test results import still imports the duration of tests.)

To disable time tracking of manual tests, add the system property disable TimeSpentRecording to the system configuration file polarion.properties (follow link if you need the location). Set the property to the value to "true", and restart the Polarion server after the change.

**Example**: com.polarion.alm.ui.forms.extensions.disableTimeSpentRecording=true

# **Display Recent results**

By default, each Test Step displays the **Recent** drop-down, which lists the Actual Result values from the most recent execution of the step during a Test Run. The Test Case Verdict field displays a similar drop-down with the values of the most recent verdicts. Users can select one of the listed previous results as the result of the current step execution or current Test Case verdict.

You can control how many recent values are displayed in these drop-down lists, or hide the lists completely (not recommended) using the maxRecentItems option in the Test Steps extension element of the Work Item form layout configuration for Test Case type Work Items.

Configuring the number of **Recent** Test Records Polarion should analyze:

Administrators can define the maximum number of test records to analyze by adding the following property in the polarion.properties file:

```
com.polarion.alm.ui.forms.extensions.testExtensionRecordsLimit = 20
```

(The default value is 20)

Polarion starts with the testExtensionRecordsLimit value, eliminates any duplicates or empty results, then displays the maximum number of results defined in the maxRecentItems property.

#### Configure the number of maxRecentItems to display in the Recent drop-down box

- 1. Enter Administration in the scope you want to configure (Global or project).
- 2. In Navigation, expand Work Items and select Form Configuration.
- 3. In the **Form Layouts** section of the page, locate the table row for the Test Case type Work Item, and click Edit.

4. Locate the following line in the configuration editor:

```
<extension id="execute-test" label="Execute Test"/>
```

5. Add the maxRecentItems option to the line, specifying the maximum number of unique recent items to display to use in the drop-down lists, or 0 (zero) to hide the lists. For example:

```
<extension id="execute-test" label="Execute Test" maxRecentItems="10" />
```

#### **Note**

Some project configurations may contain several different types of test cases: System Test Case, Software Test Case, Mechanical Test Case, etc. You will need to perform the above configuration for each of the test case types, assuming the form layout contains the extension element enabling manual Test Steps.

# **Configure cleanup of old Test Runs**

There is a job polarion.jobs.testruns.delete that deletes old Test Runs to free up system resources and speed up system re-indexing. By default, this job will not run. An administrator can enable it, and configure how the job selects Test Runs for deletion, in the global configuration and/or any project configuration. In Administration Testing Configuration: Automated Cleanup of Test Runs.

When enabling the job, the administrator may specify a query that returns the Test Runs to be deleted by the job. (See the embedded Help in the Administration topic for more information and an example of such a query.) When the job is enabled in the global scope, and no query is specified, the system will run a default query that selects for deletion all Test Runs in all projects without a project-scope configuration. Test Runs that are Test Run templates, and Test Runs having the <code>Keep in History</code> option set on them will always be ignored by the job and will never be deleted. When Test Runs are deleted by the job, no email notifications are generated, and nothing is shown in the Activity stream.

Automated Cleanup of Test Runs can be enabled and configured in projects to gain more precise control of how the job deletes old Test Runs. Again, the administrator can specify a query, and it will apply only to the current project. As in the global scope, the job will never delete any Test Runs that have the Keep in History or the Is Template option set, even if the Test Run meets the search criteria in other respects. (Note the Keep in History can be set in Test Run templates.) It is therefore not necessary to include these as elements of the selection query. Be sure that users know to set Keep in History for Test Runs that should have results preserved for traceability and proof of compliance, and check for existing Test Runs that may need to have this option set before enabling the job and deleting old Test Runs.

Remember that all projects that do *not* have a project configuration for clean-up of old Test Runs will have Test Runs deleted according to the *global* configuration settings in the **Automated Cleanup of Test Runs** topic of Administration. This could be a problem if such projects contain Test Runs that need to be preserved for proof of regulatory compliance, but to not have the <code>Keep in History</code> option set. Be sure to do some checking before enabling the cleanup job in the global configuration.

If you want to schedule the cleanup job to start automatically, note that it cannot run on behalf of system user because that user does not have write access to the Subversion repository. There is a parameter for the job, *userAccountValultKey* that can point to a key in the User Account Vault so that the job will run on behalf of this configured user, which should have the required SVN access.

Generally, the same configured user should also be used for xUnit import and automatic creation of Test Runs. If you have this configured, just use the same user for the Test Run cleanup job.

```
<job cronExpression="0 0 6 ? * SUN" disabled="false"</pre>
id="polarion.jobs.testruns.delete" name="Cleanup of Old Test Runs"
                                                                       scope="system">
  <userAccountVaultKey>xUnitImporterUser</userAccountVaultKey>
</job>
```

#### Warning

Before running the job for the first time, be sure to check with project leaders to make sure that they have set the Keep in History flag on all Test Runs that should be preserved in Baselines for compliance reporting and other purposes.

For more information, see: Preserve Test Runs in History.

# **Configure Plans**

#### **Enable and administer Plans**

# ALM

Polarion provides special pages called Plans to support planning and execution of releases and iterations. Users can create Plans in projects, set dates, add Work Items, and set properties than enable Polarion to monitor and report the status and progress of Plans. (For more information, see: Plans Topic.

If your system is updated from a Polarion version that did not yet have the Plans feature, you may need to enable it in administration. There also are several ways an administrator can configure/customize Plans:

- Define Plan custom fields.
- Define Plan statuses and status semantics.
- Configure Plan Templates.

The following sections discuss these topics.

#### **Enable Plans**

After updating your system to a version that includes the feature, from a version that did not, you can easily tell whether or not the Plans feature is enabled. In project Administration, go to Navigation and select Plans Plan Templates. If there are no templates listed, and you see a button Set Up Planning, the feature is not yet enabled. Click the button to launch the enabling process. After the process is complete, the default Plan Templates appear on the Plan Templates page.

#### **Define Plan custom fields**

Scope: Global and Project

You can create custom data fields for use in Plans only. (Plan custom field should not be confused with Work Item custom fields, which are defined for, and appear only in Work Items, or with Document custom fields, which are define for, and appear only in Documents.) You can create Plan custom fields to track any custom data you want to have associated with Plans. Plan custom fields can be of any supported data type. You can select from a list of types when you create the fields. If you want users to be able to select from a list of values for a Plan custom field, don't forget that you need to create a custom enumeration before defining the Plan custom field. (For more information, see Configure Enumerations.)

Although Plans themselves can only be created in projects, you can define Plan custom fields in the Global (repository) scope. Such fields defined globally appear by default in new Plans and Plan Templates created in projects. Project configurations can optionally override the global configuration, changing the type, semantics, or removing the field(s) from the project entirely. Plan custom fields can also be defined in projects, where they will apply only to the specific project.

- 1. Open the scope in which you want to define the Plan custom field(s)... Global(repository) or a specific project.
- Enter ► Administration, and in Navigation select ► Plan Custom Fields.
   The configuration page opens and shows a table of existing Plan custom fields, if any, and an empty table row in which you can define a new field.
- 3. In the empty row, fill in all the fields and options, which are the same as for Work Item custom fields.
- 4. To add another field definition, click the ♣ icon on the last row of the table. A new empty row is added in which you can define another Plan custom field.
- 5. Click the **Save** button when you have finished defining your Plan custom fields.

The configured Plan custom fields appear in the **Custom Fields** section of Plan and Plan Template pages, and values for them can be added or modified in the Plan properties or Plan Template properties.

#### **Define Plan Statuses**

Scope: Global and Project

Polarion provides a special field **Plan Status** for tracking the status of Plans. Users and reports can run queries for Plans with a specified value or values in this field. The default Plan Templates provide 3 Plan Status definitions: **Open**, **In Progress**, and **Done**. You can change the labels and other properties of these statuses (e.g. **Building**, **Executing**, **Completed**) and/or add more statuses (e.g. **Open**, **In Review**, **In Progress**, **Done**).

It is important to understand the Plan Status is *not* a workflow control like the Status field of Work Items. There are no actions or transitions. It is simply data that provide a way of tracking and communicating the current status of a Plan to stakeholders.

Although Plans themselves can only be created in projects, you can define Plan Statuses in the Global (repository) scope. The globally defined statuses appear by default in new Plans and Plan Templates created in projects. Project configurations can optionally override the global configuration,

changing the semantics, or adding or removing statuses. Plan Statuses can also be defined in projects, where they will apply only to the specific project.

- 1. Open the scope in which you want to define the Plan custom field(s)... Global(repository) or a specific project.
- 2. Enter 

  Administration, and in Navigation select 

  Plans → Plan Statuses.

The configuration page opens and shows a table of existing Plan Statuses, and an empty table row in which you can define a new Plan Status.

3. Edit existing statuses, add new statuses, or remove existing statuses as desired and click the **Save** button.

Most of the Plan Status properties are self-explanatory. The following may be worth noting:

- **Default**: Check the Status that should be the initial one applied to new Plans. For example, the "Open" status can be set as the default initial status for new Plans by checking this property on the relevant row in Plan Statuses. When a Plan's status changes to from the Default status to any other, the Started On property of the Plan is automatically set (to the current date).
- **Terminal**: Check the status that should indicate that a Plan is finished. For example, the default "Done" status can be set as the default end status for new Plans by checking this property on the relevant row in Plan Statuses. When a Plan's status changes to the Terminal status, the Finished On property of the Plan is automatically set (to the current date).

# **Administer Plan templates**

Scope: Project only

Plan Templates are special versions of Plans that serve as a template from which actual Plans are created. Values set in Plan Template properties become the default values for new Plans created by users and based on the Plan Template. Plan Templates can be created only in the project scope. To access a project's Plan Templates, open Administration Plans Plan Templates.

## In Plan Templates administration you can:

- Create new Plan Templates.
- Browse the list of existing Plan Templates.
- Review and modify the properties of an existing Plan Template, including the Plan report Page.
- Delete existing Plan Templates.

#### Create a New Plan Template

- 1. Open the project for which you want to create a Plan Template. (Plan Templates are per project only. You cannot create them in the Global scope.)
- 2. Open **FAdministration** → IIIPlans → Plan Templates.
- 3. On the toolbar of the upper pane, click the **Create** button.

The **Create New Template** dialog opens and presents a list of existing Plan Templates. If no custom templates have yet been defined in the project, only the default templates are listed.

- 4. In the **Create New Template** dialog, select a Plan Template as the basis for the new one you are creating.
- 5. Enter a name and ID in the dialog, and click Create.

#### **View and Edit Plan Template Properties**

- 1. Open a project, and enter **Administration**.
- 2. In Navigation, select **■Plans** → Plan Templates.
- 3. Browse the table of existing Plan Templates (upper pane), and select the one you want to review and/or modify.
- 4. In the selected Plan Template's detail page (lower pane), click the **Properties** button.
- 5. To edit properties, click the **Edit** button. (If properties are read-only as discussed above, this button is disabled.)

There is one property in Plan Templates that is only present at the template level, and not in Plans created from the template: Color. This field accepts standard hex color notation used in web development (e.g. #nnnnn, where #000000 is black, #ffffff is white, etc.) Standard web color names (Red, Blue, Cyan, etc.) can also be used. The color specified in this field of the template is applied to the icon that signifies "Plan". Once a Plan is created from a Plan Template, the color field cannot be changed in the properties of the created Plan.(Color is still editable in the Plan Template).

An important property to note is the option *Report shared from Template*. When checked, the Plan's report Page is shared in Plans instantiated from the Plan Template. Then, if a user invokes the action to customize the Plan report, they are offered the option to customize either the report Page of the instantiated Plan, or the report Page of the Plan Template. The user can perform the selected customization, assuming that s/he has the necessary permissions.

#### **Plan Template Operations**

The (Operations) menu in each Plan Template provides several functions you may need:

- Customize Plan Report: This action opens the underlying report Page of the Plan Template, enabling you to customize the layout, modify parameters of existing Widgets, or add additional Widgets to provide the desired functionality and information for the project team.
- **Delete**: Invoke this action to delete a Plan Template. Only the current Plan Template is deleted. Plans that users have created from it are not affected and remain in the project.

#### **Note**

The menu also has items for exporting the Plan Template page to PDF, or printing it.

# **Configure Plan permissions**

Administrators can review and optionally modify the default user permissions for Plans. Plan permissions can be configured in both Global/repository scope and project scope.

#### **Access Plan permissions**

- 1. Open the scope you want to configure and enter **Administration**.
- 2. In Navigation, expand ... User Management and select Permissions Management.
- 3. On the Permissions Management page, expand Plans in either tab to access Plan permissions.

For more information on configuring user permissions, see Configure User Permissions.

#### Use Plan enumeration for Work Item fields

Polarion includes a special enumeration **Plan** that can be used for Enum: type custom fields in Work Items or Documents. Such fields show a list of Plans, optionally enabling users to select one or more Plans in the list. The list of Plans can be filtered by specifying a query in the Query parameter of the enumeration. (If that parameter is left empty, the custom field list will list all Plans in the project.)

To do this, create a custom field as usual, selecting **Enum**: in the **Type** column. The enumeration selector appears, in which you should select Plan. When this enumeration is selected, a field for the query parameter appears. Enter a query using Lucene syntax to filter the list of Plans rendered in the list for end users. For example, to have the field show only open plans the guery might be: status:open.

See also: Configure Custom Fields, Configure Document Custom Fields.

# **Configure the Portal**

# **Configure Shortcuts and Favorites**

#### **Configure Shortcuts and Favorites overview**

Scope(s): Repository, Project

You can configure the global and project-specific default Shortcuts and Favorites to be shown to users in the Navigation panel. Shortcuts appear in the Global Shortcuts and Project Shortcuts topics. There is also a **User Shortcuts** topic, but administrators cannot configure default shortcuts for it. (Note that it may be necessary to use the **Expand** control to access the Shortcuts topics). Favorites (a.k.a. Favorite Shortcuts) appear in the top section of the panel.

(See: Navigation and Content for more information.)

The default Favorites you configure will be shown in the Favorites section of Navigation to new users, and to users who reset Favorites. Keep in mind that users can customize their Favorites and may choose not to show the configured defaults.

Global and project shortcuts can be modified by users who have permissions to do so via the **Manage** Shortcuts topic in Navigation (Tool view). They can add new shortcuts, and remove or change the

order of existing shortcuts. As an administrator, you can review Global and project shortcuts in the **Administration** topics and remove any user-defined shortcuts that are outdated or inappropriate.

### **Global and Project Shortcuts**

You can configure the content of the **Shortcuts** topics in **Navigation**. **Shortcuts** provide users with quick access to actions (such as creating a new Work Item), wiki pages, or subsets of Work Items (which also includes the Work Item view).

Configuration data is stored in the shortcuts.xml file for the relevant scope. You can edit this file directly in Administration, or download it, edit offline, and upload back to the server. When you configure shortcuts for a repository, they appear as nodes under the **Global Shortcuts** topic in **Navigation**. When you configure shortcuts for a project, they appear under Project Shortcuts when users open the respective project.

#### **Access the Shortcuts configuration**

- 1. Log in with administrator permissions for the scope you want to configure
- 2. If you want to configure shortcuts globally, log in to the repository you want to configure (if your system is configured for multiple repositories), enter \*Administration, and open the Repository. (See: Access Projects.)
- 3. If you want to configure shortcuts for a project, enter Administration and open the desired project. (See Access Projects.)
- 4. In Navigation, expand **Portal** and select Shortcuts.

You can either edit the configuration online in the text editor on the configuration page, or you can download the XML configuration file, edit it with a preferred tool, and upload it back to the portal.

The best way to familiarize yourself with the XML is to look at the default global configuration, and the project configuration for the example project(s) included with your Polarion product. For example, you can readily see that a shortcut to a set of Work Items involved embedding the syntax of some query in the <code>query</code> parameter of a <code>shortcut></code> element.

For example, this shortcut would show unresolved Work Items with the Suspect flag set:

```
<shortcut name="Suspect Work Items" icon="/polarion/ria/images/details/table.gif"
  pagePath="workitems" query="suspect:true" resolution="unresolved"/>
```

#### **Favorite Shortcuts**

You can configure the default **Favorite shortcuts** that appear to new portal users, and existing users who reset their Favorites. You can configure default **Favorites** for repositories, **Project Groups**, and **Projects**.

When working in repository Administration topics are available:

- ₱Portal ➡ Default Favorite Shortcuts for Projects

When working in project Administration, the project-scope configuration is in Portal - Default Favorite Shortcuts. Settings in this scope override those of the global configuration in Default Favorite Shortcuts for Projects.

This configuration requires editing XML. It is normally performed directly in the Administration page using the editor provided.

# **Configure default Favorites**

- 1. Open the desired configuration topic in Administration (one of those described above).
- 2. Edit the XML to create the desired configuration. (Check the configuration in the sample projects as examples.)

# Configure languages

### Configure languages overview

Scope(s): Repository, Project

You can optionally configure Polarion to support multiple human language translations of Work Items. When correctly configured, a Title and Description field for one or more additional languages appear in the Work Item form in tracker views (Table, for example), and users can switch languages in Documents to view translated Work Items.

Multi-language support can be configured globally for the repository and per project. You can optionally configure different languages for different Work Item types, and multiple languages for a single type.

#### There are two configuration steps:

- Define custom fields to store translated texts
- Add configuration(s) for supported translation language(s) in Administration.

#### **Define custom fields**

The first step in multi-language configuration is to define custom fields to store translated text of Work Item titles and descriptions. The procedure is the same as described in Configure Custom Fields.

You need to define two custom fields for each language you want to support: one for the translated Work Item title, and one for the description. The field for the title should be of type string. The field for description should be of type Rich Text (multi-line). You need to define this field pair for each Work Item type that needs to support multiple languages, and in each Work Item custom field configuration you need this field pair for each language that must be supported.

For example, suppose that for Requirement type items you need to support translations for German and Japanese. In the **Custom Fields** configuration for Requirement, you would need to specify:

Two fields for German, with IDs like requirement title de (string field for the German title), and requirement descr de (multi-line rich text field for the German description). The Name property can be something like Title (DE) and Description (DE), or the value could be in the target language (e.g. Titel and Beschreibung).

• Two fields for Japanese: with IDs like requirement\_title\_jp and requirement\_descr\_jp, and Names like Title (JP) and Description (JP) (or the words in Japanese).

# Configure multiple languages

You perform this configuration in the \*Administration \*Portal Languages in the scope you want to configure. There are two configurations you can perform on this page:

- 1. Default Language label.
- 2. One or more translation languages for Work Items.

#### Default language label

The first item in the table on the Languages page is the configured label for the Document Editor control that enables users to switch from a translation to the portal default language. In English versions of the Polarion portal, the value is the English string "Default Language".

To change the default, click the **Create** button on the table row. Then on the Default Language page, change the value of the label and click the **Save** button.

### Add a language

Before adding a language to the configuration, be sure you have performed the step of creating custom fields for the translated Title and Description for the target language.

- 1. On the Languages page, click Add New Language.
- 2. In the **Add New Language** dialog, enter a name for the language. This name will appear to end users in the Languages menu of the Document Editor, which enables them to switch to a different translation of the document. For example, if the language you are configuring is German, you might enter German or Deutsch.
- 3. Click **Next** to proceed. The language configuration page appears.
- 4. In the language page, select a Work Item type in the Work Item Type list.
  - If you choose **All Types**, the configuration will apply to *new* Work Items of all types, but not to existing Work Items of any type. If you want the existing items to have translation fields for this language, you will need to configure the specific type(s).
- 5. In the **Title** column, select the name of the custom field configured for this language for the translated Work Item title.
- 6. In the **Description** column, select the name of the custom field configured for this language for the translated Work Item description.
- 7. If you want to configure this language for additional Work Item types, click the ♣ icon and repeat the previous 3 steps.
- 8. Click **Save** when finished.

You should repeat this configuration for each language for which you want to support translations.

On the Languages page you can edit any existing language configuration, or delete a language configuration. Note that deleting a language configuration will simply remove the language item from the Switch Language menu. It will not remove any translation data that has been added to Work Items. If you wanted to remove the data, you would need to remove the custom fields from the Work Items configuration and re-index the system.

See also: View Work Item Translations.

# **Configure interface views**

# Configure interface views overview

# ALM

See Views-related Features and Licenses for details.

You can configure different variants of the Polarion user interface for end users who work with the Polarion platform in different contexts. For example, one user may function as a "Manager" at some times, and as a "Developer" at other times. You can configure different Interface Views ("Views") which provide varying levels of information presentation, displaying just the navigation topics, Work Item field content and Work Item form layout needed for the selected work context.

A default set of Interface Views is defined in the global configuration, and in Project Templates. Project administrators can override the global or template configuration and define a custom set of Interface Views for any project. The default Interface Views in the default Project Templates are pre-configured to provide just the information typically needed when working in the role described by the name.

Generally, Interface Views should have descriptive names that describe the role of people for whom they are intended. For example:

- Default (which shows all available information)
- Requirement Engineer
- Tester
- Developer
- Manager

Bear in mind that Polarion's default Project Templates can be customized, and/or custom variants can be created. What Interface Views exist, what information any View provides, and how it is laid out, depends on how the template developer configured Interface Views in the project template.

#### Warning

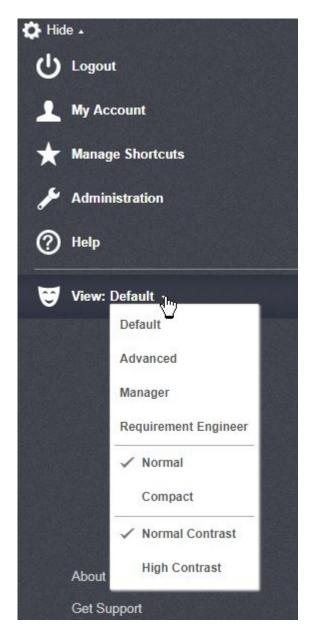
Interface Views are not a security feature. For example, they will not prevent some users from accessing some information. User roles and permissions should be used to control users' access to portal content.

## **Configuration scope**

You can define Interface Views globally for each Repository, and for each project. The global configuration is inherited by each project, unless overridden by a project template, and the settings for each of the globally defined Views can be modified in projects in addition to those provided by the Project Template.

You can customize Interface Views to change the display of navigation topics, Work Item fields, and the Work Item form layout. You can also create new Interface Views that specify the Navigation topics, Work Item fields and Work Item form layout to display when a user selects the View in Navigation. All configured Interface Views appear in the **Views** list in the Tools mode of the Navigation panel (below the Help link).

To see the interface Views currently configured for a project or change to a different View, click the icon in **Navigation**, then with the panel in Tool mode, click the list next to the mask icon and select the desired View name.



(The current View in highlighted in the list.)

You must have Administrator permissions for the scope you want to configure (Repository or Project).

#### **Process overview**

The following Administration topics are needed to configure Interface Views:

- **₱Portal → Views**: This is where you specify what interface Views exist in the configuration for the scope you are configuring.
- **₱Portal → Topics**: This is where you specify which Navigation topics appear in each Interface View.

• Work Items Form Configuration: This is where you specify what information is shown on the Work Item form (Form Filters), and the form layout to be used when a user is working in the Interface Views.

The main configuration process steps are:

- 1. Create a new interface View in the desired scope, or access an existing interface View.
- 2. Configure field filtering for the Work Items form to control what fields appear in the form when a user invokes the Interface View.
- 3. Configure the layout of Work Items form fields to control how the fields shown via the filtering configuration are laid out when a user invokes the Interface View.
- 4. Configure the list of Navigation topics to display when a user invokes the Interface View.

Following sections in this topic provide more detail about each of these main steps.

It is important to note that when a new Work Item form filter or Work Item form layout configuration is created, its content is pre-filled with its parent configuration. The inheritance scheme is as follows:

- From default to specific Interface View
- From unspecific to specific Work Item type
- From global to project scope

#### Views-related features and licenses

When at least one Polarion ALM license is present on the server, the following View-related features are available:

- Multiple Views are configurable, and accessible to users in the web interface.
- Form Configuration (layouts and filters) per View.
- Form Menus (linked and linking) per View.
- Read-only Fields per View.
- Topics configuration per View.

If no Polarion ALM license is present:

- Form Configuration (layouts + filters), Read-only Fields, and Topics are configurable only for the
   \_default View.
- Interface Views are not visible or accessible to users.
- Form Menus (linked and linking) are not exposed in the user interface and the configuration is not applied.

#### Define a new interface view

- 1. Open the project or repository in which you want to create a new Interface View, and enter Administration.
- 2. In the **Navigation** pane, expand **Portal** and select Views. The Views page appears and displays a table listing the Interface Views currently existing the scope in which you are working.
- 3. In the table, click the **Create New View** button. An edit form appears in the bottom section of the page.
- 4. Enter a unique ID for the new View. Use letters, numbers, underscore, but no spaces. For example: techWriter.
  - The system will derive a value for Name from your input. This value will appear in the Interfaces list in the Tools view of Navigation when a user opens the project or repository. For example, if you enter techwriter, then Name will appear as Tech Writer.
- 5. If you want the Interface to be available only to users who have a specific role assigned them in User Management, specify the role(s) in the **Limited to Roles** section.
- 6. Click the Create button to create the new Interface View. (To show the new View in the table click Refresh.

A newly created a new Interface View does not differ from the default in terms of Navigation topics, Work Item fields shown, and Work Item form layout. It is necessary to address these configurations for the new View. They are covered later in this topic. See Configure Work Item Form Filters, Configure Work Item Form Layout, Define Read-only Fields, and Control Navigation Topics.

#### Access an existing view definition

You may need to access an existing Interface View configuration to modify roles, or delete the definition.

- 1. Open the project or repository where the Interface View you want to access is configured and enter Administration.
- 2. In the **Navigation** pane, expand **Portal** and select **Views**. The administration page displays a table listing the Interface Views currently existing the scope in which you are working.
- 3. In the table, select one of the existing Views. Detail information appears in the bottom section of the page.

#### **Delete a View definition**

If you want to delete an Interface View definition, select it according to the steps in the previous section. When the desired View is selected, click the **Delete** button in the toolbar of the detail section of the administration page.

#### **Configure Work Item form filters**

You can configure filtering of the Work Item form for any Interface View. Filtering can hide some fields, or set some fields as read-only for different workflow actions and statuses. Filtering is activated by clicking the button on the Work Items form's toolbar. Remember that this is not a mechanism for restricting access to fields. Users can restore visibility and editing of fields hidden by filtering at any time.

- 1. Open the project or repository that has the Interface View for which you want to configure form filtering, and enter Administration.
- 2. In the **Navigation** pane, select **Work Items Form Configuration**. The Form Configuration page appears.

The Form Configuration page contains two main sections. The first section is **Form Filters**. You can either create a new form filter, edit an existing form filter, or remove an existing form filter.

#### Create a New Filter

- 1. In the Form Filters section of the Form Configuration page, click the Create New Form Filter button. The Create New Form Filter dialog appears.
- 2. In the dialog's **View** list, select the View you want to configure. If you want the form filtered only for a specific type of Work Item, select the type from the **Work Item Type** list. After selecting the desired list value(s), click the **Next** button. The **Form Filter Configuration** page loads in your browser.
  - If you are working in a project, values in the **Actions** and **Statuses** sections are inherited from the global configuration. You can make changes as desired for the project.
- 3. If necessary, scroll the page to expose the embedded Quick Help info for the page. Use this information to specify the Actions and Statuses comprising the filter.
- 4. When you have finished with the filter parameters, click the **Save** button in the bar below the page heading.
- 5. To return to the **Form Configuration** page, use the text link **Return to Form Configuration** at the top of the page.

#### Modify an Existing Filter

- 1. Navigate to the **Work Items Form Configuration** topic as described earlier in this section. Be sure you are in the correct project or repository.
- 2. In the **Form Filters** section, find the filter you want to modify in the table.
- 3. Click the **Edit** button. The Form Filters page appears.
- 4. Change the settings on the Form Filter Configuration page, referring to the embedded Quick Help information (scroll down if it is out of view).
- 5. Click **Save** when finished, or the **Cancel** button to undo unsaved changes.

### Remove an Existing Filter

1. Navigate to the **Work Items Form Configuration** topic as described earlier in this section. Be sure you are in the correct project or repository.

- 2. In the **Form Filters** section, find the filter you want to remove in the table.
- Click the **Delete** button and respond affirmatively to the confirmation prompt.

#### **Note**

Global configuration for the -- Default -- View plus the -- unspecific -- type cannot be deleted because other configurations are derived from it.

# **Configure Work Item form layout**

# ALM REQ QA

#### **Note**

If Polarion Requirements or Polarion QA is running on a server where no Polarion **ALM** license is present, then multiple Interface Views are not available, and only the form layout of the Default View is configurable.

For any Interface View, you can configure the layout of the fields on the Work Item form. This form enables viewing and editing of Work Item details in the lower half of the Table presentation of Work Items (both Document-based and Tracker-based) when you select a Work Item in the table. This configuration requires editing an XML configuration file. Administration provides access to this file and a basic online XML editor enabling you to edit the file online without leaving your browser. Alternatively, you can copy the XML from the online editor, paste it into your favorite XML editor to work on it, and finally past the modified XML back to the online editor.

This section explains how to access the configuration data and the online editor. You will find comments embedded in the XML code, plus embedded Quick Help on the editor page that provides information about the elements and attributes used in the configuration.

#### Access the form layout configuration:

- 1. Open the project or repository for which you want to configure Work Items form layout, and enter **Administration**.
- 2. In the Navigation pane, select \*Work Items Form Configuration. The Form Configuration page loads.
- 3. Find the Form Layouts section of the page (scroll down if necessary). This section enables you to create a new form layout, and presents a table of existing Views having a customized Work Item form layout. You can edit or remove any existing form layout.

#### **Create a New Form Layout**

The Interface View for which you want to create a custom form layout must already exist. That is, you cannot create new View during the form layout configuration process.

1. Navigate to the Form Configuration page as described in the previous set of steps.

- 2. In the Form Configurations section, click the Create New Form Layout button. The Create New Form Layout dialog appears.
- 3. In the View list, select the Interface View you want to configure. If you want the form layout to apply only for a specific type of Work Items, select the type from the Work Item Type list. After selecting the desired list value(s), click the Next button. The Form Layout Configuration page loads in your browser.
- 4. In the embedded XML editor, edit as needed to create the layout you want for the selected View and Work Item type (if you selected one). Refer to the embedded Quick Help text on the page (scroll down if necessary to see it).
- 5. Save your configuration when you finish editing. As soon as you change anything in the XML editor, the **Save** and **Cancel** buttons become enabled. These enable you to save changes or abandon any unsaved changes.

#### **Modify an Existing Form Layout**

- 1. Open the project or repository which contains the Interface View you want to modify.
- 2. In the Navigation pane, select \*\*Work Items Form Configuration.
- Locate the table of existing layouts in the Form Layouts section of the Form Configuration page (scroll down to see it if necessary). Click the Edit button on the row of the layout you wish to modify. The Form Layout Configuration page loads in your browser, and the XML code for the selected layout loads in the embedded XML editor.
- 4. Modify the XML code as desired, referring to the page's Quick Help text if necessary. Click the **Save** button when finished.

#### Remove an Existing Layout

- 1. Navigate to the **Work Items** Form Configuration topic as described earlier in this section. Be sure you are in the correct project or repository.
- 2. In the **Form Layouts** section, find the layout you want to remove in the table.
- 3. Click the **Delete** button on the layout's row in the table, and respond affirmatively to the confirmation prompt.

#### **Note**

Global configuration for the -- Default -- Interface View plus the -- unspecific -- type cannot be deleted because other configurations are derived from it.

## **Define Read-only fields**

Unless configured as read-only, many fields can be edited "in place" without placing the entire Work Item form into Edit mode. In each Interface View configuration, you can define which fields should be read-only when the entire Work Item form is *not* in Edit mode. The configuration can include custom fields. You can perform this configuration for any Interface View in which you want specific fields to

always be read-only, either always, or when the Work Item enters a specific workflow status. For example, when a Requirement has the Accepted status, you can make the Description field read-only for anyone using, for example, a custom "Customer" View.

The same read-only restrictions provided by the configuration apply in the Multi Edit presentation of Work Items, and to bulk-edited Work Items. In the Matrix presentation, only restrictions for linked Work Items are applied, meaning that links cannot be added. In the Time Sheet presentation, only restrictions for work records, time spent and remaining estimate are applied, meaning that work records cannot be added or edited.

This configuration is available in both global (repository) and project scopes. Remember the project settings override the global settings, which are the default for new projects.

- 1. Log in to the portal with administrator permissions for the scope you want to configure.
- 2. Open the repository or the project you want to configure, and then enter Administration.
- 3. In Navigation, expand Work Items and select Read Only Fields. The Read-only Fields page loads in your browser.
- 4. To create a new configuration, click the Create New Read-only Fields button. The Create New Read-only Fields dialog appears.
- 5. In the View list, select the Interface View in which you want a field to be read-only. If the configuration should apply to a specific Work Item type, select the type in the Work Item Type list. Then click Next. The Read-only Fields Configuration page appears.
- 6. In the **Read-only Fields** table, enter a status ID and one or more field IDs (comma-delimited) in the respective columns. Click the • icon to add more rows to the table and configure additional statuses and fields. Refer to the Quick Help text on this page for reference information and links to values.

Note that there are 2 constants defined for the **Status ID** field which can be specified instead of an actual status ID value:

- default: the fields specified in Fields will be read-only when no status-specific configuration exists
- create: the fields specified in Fields will be read-only when a new Work Item is in process of being created but is not yet saved.

#### **Note**

If the table already contains some configured statuses and fields, it is possible to remove these by clicking the - in the relevant row if these configuration are no longer desired.

7. When you have completed the configuration, save your changes using the **Save** button located on the page.

#### Tip

You can use a special command <code>@all</code> to specify that all fields (including custom fields) should be read-only. You can specify exceptions, i.e. writable fields, by specifying the desired fields prefixed with '-'. More information, command syntax, and an example are provided in the embedded Quick Help on the configuration page in <code>Administration</code>.

# **Configure Navigation topics**

For each Interface View configured, you can control which topics appear in Navigation when a user selects the View. For example, you might create a tech writer view, and decide that users invoking it only need to see the **Work Items** and **Documents and Pages** topics in addition to the **Home** and project and user Shortcuts topics.

- 1. Open the repository or project you want to configure and enter \*Administration.
- 2. In **Navigation**, expand **Portal** and select **Topics**. The Topics page loads.
- 3. If you want to create a new configuration, click the **Create New Topics Configuration** button. In the dialog, select the Interface View you want to configure.
  - If you want to modify an existing configuration, click the **Edit** button on the row of the configuration you want to modify.
  - In both the above cases, an online XML editor appears with the XML code of the selected configuration.
- 4. In the online XML editor, comment out or remove elements in the correspond to the Navigation topics you do not want to show in the Interface, leaving only the elements for the desired topics. For example, your configuration for the previously mentioned Tech Writer View might be:

5. Save the changes to the file.

# **Configure Reports**

# Source code Reports

There are two basic types of reports: Source code reports which appear in the Reports topic of the portal, and Audits/Metrics reports which appear in various dashboards. All reports are customizable in the Administration interface by means of XML configuration files. This section explains the basics of how to configure the different types of reports.

You can configure source code reports for each project. If you are not familiar with the basics of the different scopes, you may want to review The Administration Interface.

You can configure which reports appear in the **Reports** topic, specifying the calculation for each report. You can also customize the calculation of each report. The default reports are:

- Mayen Site
- Unit Test Coverage
- Unit Tests
- Javadoc
- XRef

Two additional reports are available, but are commented out in the default reports configuration file:

- **PMD**
- Checkstyle

There are 2 files involved in this configuration:

- reports-config.xml: defines the list of source code reports shown in the Reports topic of the portal.
- descriptors.xml: configures the calculations used by the reports defined in reports-config.xml.

## Access the configuration files

To access the configuration files for source code reports:

- 1. Log in with administrative rights for the project you want to configure.
- 2. Enter the **Administration** interface.
- 3. Navigate to and select the project in the Open Project or Project Group dialog. (See: Access Projects).
- 4. In the **Navigation** pane, expand the **Reports** topic.
- 5. To access the reports-config.xml file, click on the Source Code Reports topic.

To access the descriptors.xml file, click on the Report Descriptors topic.

- 6. Use the appropriate link in the **Configuration** section of the respective configuration page to download the configuration file to your local system, or edit online with the text editor provided.
- 7. If you edit the file offline, use the controls in the **Upload New Project Configuration** section to upload the modified file back to the Polarion repository.

You will find comments in the configuration file that explain the various elements and attributes of the configuration.

#### **Note**

Source code reports can be computed only if the build artifact (containing Java sources) is a maven2 build artifact, and the build process is running without errors.

#### Work Item metrics

#### Work Item metrics overview

Work Item metrics are Work Item-related values (facts) that are calculated by Polarion on projects and project groups. There are two related calculations: **trackeranalysis**, which calculates metrics on projects, and **trackeranalysis-projectgroup**, which calculates metrics on project groups.

Value of any Work Item metric is one of the following types:

- integer (int): non-negative (0, 1, 2, ...)
- distribution: i.e., a set of integer sub-values that is rendered as a bar with colored stripes, where
  width of each stripe is proportional to the integer sub-value. Each sub-value has an ID (called
  subkey) so that it can be referenced e.g. on a dashboard, and a label that is displayed as tool-tip
  when hovering above the respective stripe in the bar.
- **percentage**: e.g 70%, rendered as a colored bar in a way similar to distribution.
- items-traceability: defines a notion of traceability of Work Items; the value is a distribution of traceable and not-traceable items that satisfy 'itemsQuery', where traceable items are those, from which one can follow the specified links ('link' elements) in specified directions and find at least one item that satisfies the 'coverageQuery'; items that satisfy both 'itemsQuery' and 'coverageQuery' are automatically considered traceable without checking links.
- commits-traceability: defines a notion of traceability of repository commits; the value is a
  distribution of traceable and not-traceable commits, where traceable commits are those, that are
  linked to at least one item from which one can follow the specified links ('link' elements) in
  specified directions and find at least one item that satisfies the 'coverageQuery'
- field-sum and field-average: calculates the sum or average of the values stored in a custom field.

You can configure Work Item metrics for projects. If you are not familiar with the basics of the different scopes, you may want to review: Administration Interface.

This configuration contains elements that define values that may be shown on a dashboard, or used in the definition of a trend chart.

#### Access the configuration files

Work Item metrics are configured in the workitem-metrics.xml file. To access this file:

- 1. Log in with administrative rights for the project you want to configure.
- Enter the **Administration** interface.
- Open the project in the Open Project or Project Group dialog. (See: Access Projects).
- 4. In the **Navigation** pane, expand the **Reports** topic.
- 5. Click on the **Work Item Metrics** topic.
- 6. Use the appropriate link in the **Configuration** section to download the configuration file to your local system, or edit online with the text editor provided.
- 7. If you edit the file offline, use the controls in the **Upload New Project Configuration** section to upload the modified file back to the Polarion repository.

The configuration file contains comments that explain the elements, their use, and options for customization.

### Work Item metrics configuration syntax

The workitem-metrics.xml file defines which Work Item metrics will be calculated on projects during the trackeranalysis calculation. The basic XML syntax and elements are shown in the following code example:

```
<workitem-metrics>
    <metrics>
       <entry>
          <key>KEY</key>
          <label>LABEL</label>
          <description>DESCRIPTION</description>
          <value type="TYPE" ...>
          </value>
       </entry>
    </metrics>
</workitem-metrics>
```

#### Where...

- KEY is a unique identifier for the metric, to be used when referencing the value (e.g. in a dashboard). Required.
- LABEL is a text label for the metric. Optional (only for in-place defined metrics).
- DESCRIPTION is a longer text description for the metric. Optional (only for in-place defined metrics).

• TYPE is the type of the metric (integer, distribution, percentage). Required (only for in place defined metrics).

#### Internally defined Work Item metrics

By default the configuration contains a number of internally defined metrics. For example:

You can optionally remove any of these elements. If removed, the internally defined metric will not be calculated during the **trackeranalysis** calculation. For the complete list of the internally defined Work Item metrics, see: Internally Defined Work Item Metrics.

#### In-place defined metrics

The way the value is calculated is described by the value <code>subelement</code>. There are four types of in place defined metrics:

- int
- distribution
- items-traceability
- commits-traceability

#### int

Integer value defined as the size of the search result of a Work Item query. Example:

In this example the value of the metric OPEN will be the number of Work Items with no resolution value set.

#### distribution

There are 2 ways a distribution value can be defined: using an enumeration, and using queries.

#### Definition using an enumeration

#### Syntax:

```
<value type="distribution" enumeration="..." query="..." reverseOrder="..."/>
```

Where...

- enumeration is the specification of an enumeration (required), which can be in one of the following formats:
  - o enum: ENUM\_ID all options from an enumeration with the specified ID, including type-specific ones.
  - o <code>wi\_Type:enum:ENUM\_ID</code> those options from an enumeration with the specified ID, that are valid for the specified Work Item type.
  - o FIELD: (deprecated) ID of a single-valued Work Item field of an enumeration type. Can be custom, but must not be Work Item type-specific.
- query is a Work Item query in which substrings "\$option\$" will be replaced by the ID of an option from the enumeration.
- reverseOrder if set to true, the values in the distribution will be sorted in reverse order with respect to the sort order of the enumeration options. (Optional)
- unresolvedonly (deprecated) if set to true, only unresolved items will be counted. (Optional)
- resolvedonly (deprecated) if set to true, only resolved items will be counted. (Optional)

This method uses options from an enumeration type, severity, priority, resolution, or any custom enumeration). Each integer value that is part of the distribution corresponds to one option from the corresponding enumeration, and is equal to the number of Work Items matching a query created using the option ID in a way that depends on attributes of the value element: if there is enumeration id specified, query from the query attribute will be used with <code>soptions</code> replaced by the option ID. If field ID is specified, a query matching items having the option set in that field will be used (the query will possibly be extended based on the remaining attributes).

The order of values in the distribution is defined by the sort order of the options in the enumeration. ID and label of sub-values is ID and label of the corresponding enumeration option.

## **Example:**

#### **Definition using queries**

#### Syntax:

#### Where...

- query the content of the element is a Work Item query. (Required)
- id is the ID of the sub-value of the distribution. (Required)
- label is the label of the sub-value of the distribution. (Required)

This method uses sub-elements <code>query</code> of the value element. Each such sub-element corresponds to one value in the distribution, and the value is defined as size of its search result.

#### Example:

#### items-traceability

#### Syntax:

#### Where...

- itemsQuery is a query that defines the set of Work Items, starting from the path of which, and consisting of Work Item links, will be searched for.
- links defines which links can be part of the path.
- link/id is the ID of the link role (or a regular expression matching link role IDs) that can be part of the path.
- link/backLink if set to true, only incoming link of the matching ID can be in the path (otherwise only outgoing).
- coverageQuery is a query that defines a set of items whose path is searched for. (You can use the special value has-linked-revisions that matches items having at least one linked revision.)

Each metric of this type defines a notion of traceability of Work Items. The value is a distribution with two integer sub-values: count of traceable and not-traceable items among those that satisfy <code>itemsQuery</code>. Traceable items are those, from which one can find a path consisting of links matching some conditions (defined by <code>link</code> elements) that leads to at least one item that satisfies the

coverageQuery. Items that satisfy both itemsQuery and coverageQuery are automatically considered traceable without checking links. (Empty path is always acceptable).

### Example:

```
<entry>
   <key>TRACEABILITY-REQUIREMENTS-COMMITS</key>
   <label>Traceability Work Items to Commits
   <description>
       Rate of Work Items that are linked to commits.
   </description>
   <value type="items-traceability">
       <itemsQuery>HAS VALUE:resolution</itemsQuery>
       ks>
           <link id=".*" backLink="true"/>
           <link id=".*" backLink="false"/>
        </links>
        <coverageQuery>has-linked-revisions</coverageQuery>
   </value>
</entry>
```

#### commits-traceability

This is very similar to items-traceability. The only difference is that there is no element itemsQuery; - the path is searched from all Work Items that are linked to at least one revision (commit) in the project. The sub-values in the distribution correspond to the count of project revisions that are linked to traceable Work Items, and to the count of project revisions that are not linked to a traceable Work Item.

### Example:

```
<entry>
    <key>TRACEABILITY-COMMITS-REQUIREMENTS</key>
    <label>Traceability commits to Work Items</label>
    <description>
       Rate of Commits that are linked to Work Items.
   </description>
    <value type="commits-traceability">
        ks>
            <link id=".*" backLink="false"/>
            <link id=".*" backLink="true"/>
        </links>
        <coverageQuery></coverageQuery>
    </value>
</entry>
```

## field-sum and field-average

Calculates the sum or average of values stored in a custom field. (Configured as the 'field' attribute of Work Items returned by a query that is specified as the 'query' attribute.) Calculations are only possible for number (integer, float, currency) type fields.

## **Example:**

```
<value type="field-average" query="type:testcase" field="passRatio"/>
```

## **Work Item Quality report**

You can configure the Work Item Quality report for each project. If you are not familiar with the basics of the different scopes, you may want to review Administration Basics: The Administration Interface.

This configuration enables you to define which of the available metrics should appear in the dashboard.

## Access the configuration files

Work Item metrics are configured in the workitem-quality-report.xml file. To access this file:

- 1. Log in with administrative rights for the project you want to configure.
- 2. Enter the **Administration** interface.
- 3. Navigate to and select the project in the **Open Project or Project Group dialog** (see: Access Projects).
- 4. In the **Navigation** pane, expand the **Reports** topic.
- 5. Click on the Work Item Quality Report topic.
- 6. Use the appropriate link in the **Configuration** section to download the configuration file to your local system, or edit online with the text editor provided.
- 7. If you edit the file offline, use the controls in the **Upload New Project Configuration** section to upload the modified file back to the Polarion repository.

## **Process Quality Report**

You can configure the Process Quality report for each project. If you are not familiar with the basics of the different scopes, you may want to reviewAdministration Interface .

## Access the configuration files

Process audit rules for the process quality report are configured in the <code>rule-settings.xml</code> file. The file is commented to help you understand the elements and options. To access this file:

- 1. Log in with administrative rights for the project you want to configure.
- 2. Enter the **Administration** interface.
- 3. Navigate to and select the project in the **Open Project or Project Group** dialog (see: Access Projects).
- 4. In the **Navigation** pane, expand the **Reports** topic.
- 5. Click on the Process Quality Report topic.
- 6. Use the appropriate link in the **Configuration** section to download the configuration file to your local system, or edit online with the text editor provided.

7. If you edit the file offline, use the controls in the Upload New Project Configuration section to upload the modified file back to the Polarion repository.

## **Configuring Documents and Pages**

## **Configure Documents and Pages overview**

The **Documents & Pages** topic of **Administration** presents the following configuration options:

- **PDF** Export Configuration
- **Document Custom Fields**
- **Document Types**
- **Document Workflow**
- Attachment Size Limit

A related topic, covered in the chapter on Advanced Administration, is Dashboard Customization.

## Configure Document types and workflow

Sometimes it can be useful classify Polarion LiveDoc documents ("Documents") by type. For example, you might want to have different Document types for different types of specifications. Also, there can be a need to have workflow control over entire Documents in order to establish traceability for some process related to Documents. For example, you might want to have Document statuses such as "Draft", "Under Review", "Reviewed", "Approved", etc. and be able to transition entire Documents through these statuses.

When Document types are configured, users can assign a type to new or existing Documents in the Document Properties sidebar of the Document Editor, and view the assigned type there.

#### Note

Most of the default project templates have preconfigured Document types and Document workflows. (There are more project templates available on the Polarion Extensions Portal that may also have these configured). Explore whether a template can meet your needs before deciding to customize.

The default Project Templates and demo projects are pre-configured not only with Document types and workflow, but also underlying permissions that go together to help control the workflow. For example, the content of Documents of the Requirements Specification type (predefined in several templates) cannot be modified if the workflow status is inReview or Approved.

You can configure Document types and Document workflow globally and per project. Remember that the Global configuration provides the default that is used for all projects not having a project-scope configuration. Document types and Document workflow must be explicitly enabled by an administrator before the features are available to end users. The following sections explain the configuration procedures.

## **Security: Best Practices for Pages**

#### Warning

Poorly written code on a Wiki or LiveDoc <sup>™</sup> (Rich Text) page can leave your system open to XSS (Cross-site scripting) attacks using malicious Work Item and API calls. (Anything, Java code included, can be added to a Work Item in the document and the API will return the actual value of the target fields.)

## **Prevent Cross-site scripting (XSS) Vulnerabilities**

- Restrict the number of users that can modify Wiki and LiveDoc ™ (Rich Text) pages.
- Administrators should notify any users that modify Wiki and LiveDoc ™ (Rich Text) pages of the information below:
- Use the Rendering API (If \$workItem is an instance of com.polarion.alm.shared.api.model.wi.WorkItem):

#### \$workItem.title.render

- If it is not possible to use the Rendering API the use the following;
  - (If \$workItem is an instance of com.polarion.alm.tracker.model.IWorkItem):
  - On Wiki Pages: **\$esc.escape(\$esc.escapeForHtmlTag(\$workItem.title)**
  - On LiveDoc ™ (Rich Text) Pages: \$esc.html(\$workItem.title)

## **Set up Document Types and Workflow**

If you want to use Document types and Document workflow, you need to first set up the feature in Administration.

- 1. Open the scope in which you want to enable the features (Global or some project) and enter Administration.
- 2. In Navigation, expand **Documents and Pages** and select **Document Types**.
- 3. On the toolbar of the Document Types page, click the **Set Up Types** button. (If the button is not present, it means that Document types and workflow are already enabled for the scope.)
- 4. In the **Set Up Document Types** and Workflow dialog, click the **Start** button.

The setup operation will automatically copy default types and workflow to the current context. After the process is finished, you can customize types and workflow as described in the following sections.

## **Configure Document Types**

This configuration provides users the possibility to classify Documents according to type. It should be completed first before configuring Document workflow. For the Global scope, review the default types provided by Polarion and modify the configuration if needed to provide a common denominator for all projects. Project teams will need to decide whether the Global types meet their needs and if not, work with the administrator to modify the project configuration accordingly.

To access the configuration topic:

- 1. Open the scope you want to configure (Global or project).
- 2. In Navigation, expand Documents and Pages and select Document Types.

If your installation does not have Document types and Document workflow set up yet, you will see a button Set Up Types when you access the Document Types page. Click this button and follow the dialog instructions.

On the **Document Types** page you can...

- Add new Document type definitions.
- Modify existing Document type definitions.
- Delete existing Document type definitions.

When you delete a Document type, any Documents that have been assigned that type will display the type ID in the **Type** field in Document Properties. (Normally, the type name and icon are displayed in the field.)

#### Note

Most of the fields should be self-explanatory. However, the following may need some clarification:

- **Default**: When checked, this type appears as the default selection in the Type field when users create a new Document. Check only one type.
- **Hidden**: If checked, the type will not appear in select lists and users will not be able to select it as the Document type for new or existing Documents. This is mainly for legacy types which have been applied to Documents in the past, but which are no longer valid and should not be applied to new or existing Documents. (Documents with a legacy type assigned them will keep that type as long as the type definition exists in the configuration. Users can change such Documents to any of the types not marked as Hidden in this configuration.)

## **Configure Document Workflow**

#### Note

This topic covers workflow configuration for **Documents**. Do not confuse this with workflow configuration for Work Items.

Workflow controls a process to ensure that no steps are missed or skipped. Workflows for Documents define and control process for entire Documents. Document workflow applies only to the Document, and is separate and apart from any workflow defined for the Work Items it contains. It enables you to trace the history of a Document's progress from inception to completion of whatever process you need for Documents.

A workflow consists of a set of named statuses and status transitions, transition conditions and dependencies that a Document passes through in its life cycle. For example, consider the following set of status definitions:

- 1. **Draft** (which might be the initial status for new Documents)
- 2. In Review
- 3. Reviewed
- 4. Rejected

If a user changes the Document's status from **Draft** to **In Review**, or **Reviewed**, this invokes an action (**Send to Review**, for example). The action triggers a status **transition**, which is automatically logged in the Document's history with details such as date, time, user executing the action, etc. Actions may be configured to trigger some system function: creating a linked Work Item, or clearing some field's value, for example. The execution a workflow action can be conditional: one or more conditions are checked and must be satisfied before the user can execute the action. For example, if a Document has a custom field **Draft Finished**, a function could be configured to clear that field's value when the **Back to Draft** action is executed. A condition might check that the field value is not null, or is not earlier than some specified date.

You can create and customize Document workflows and transitions in several scopes: globally (for all projects), project-specific for individual Projects, and/or Type-specific, which applies only to a specific type of Document in a project. (Type-specific can be configured both globally and in projects.) Polarion looks for the most specific workflow definition first and proceeds toward the most generic in the following search sequence:

- 1. Project-specific and Work Item type-specific
- 2. Global and Work Item type-specific
- 3. Project-specific
- 4. Global

#### **Configuration Process Summary**

#### Generally you will perform the following operations in the order listed:

- 1. **Configure Document custom fields:** define any that will be needed for Documents in the scope. If your Document workflow will not have any functions and conditions that reference Document custom fields, you can skip this operation.
- 2. **Configure Statuses:** Determine the statuses a Document can have at various stages of its lifecycle (e.g. "Draft", "In Review", etc.) and create a Status definition for each one in the **Statuses** section of the Workflow Designer.

3. Configure Actions: Determine what actions are needed to transition a Document from one status to another throughout your Document authoring process, and create an Action definition for each one in the **Actions** section of the Workflow Designer.

Action names appear in the drop-down list of a Document's **Status** field in Document Properties. The name indicates to the user what transition will take place as a result of invoking the action. For example, a "Send to Review" action might transition a Document to an "In Review" status.

You can require users to log an electronic signature when invoking any Work Item action by checking the **Requires Signature** column on the respective row in the Actions table.

- 4. Configure Functions and Conditions: This is optional, depending on whether or not any system functions should execute when an action is invoked by users, and if the execution should be conditional. (For available conditions and functions, see: Workflow Conditions and Functions for Documents.
- 5. **Configure Transitions:** Now you can specify how the Document transitions from one status to another in the **Transitions** section of the Workflow Designer page.

## Tip

It is recommended that you create a sandbox project based on one of Polarion's default project templates and study the Document workflow configurations for different Document types. You may find that the default configuration meets your needs. If not, you can get a better idea of what you need to customize and how to implement your custom workflow

#### **Use the Transitions Matrix**

When defining transitions using the Workflow Designer's Transitions matrix, it's important to keep two things in mind:

- You are defining what transitions are allowed between any 2 Statuses.
- Therefore, you don't need to specify an action for every transition... just those transitions you want to allow to occur between 2 statuses.

Consider an example of a workflow for a Requirements Specification. Assume that, among others, the statuses **Draft** and **In Review** are defined.

It makes sense to allow transition from the **Draft** status to the **In Review** status, so at the intersection of these two, you specify the **Send to Review** action (which has already been defined in Actions).

Assuming the organization's authoring process does not allow transition directly from **Draft** to Reviewed, it would not make sense to specify Send to Review where Draft intersects Reviewed in the matrix. No action would be specified, and the transition cell should be left empty.

For some statuses, you may want to allow more than one possible transition. For example, suppose you have the status definitions Draft, Reviewed and Rejected. For Documents having the In Review status, let's say there are three possibilities in the process: a stakeholder can approve the Document and give it the **Reviewed** status, or decide to send it back for changes, giving it the Draft again, or reject the Document and give it the Rejected status. The workflow can be configured to support this process but allowing 3 transitions on the In Review status.

Transition from In Review status to Draft status via the Back to Draft action.

- Transition from In Review status to Reviewed status via the Reviewed action.
- Transition from In Review status to Rejected status via the Rejected action.

## **Configure Document custom fields**

You can create custom fields for Documents. Document Custom Fields are a property of each Document, and should not be confused with custom fields of Work Items, which are a property of Work Items. Administrators can define Document Custom Fields globally, as the default for all new Documents, and also in specific projects. As with all global configurations, project configurations of Document custom fields can override the global configuration.

Users can access Document Custom Fields in the Document Properties panel of the Document Sidebar in the Document Editor.

## Access Document custom fields configuration:

- 1. Open the project you wish to configure, or the repository to configure the Global scope. (You must have administrator permissions for the scope.)
- 2. Open Administration, expand Documents and Pages, and select Document Custom Fields.

The **Document Custom Fields** page displays a table listing the Document Custom Fields currently configured for the current scope. You can:

- Add new field definitions. Enter data in the last row of the table, or click the → icon to add a new empty row.
- Delete existing field definitions. Click the icon on the row of the file definition you wish to delete.
- Modify properties of existing field definitions.

### Notes on the configuration

- The ID column value must be unique per configuration.
- Changing the type of existing fields is not recommended if Documents already contain field data of a different, incompatible type.
- To allow the field to store multiple values, check the box in the **Multi** column.
- To make the field required, check the box in the **Required** column. If checked, users will be prompted to supply a value before they will be allowed to save changes to a Document if any required Document Custom Field does not contain a value.
- You can optionally specify a default value for each Document Custom Field. You might consider
  doing this for required fields. Be sure the value specified is compatible with the Type. If the Type
  is Enum, be sure the default value exists in the enumeration specified.
- Document custom fields are copied when a Document is reused or branched. Users will not be
  able to reuse or branch a Document if there are validation errors in any of its custom fields.
  Document custom fields are always editable in reused/branched Documents. Custom fields in a
  derived Document are ignored when the Document is updated.

Changes to Document custom fields and custom field values are automatically tracked in the Document history, and are show in comparisons of Document revisions.

## **Configure Signatures**

## **Configure Signatures overview**

# ALM REQ QA PRO

Some organizations have a formal review and sign-off process for specifications and other documents. Such a process can be supported in Polarion using electronic signatures, including secure signatures compliant with common regulatory standards. This is accomplished in the Document Workflow configuration. You can set the Requires Signature option on any configured Document workflow action - an action such as Mark Approved, for example. Some additional parameters can be set specifying a signature policy. It is also possible to configure a default set of invited reviewers/signers via a workflow function with appropriate parameters.

When this option is set for a Document workflow action, the action cannot be invoked by Document users, and the Document cannot transition to the next status in the workflow until the necessary users have added their electronic signature, and the signature policy is satisfied. For an example, if the workflow has an action "Mark Approved" which transitions the Document to the status "Approved", the status transition is blocked until invited reviewers have signed off, and the signature policy is met. If the policy requires that all the invited signers must sign, the transition is blocked until all invited people have electronically signed approving the transition.

## Tip

Document-related signatures are U.S. CFR 21 Part 11 compliant secure electronic signatures.

For security, it is highly recommended to use the https protocol for access to Polarion when your system is configured to use electronic signatures.

You can configure any Document workflow action to require signature(s) before the status transition. Whether or not a given workflow action requires signatures depends on the process adopted by the project team or organization. Different stages in the process may require the signatures of a different set of signers, which means there could be several rounds of review and sign-off before a Document is considered "final" or "approved". It is the role of the Document owner or manager to decide which action or actions require signatures, to invite people to review and sign electronically, and after all signatures are collected, to transition the Document to the next workflow status. The administrator's role is to configure the Document workflow according to the required process, and to configure the Requires Signature option for the action or actions that require signatures.

For additional information see Sign Documents.

#### **Process overview**

The main steps in the process to configure Document signatures is as follows:

Configure Document types. The project owner or leader should decide what types are needed for the project. Some default types are pre-defined in most project templates. You can modify

the default types, add new types, or remove existing types as needed to meet the needs of the project. For information, see Configure Document Types.

- Configure Document workflows. Again, the project management should specify their process for each Document type - what statuses each Document type can have, and the actions that can transition the Document to them. You create a workflow configuration that applies to all Document types, and type-specific workflows for different Document types. For information, see Configure Document Workflows.
- Configure workflow actions. Your project management should specify which Document workflow actions in which Document types require signatures before the actions can be invoked by users to transition the Document status. The following sections explain how to do this.

## Configure a workflow action

This section assumes that Document workflow has already been configured according to the requirements of the project leader or manager, who has also specified which workflow actions must require signatures before the Document status can transition, and the signature policy.

To access the configuration:

- 1. Log in with administrator permissions for the scope you are configuring (global, or a specific project) and enter Administration.
- 2. In Navigation, expand the Documents and Pages topic and select Document Workflow.
- 3. Assuming that Document types have been defined as previously discussed, click the **Edit** button on the row of the type you want to support signatures. (If the workflow and signature policy are the same for all types, then edit -- **All Types** -.)
- 4. Scroll the page until you see the **Actions** section. This is where you configure one or more workflow actions to require signatures.

### **Enable Signatures**

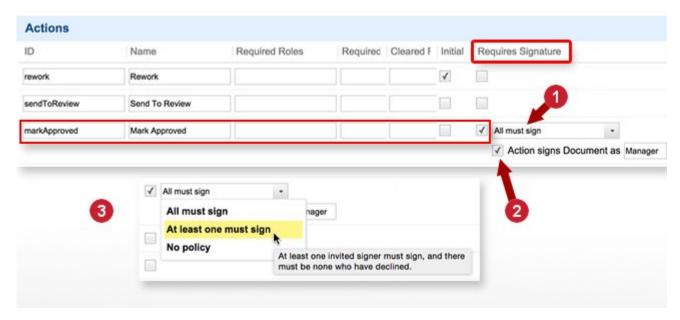
The Actions section of the page is formatted as a table. Each configured workflow action occupies a row. The table has a column named Requires Signature, which contains a check box. Locate the row of the action(s) that must require signatures, and check this box.

When you check Requires Signature, two additional fields appear in the column (see figure below).

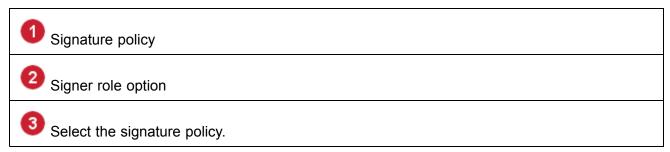
- Signature policy list
- Signer role option

#### Tip

Depending on your screen resolution, you may need to resize the Requires Signature column wider to see these fields. Drag the column separator in the column header.



How to enable signatures and set signature policy on a workflow action.



The signature policy options provide some restrictions on when the workflow action can be invoked. Hints appear when you hover over an item in the list, which should be mostly self-explanatory. The No Policy item is the least restrictive. It means that the Document owner can invite people to review and sign a Document before it transitions to a new Status, but the owner can invoke the workflow action and change the Document status whether or not anyone signs. For example, the team's process might specify that if no one signs a specification within 5 business days after being invited, Documents can move forward in the process. If some invitees do sign the Document, that will be recorded in the history.

The signer role option provides and optional additional datum that may be desirable in some situations, particularly in regulated environments. When enabled, you can specify the role of the people who sign the workflow action. The actual value should be specified by the project owner along with the other signature requirements. For example, if signatures are required to transition a Document from **Draft** to **In Review**, a value such as **Team Leader** might be appropriate. If signatures are required to transition from In Review to Approved, a value like "Project Lead" or "Analyst" might be specified.

## **Automate invited signers**

This is an advanced topic requiring some familiarity with Workflow functions and conditions. In cases where the set of people who should always be invited to sign workflow actions is static, you can configure the action with the AddDefaultsigners function. The function selects users according to its parameter settings, and adds them to the list of invited signers in the Signatures sidebar when

the Document enters the relevant status. The users may be specified in function parameters as a list of user IDs, or a list of user roles, or both.

For example, suppose the same users should always be added as signers when a requirements specification Document enters the "In Review" status. These users must all sign before the "Mark Approved" action can be invoked, and the Document transitioned to "Approved" status. Further suppose that all these users have the role "project\_approver" assigned to them for the project. You can specify that role in the relevant function parameter (details below), causing all users with that role be added as invited signers when a Document enters a Status from which that Action is allowed in the workflow configuration.

Suppose that in addition to people with the "project\_approver" role, one or two other people should always be added as signers for the workflow action. You can specify their user IDs in another function parameter, and they will also be added.

To configure automatic addition of invited signers:

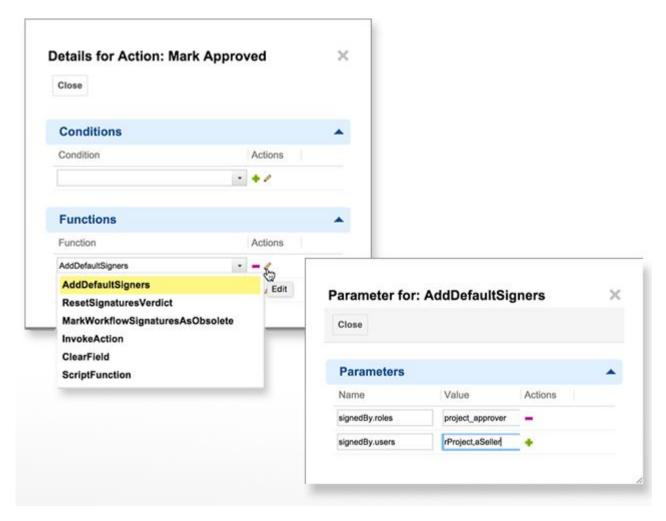
- 1. Before starting, make a note of the user role(s) and/or user IDs you want to specify in this configuration.
- 2. Open the Workflow Designer ( ♣ Administration → ☐ Documents and Pages → Document Workflow) and edit the Document type you want to configure. The Requires Signature option should be checked, and the signature policy selected.
- 3. Click the ∠ (Edit) icon on the action's row to open the **Details for Action** dialog.
- 4. In the Functions section, select AddDefaultSigners in the Functions column.
- 5. Click the (Edit) icon to open the **Parameter for** dialog.
- 6. If default invited signers should have a specific role, then in the **Name** column enter the function parameter name: signedBy.roles, and in the **Value** column enter one or more user role IDs. For example, project approver. If you specify multiple role IDs, separate them with a comma.

If you want default invited signers to be specified explicitly, then in the **Name** column enter the function parameter name: signedBy.users, and in the **Value** column enter one or more user IDs. If you specify multiple user IDs, separate them with a comma.

#### Note

You can specify both parameters. For example, the "project\_approver" role might add all but one or two necessary people as default reviewers/signers. Supposing that the other people have a different role, but not everyone with that role is needed, you can add just the IDs of the necessary users in addition to people with the specified project role.

7. Click **Close** in both dialogs and save the Document workflow configuration.



User Document workflow function and parameters to add a default set of reviewer/signers to an action (Add parameter(s) needed to automate addition of default signers to workflow action.)

### Warning

Users with the specified roles and/or IDs are only added to the list of invitees who should sign a workflow action. These users are not automatically notified to review and sign.

The Document owner or manager must still contact the invitees at the appropriate time and invite their review and sign-off. (See Invite People to Sign.)

See also: Workflow Functions (Documents).

#### **User permissions for Signatures**

There are two configurable permissions in Administration User Management Permissions Management that you can set to control which users can sign Documents:

Permission to SIGN/DECLINE - this permission allows user to sign or decline document

 Permission to MANAGE SIGNATURES - this permission allows user to invite/remove signer or reset verdict on document

These permissions can be set in global and/or project configuration. They work for Documents and also Custom sets of Documents, and they are also applicable for the dynamic role document author.

## **Document Signatures custom set**

In projects based on Polarion Project Templates, requirements specification type Documents are tied to a permissions Custom Set named **Submitted and Released Requirements Specifications**. The permissions allowing modification of this type of Document are denied for most roles. This, in effect, makes the Document read-only when the workflow status is any except **Draft**.

## **Configure Document merge actions**

## **Merge terminology**

Users with the requisite permissions can merge Work Item content between two Documents. A default set of merge actions is available to users. It is defined in the configuration file actions.xml. This file specifies which actions are available in the Merge sidebar for a Document that is the target of the merge action a user want to perform. The Merge sidebar is accessible when comparing two Documents.

Administrators can modify the configuration to exclude any of the actions defined in this file. For example, if the project's managers decide that copying a Work Item should not be allowed, the **Copy the Work Item** action can be excluded in the configuration. When excluded, the action is hidden in the sidebar, even if it would otherwise be a valid action. Allowed actions can optionally be conditional, allowed only if the conditions described in the <conditions> element of actions are satisfied. You can find more detailed information in comments at the end of the actions.xml configuration file.

In the same configuration file you can also control merging of Work Item **fields**, specifying which fields should and should not be included when merging Work Items.

This configuration is available in both global and project scope, and requires editing the underlying XML configuration file. You can do that either online in Administration, or offline in an XML editor, uploading the modified file back to the Administration page.

## **Terminology**

The actions.xml file contains explanatory comments and action descriptions that use some terms that it is important to understand.

- **Source Document**: the Document the user is merging content FROM.
- Source item: a Work Item contained in or appearing via reference in the Source Document.
- Target Document: the Document the user is merging content TO.
- **Target item**: a Work item contained in or appearing via reference in the Target Document.
- Overwritten item: a Work Item in the target Document that is referenced from the source Document, but the content is changed in the target Document.

## Working with the configuration file

To access the configuration file:

- 1. Log in with administrator permissions for the scope you want to configure, and open the project (or Repository for a global configuration).
- 2. Enter Administration, and in Navigation select Documents and Pages Merge Actions.

In the editor on the page, you modify the XML code as necessary and save the change. Alternatively, you can use the link on the page to download the XML file, modify it offline, and upload it back to the page.

Configurable elements include:

- The action's title and description (<title> and <description> elements). Users see these strings in panels inside the Merge sidebar.
- The background color of the action selector in the Merge sidebar, via the color attribute of an <action> element. The value is a CSS-style hexadecimal... #FFFFFF, for example.
- The color of text (foreground) in an action selector in the Merge sidebar (rendered from the <title> and <description> elements of an <action> element). Specified in the text-color attribute of an <action> element.

Example of the color attributes in an action element:

```
<action id="com.polarion.unmarkWorkItem" color="#c0392b" text-color="#FFFFFF">
```

### **Disable Merge actions**

If managers decide that one or more of the default set of merge actions should not be available to users, simply enclose the relevant <action> element(s) in a XML comment marker: <!-- -->

#### **Example:**

```
<!-- <action id="com.polarion.deleteWorkItem" color="#c0392b" text-color="#FFFFFF">
   <title>Delete Work Item</title>
    <description>
        Delete Work Item from the Document and the repository (but not from History)
   </description>
</action> -->
```

### aiT

Comment out actions rather than delete them. Should things change in the future, actions can then be easily enabled by removing the comment markers.

## **Use Merge action conditions**

You can optionally configure merge actions to be conditional contingent upon some condition(s) being satisfied. The <conditions> element can contain one or more <condition> elements and logical operator tags to define simple or complex conditions, including the result of a query on Work Items. The following points are important to keep in mind when specifying conditional merge actions:

- The <conditions> element is a child of the <action> element. Only one <conditions> element is allowed within any <action> element. Any that appear after the first one are ignored.
- The <conditions> element can contain multiple <condition> elements.
- You can construct complex conditions by using nested <and>, <or>, and <not> tags within a <condition> element.
- The <not> tag can contain multiple tags and the result is the same as if they were inside an <and> tag. Therefore, the <not> tag actually works like a NAND logical operator.
- Empty <and>, <or> tags evaluate to "true". Empty <not> tags evaluates to "false".
- Unknown or invalid elements and tags, and <condition> elements having an unknown/invalid id attribute value are ignored.

## **Conditions configuration format**:

The following example also shows the possible condition ID values, except for query-based conditions - information for that follows the listing.

### **Query-based Merge action conditions:**

If a <condition> element has one of the following values specified in its id attribute, the element can contain a Lucene query on the respective artifact (see Example, below):

- com.polarion.sourceWorkItemQuery The element value should be a query on the attributes of the Source Work Item.
- com.polarion.targetWorkItemQuery The element value should be a query on the attributes of the Target Work Item.
- com.polarion.sourceDocumentQuery The element value should be a query on attributes of the Source Document.
- com.polarion.sourceDocumentQuery The element value should be a query on attributes of the Target Document.

#### **Example:**

```
<action id="...">
    <conditions>
       <-- Work Item queries: -->
       <condition id="com.polarion.sourceWorkItemQuery">type:task</condition>
        <condition id="com.polarion.targetWorkItemQuery">type:requirement AND
status:draft</condition>
        <-- Document gueries: -->
        <condition id="com.polarion.sourceDocumentQuery">type:generic</condition>
        <condition id="com.polarion.targetDocumentQuery">type:test-spec AND NOT
status:draft</condition>
   </conditions>
</action>
```

#### Note

- If no query is specified in the element (i.e. the query is empty), the condition is ignored.
- If a query is specified, but no Work Item exists on the relevant merge side for the current merge action, then the condition is ignored even if the query can return some results in some other context.

## Configure Work Item field merging

When you merge Work Items between two Documents, you may want to exert some control over which fields are merged. Two actions for fields are present and enabled by default in the actions.xml file:

- Merge All Fields: merges all supported Work Item fields. You can modify the description, or comment it out to if the action should not be available to users.
- Merge Some Fields: merges only the fields specified in the com.polarion.mergeFieldsAction in the actions.xml file.

You specify the fields to be copied in <field> elements within the <fields> element. Other fields are skipped. You can also explicitly specify field that should not be copied in <field> elements within the <excluded-fields> element.

See also: Fields Not Supported for Merging.

#### **Note**

When comparing Documents across different projects, merge actions use settings from the project where the Target is stored.

## **Configure PDF Export**

### **Overview - Configure PDF Export**

You can configure the output of all types of pages and Documents exported to PDF. You can specify header and footer content and formatting as well as some content formatting. You can optionally

include a watermark image in the PDF output. The configuration is available in the Repository (global) scope and for each project. The configuration is performed by editing an XML configuration file, on or off line.

To access the configuration file:

- 1. Open the project you want to configure, or select the Repository scope in the Open Project dialog.
- 2. Click the Administration link in the tool view of Navigation. (It appears only if you have administrator permissions for the scope you are working in.)
- 3. In the Navigation panel, expand the Documents and Pages topic and select PDF Export Configuration.
- 4. If working in the project scope, and no project configuration exists yet, click **Paste global configuration** to import the XML of the global configuration into the project.
- 5. In the XML editor field, edit the configuration as desired. You will see instructional comments in the configuration file. Save the changed file.

### Tip

You can optionally download the XML configuration file using the link found on the Configuration page, edit the configuration offline using your preferred editor, and then upload the modified configuration file via the file selection and upload controls in the Upload... section of the page.

By default, the configuration implements a simple HTML table, presented in CDATA elements inside the <header> and <footer> elements of the XML. You can optionally modify the table structure to add rows or columns, or use some other basic HTML elements such as paragraphs. In the table, you can specify such things as width, alignment of cells, cell padding etc. as long as you use standard HTML.

There are several properties you can specify, which embed data values into the header or footer of generated PDF documents. For example, the default configuration references <code>\$[documentTitle]</code> in the <code><header></code> element, which causes the title of the exported Document or page to appear in the header of the generated PDF document. The full list of available properties is documented in: PDF Export Header/Footer Properties.

#### qiT

Power users who know XML can configure PDF header and footer for individual Documents or pages. The export dialog contains a link **Configure**..., which opens the **Configure Headers**, **Footers and Watermarks** dialog, which contains an embedded XML editor. The current Document or page configuration is pre-loaded in the editor. If there is no Document or page configuration yet, the XML from the project configuration is loaded, and the user can modify it and save it as the Document or page configuration.

Users of non-ASCII character sets may prefer to use the  $\sharp[documentTitle]$  property rather than the  $\sharp[documentName]$  property, which supports only ASCII characters.

## **Reference Page Variables and Parameters**

You can declare custom Velocity variables and/or page parameters in classic Wiki pages, and reference them in the header/footer export configuration. By using conditional logic in the configuration, you can include or exclude content of the variables or page parameters in the header/footer of exported documents.

For example, you could include one or both of the following in some Wiki page(s):

```
#set ($confidential = "true")
{parameter:internalDoc|type=boolean|value=true}
```

Then, in the Wiki export configuration you could reference these as shown in the following examples:

```
<!-- Example referencing Velocity variable -->
#if($confidential)
  $[projectName] $[documentTitle] (rev. $[revision])
CONFIDENTIAL
  $[projectName] $[documentTitle] (rev. $[revision])
#end
<!-- Example referencing page parameter -->
#if($pageParameters.internalDoc.equals(true))
 $[projectName] $[documentTitle] (rev. $[revision])
INTERNAL USE ONLY
#else
  $[projectName] $[documentTitle] (rev. $[revision])
#end
<!-- Example referencing both -->
#if($confidential)
  $[projectName] $[documentTitle] (rev. $[revision])
CONFIDENTIAL
#else
  #if($pageParameters.internalDoc.equals(true))
    $[projectName] $[documentTitle] (rev. $[revision]
INTERNAL USE ONLY
  $[projectName] $[documentTitle] (rev. $[revision])
#end
```

#### Tip

If a Document has no value in the Title field referenced by \$[documentTitle], the value of the Document name (ID) will be used in exports.

#### Change embedded fonts in exported PDF

This advanced re-configuration may be necessary if you are exporting documents containing characters from various languages which need to be supported in the PDF. It requires editing of the main configuration properties file polarion properties, and generation of a properties file using

Java. In the following steps [POLARION\_PLUGINS\_DIR] refers to the location of the "plugins" folder in your installation.

1. In the polarion.properties file, set the configuration property pdf.export.fonts.dir to the location of the folder containing the fonts that you want to use in exported PDF.

```
IMPORTANT: specify the folder as a file URL. E.g., file:///fonts/directory
```

2. Generate a configuration file for PD4 by running the following Java command from your command line:

```
java -jar POLARION_PLUGINS_DIR/com.polarion.alm.ui_*/lib/pd4ml.jar -configure.fonts
/fonts/directory
```

3. The pd4fonts.properties file is generated in the folder specified as the argument to -configure.fonts in the java command above. Edit this file and map the default font names (Times New Roman, Arial and Courier New) to the names of the custom fonts you want embedded in PDF export.

## **Specify PDF margin size**

You can optionally specify the width of all page margins in the PDF output. Margin widths are specified as attributes of the <pdf> element. For example: <pdf marginLeft="40" marginTop="25" marginRight="20" marginBottom="25">. Place the <pdf> element immediately following the <export> element:

Any margins not specified default to 20 pixels in the PDF output.

### Configure different first page header and footers

The scope attribute of the header and footer elements in the PDF export configuration, applicable for projects and individual documents, enables specification of different a header/footer for the first page of an exported PDF document. Modifiers even, odd and skiplast can be optionally used.

## For example:

```
<a style="color:rgb(128,128,128); font-size: 8px !important;"</pre>
href="https://polarion.plm.automation.siemens.com">Polarion ALM</a>
       ]]>
</header>
# Header for other pages:
<header height="35" scope="2-72, even, skiplast"><! [CDATA[</pre>
   <table width="100%" style="color:rgb(128,128,128); font-size: 9px !important;
margin-bottom: 20px;">
       <img width="30" height="32"
src="/polarion/ria/images/logo.gif">
          $[projectName] $[documentTitle] (rev. $[rev
       ]]>
</header>
```

## **Configure export watermarks**

You can create or modify the global or project-scope PDF Export Configuration to include a watermark image on the pages of exported PDF documents. Users with permission to read or modify Documents can enable or disable an option **Include header, footer and watermark**, which appears in the **Export to PDF** dialog, and subsequently edit the export configuration to use a watermark image if one is not yet specified, or a different watermark image of one is specified, and/or change the opacity of a specified watermark. The configuration involves the following main steps:

- 1. Upload a watermark image to the required repository folder location, the name of which serves as the watermark ID referenced in a PDF Export Configuration.
- 2. Add a <watermark> element to the global and/or project PDF Export Configuration at the required place in the XML code.

## Warning

The file name of the watermark image file must correspond to the name of the folder that contains it. For example, if the watermark folder is confidential, the valid image file names would be **confidential.png** or **confidential.jpg** 

### Add a watermark image to the repository

To add a watermark image to the repository:

- 1. If performing a project-specific configuration, open the desired project with project administrator permissions. If performing a global configuration, open the Repository.
- 2. Open the Repository Browser, and navigate to .polarion/wiki/. If the wiki folder/directory does not exist, use the Add Directory icon to add it as a sub-directory of .polarion.
- 3. If it does not already exist, create another new sub-directory, under .polarion/wiki/, name it watermarks and navigate into it.

- 4. Create a new sub-directory for the watermark image you want to use in PDF exports for the scope you are configuring. The name of this folder/directory is used as the watermark ID in the PDF Export Configuration, therefore it must not contain spaces. For example, for a watermark that identifies a document as "CONFIDENTIAL", name the folder/directory something like confidential.
- 5. Upload the desired watermark image file to this folder/directory using the Upload File icon of the Repository Browser. The image should be a .PNG file with a transparent or white background, or .JPG with white background. Be sure that the file name matches the name of the containing folder.

When you finish these steps, the Repository Browser should be showing a path to an image file that is something like:

[PROJECT NAME]/.polarion/wiki/watermarks/confidential/confidential.png (project-scope configuration)

## Include a watermark in PDF export

To configure the project PDF export to include a watermark:

- 1. Log in with administrator permissions for the scope you want to configure (repository or project), and access the repository or project.
- 2. Open **Administration** → **BDocuments** & Pages → PDF Export Configuration.
- 3. If working in a project and there is no project configuration yet, click the **Paste Global Configuration** to load the XML code of the global PDF Export Configuration into the XML editor of the page.
- 4. In the first <pdf> element, immediately following the closing </footer> tag, enter the following on its own line:

<watermark id="watermark\_directory"/> (replace watermark\_directory with the name of the
repository directory containing the watermark image you want to appear in PDF exports).

```
Example: <watermark id="confidential"/>
```

```
Example: <watermark id="confidential" opacity="33"/>
```

6. Save the configuration.

Here is an example of showing where the watermark element should be placed:

Documents will now export to PDF with the configured watermark image. A global configuration may be overridden in individual projects with a project-scope configuration. Users with permissions to read or modify Documents will see the project-scope PDF Export Configuration when they invoke PDF

export (or the global configuration of no project-specific configuration exists), and they can modify the XML they are presented. Their options include choosing not to include any watermark (via the Include header, footer and watermark option in the export dialog).

You are not confined to using a single watermark image for all PDF exports. You can optionally set up to use different images for different PDF export options, and create multiple image sets to support different watermarking requirements. The next sections describe how to do this.

## Multiple watermark images

You can have multiple sets of watermark images in the repository at the same time, to be used automatically by the Exporter when end users invoke different export scenarios: different paper size or orientation, for example. For this, you create a folder/directory under .polarion/wiki/watermarks and populate it with appropriately named image files that are automatically applied to the output when a user selects different export options. The repository directory and image files should have this form: .polarion/wiki/watermarks/<watermarkId>/<watermarkId orientation papersize.extension>, Where folder watermarkId is a valid Subversion repository folder name.

Within that folder you can have multiple image files using the file name convention previously mentioned, where:

- watermarkId is the same as the containing folder name.
- orientation is one of these literals: portrait or landscape.
- papersize is one of these literals: A4, A3, letter, legal, or tabloid.
- extension is one of these literals: png or jpg according to the actual file type.

#### Valid file name patters are:

- <watermarkId orientation papersize.extension>
- <watermarkId orientation.extension>
- <watermarkId.extension>

Image file names that do not correspond to the pattern (e.g. there is only the paper size element in the name, but no orientation) will be ignored. For example, confidential portrait.png, confidential landscape.png, or confidential.png are valid, but confidential A4.png is not, and will be ignored if specified in a PDF export configuration.

### Repository setup example:

You might have a set of images that show the word "CONFIDENTIAL" in exported documents, with different images to be applied for different PDF export options selected by users when exporting. An example of a repository folder for this is .polarion/wiki/watermarks/confidential. In that folder you could have the following image files:

- confidential\_portrait\_A4.png
- confidential\_landscape\_A4.png
- confidential\_portrait\_letter.png

## confidential\_landscape\_letter.png

## **Extend for different requirements:**

It is possible to extend the basic concept. For example, you could add another folder, .polarion/wiki/watermarks/secret and populate it with another set of watermark images (e.g. secret\_portrait\_A4.png, secret\_landscape\_A4.png, etc.)

Each PDF export configuration element in Administration Documents & Pages PDF Export Configuration can reference one set of watermark images for each export type by a short name which corresponds to the name of the repository folder that contains the image files. For example: "confidential" for images contained in .polarion/wiki/watermarks/confidential/, or "secret" for images contained in .polarion/wiki/watermarks/secret. Syntax for each reference is: <watermark id="[id]" opacity="<opacity number>"/>.

- The id attribute is required.
- The opacity attribute is optional. Valid value range is 1-100; if not specified, the default is 50.

See also: Including a Watermark in PDF Export.

## How watermarks are applied:

Assuming compliance with the repository folder setup and image naming convention, Polarion automatically applies the most appropriate watermark image from the folder represented by the *id* attribute *when an end user executes an export to PDF*. For example, if the user invokes the export operation specifying landscape orientation and letter size paper, Polarion looks for an image file in the repository the file name of which contains "landscape" and "letter" according to the aforementioned naming convention. If not found, it searches for the closest match to the configuration.

Following the previous example, if the PDF export configuration contains <watermark id="confidential" opacity="33"/>, if the user invokes PDF Export and selects options for Landscape orientation and Letter paper size, Polarion will apply the image confidential\_letter\_landscape.png with 33 percent opacity to pages in the PDF output file. If that image is not present, will look among the remaining available images for one with the closest match to the configuration and user-selected export parameters. If none is found in the project, the global scope is checked, and if no appropriate image is present, no watermark is displayed in the output, and an error is written to the log files.

#### Tip

You can set up the export configuration discussed here in both global and project scopes. These defaults may be overridden by a user in a document-specific configuration when exporting a Document or Work Items. If no project-scope images are available, images from the global configuration are used.

It is possible to override global-scope watermarks by creating a watermark folder in the project repository having the same name, and containing images having the same file names as those in the global configuration. By doing this, you can use the same XML element in global and project PDF export configurations (e.g. <watermark id="confidential" opacity="35"/>), and the image from the appropriate scope will be used. For more detailed information, see PDF Watermarks Order of Precedence.

Organizations can optionally implement restrictions via the plugin API to override PDF export configurations, placing restrictions on individual exports by individual users, to

include a different watermark in the exported documents and/or to conform to other required policies. If such a plugin is in use, then when a use invokes PDF export, the PDF **Export Wizard** dialog shows a message: "Company policy can override these settings".

## **Configure Highchart export**

Polarion integrates the Highcharts library, which enables users to easily create and show various types of simple and complex charts in LiveReport type Pages and Classic Wiki pages. Phantom JS, used for preview and PDF export of Highcharts, is also installed. By default, it starts automatically, and automatically restarts in case the Phantom JS process is killed. Administrators may optionally disable auto-startup and auto-restart by setting the following system property in the polarion properties file (follow link if you need the location):

```
com.polarion.highchartExportServer.autoStart=false
```

If set this way, the administrator is then responsible for starting/restarting the process manually. If not started, then users exporting pages containing Highcharts will not be able to preview them, such pages will not export to PDF properly, and errors will be logged.

To manually restart the process, issue one of the following system commands, depending on your operating system:

- Windows: phantomjs.exe highcharts-convert.js -host 127.0.0.1 -port 34567
- Linux: ./phantomjs highcharts-convert.js -host 127.0.0.1 -port 34567

The library is located under the plugin folder com.polarion.alm.ui. The path is polarion\plugins\com.polarion.alm.ui N.N.N\phantomjs\, where "N.N.N" is the number of your currently installed Polarion version (e.g. "3.8.0").

There are 2 additional system properties for Phantom JS, both optional:

- com.polarion.highchartExportServer.host Defines the host name that will be used by the chart renderer server. The default is 127.0.0.1.
- com.polarion.highchartExportServer.port Defines the port that will be used by the chart renderer server. The default is 34567. Administrators may change this value to prevent conflict if that port is already in use.

### Kill phantomis Process on Restart

During Polarion start-up, the external phantomis process starts and binds to localhost:34567. If the Polarion server process is killed or somehow terminates abnormally, the phantomis process has to be terminated before restarting Polarion. The system does this automatically on restart if the system property com.siemens.polarion.ui.killPhantomJsProcesses is set to "true" (the default).

If you system uses separate phantomis processes, which would be killed by this default behavior, you can set the value of the system property to "false". If you do, then before restarting Polarion, you should terminate the phantomis process manually to prevent multiple instances, which can result in phantomis failing to start, and/or problems with binding to the port. Use one of the following command-line commands:

Linux: pkill -f phantomjs

Windows: taskkill /F /IM phantomjs.exe /T

After Polarion restarts, re-initialize any separate phantomis process.

## Configure attachment size limit

You can limit the maximum file size of attachments to Pages, and all other content that accepts attachments. See Advanced System Tuning.

## **Configure formulas**

## Configure the NodeJs server for formulas

Polarion uses NodeJS and the MathJax library to convert LaTeX mathematical formulas into SVG images. NodeJS starts automatically by default, and automatically restarts if the NodeJS process is killed. Administrators can disable the auto-startup and auto-restart of NodeJS by setting the following system property in the polarion.properties file:

com.siemens.polarion.nodeJsServer.autoStart=false

#### Caution

If set to false, administrators must start or restart the process manually. If it is not started, then users will not be able to see formulas and they will be replaced with a warning image in both Polarion and exported artifacts. (PDF or Microsoft Word.) An error message will also be logged.

To manually restart the process, issue one of the following system commands:

Windows: node.exe mj-formula-convert.js -host 127.0.0.1 -port 34568

Linux: ./node mj-formula-convert.js -host 127.0.0.1 -port 34568

There are three additional, optional system properties for NodeJS:

com.siemens.polarion.nodeJsServer.host

Defines the host name used by the NodeJS server. The default is 127.0.0.1.

com.siemens.polarion.nodeJsServer.port

Defines the port used by the NodeJS server. The default is 34568. Administrators may change this value to prevent conflicts if the port is already in use.

```
com.siemens.polarion.nodeJsServer.queueLength
```

Defines the maximum number of connection requests that can be queued to the NodeJS server before the system starts rejecting incoming requests. The default is 511.

### Kill NodeJS process on restart

During Polarion start-up, the external NodeJS process starts and binds to localhost:34568. If the Polarion server process is killed or somehow terminates abnormally, the NodeJS process must be terminated before restarting Polarion. The system does this automatically when restarted if the com.siemens.polarion.nodeJsServer.killoldProcess system property is set to true. (The default.)

If your system uses a separate NodeJS processes that would be killed by this default behavior, you can set the value of the system property to false. If you do, then before restarting Polarion, you should terminate the NodeJS process manually to prevent multiple instances, which can result in NodeJS failing to start, or problems with the port binding. Use one of the following command-line commands:

Windows: taskkill /F /IM node.exe /T

Linux: pkill -f node

After Polarion restarts, re-initialize any separate NodeJS process.

## **Configure notifications**

## Notifications, events and targets

Polarion can send email notifications to users when changes occur to Work Items. Certain kinds of changes trigger notification events. Examples of changes that trigger notification events include:

- A new Work Item is created or an existing Work Item is updated: its status, assignee, or approval state changes, for example.
- A new user is created.
- A new Document is created in a project.
- A build finishes or fails.

Notification events have targets which represent some user or aggregation of users. For example, a status-changed event normally has a target representing the author of the Work Item. Therefore, when a change in the Work Item's status triggers the event, a notification is sent to the Work Item's author (i.e. the user who created the item). Some targets represent an aggregation of users. The all-watchers target encapsulates all users who have a Watch set on a Work Item.

Event's can have multiple targets. For example the **status-changed** event might also have targets "all-watchers" and "all-voters" in addition to the current-author target. If the same user happens to be in multiple targets for the same event, only one notification is sent to that user.

#### Note

For users that are both **Included** and **Excluded**, the general rule is as follows:

When a user is in a group that is both included and excluded, then by default they will not receive notifications.

The default notification scheme is designed to be sufficient for many organizations, but may not be exactly what yours needs. For example, you may want to reduce the overall number of notifications going across your network, or add targets, or change default targets of some events. Therefore, the notification scheme is configurable. This chapter explains how to access and perform configuration of notifications.

## **Enable E-mail notifications**

Before Polarion can send email notification to users, it is necessary to enable the notifications feature, specifying the SMTP server and other parameters. This setup is typically done during the installation and initial configuration process, so details are provided in the PDF installation guide document bundled with Polarion distributions. On Windows systems, the full installer presents a screen for enabling notifications and specifying basic configuration settings.

Once you have email notifications enabled, you can proceed to customize the configuration for the notifications feature as described in the remainder of this chapter.

## Access the configuration

Basic notification properties are configured in the global scope for each Polarion repository. Targets can be configured both globally, and in individual projects to override the global configuration.

- 1. Make sure you are logged in with administrator permissions for the scope you want to configure.
- 2. Enter the Administration interface (click the \*Administration link in the tool view of Navigation).
- 3. In the **Navigation** pane, expand **Notifications**. If you are working in the global (Repository) scope, the following Navigation links appear under Notifications:
  - **Settings** global configuration for some basic settings related to properties of emails sent by the system.
  - Events and Targets global configuration of targets for notification events.
  - Layouts global configuration for the layout of email notification messages for notifications related to Work Items.

If you are working in the project scope, only the **Events and Targets** and **Layouts** topics are present, providing access to the project-scope configuration for these items. Settings in the project scope override the same settings in the global configuration.

## **Configure settings**

The **Settings** page provides fields enabling you to specify properties of notification email messages such as a prefix for the subject line and an email address to be used as the sender address in outgoing emails. You can refer to the Quick Help text embedded in the Settings page for information and examples.

## **Configure event targets**

The **Events and Targets** administration page provides table of all available notification events for the scope in which you are working. For each event, you can specify add targets or remove existing targets, define if targets are included or excluded from notifications, and set values for several Advanced properties for each event.

The objective here is to decide which events you want to have trigger notifications (and just as importantly, which events you *do not*) want triggering notifications, and which users should receive notifications as a result of the events you decide to include. You should make sure you include all the

desired events in the configuration, and remove any events for which you don't want any notifications triggered. For each event you include, you need to be sure that the target(s) include all the users who need to get notification about the event.

Reference information explaining the events and targets is available in the Events and Targets page where you configure the settings (scroll the page to see it). In particular, you will need to understand which actual users are represented by each target in order ensure that no one is left out who needs to get a notification.

## **Understanding the Events Table**

Each row in the Events table represents a notification event. The name of the event appears in **bold** type in the Event column.

The targets currently configured for each event are listed in a tabular format in the **Targets** column. Each target occupies a row in this sub-table. The last sub-table row enables you to add another target to the event using a drop-down list of targets. You can remove existing targets listed in this sub-table by clicking the - icon on the desired row. You can add multiple targets using the - icon to create a new row in the sub-table for an additional target.

You can add events or remove events by adding or removing rows in the Events table. Remove an event row using the - icon in the row's Actions column. The last row in the table enables you to add an event, using a drop-down list of events to specify which event. (You will probably need to scroll down to see this row.) You can add multiple events using the • icon.

The following tips are useful to keep in mind:

- The in the Actions column is disabled in the project scope when there is no project-scope configuration and the project is using the global configuration. You need to create a project-scope configuration using the Create button that appears above the Events table when you are working in the project scope.
- Advanced settings are hidden by default. Click the **Show Advanced Settings** button in the page toolbar to show or hide these settings. Advanced settings have to do with setting priority values between multiple configuration elements for the same event. For details, see the embedded Help on the Notification Events and Targets administration page.

## Customize the layout of notification emails

Notification Layouts provide a way to customize the layout of email notification messages pertaining to new and updated Work Items. A Notification Layout defines what fields are displayed in notification email messages, and how those fields are arranged. Layouts can be configured in both repository and project scopes.

There are two default layouts, which you can customize, or you can create your own custom layouts by changing these files. To actually use a customized layout, you need to configure an optional attribute named layout for particular Work Item related events.

The Notification Layout Editor provides an embedded XML editor which enables you to configure the layout without leaving your browser. In-line Help contains extensive comments on the various elements and how to structure them to achieve the desired layout.

## **Create a new Notification Layout configuration**

- 1. Log in with administrator permissions for the scope you want to configure.
- 2. Open the repository or project you want to configure and enter the Administration interface if you are not already there when you log in.
- 3. In **Navigation**, expand **Notifications** and select **Layouts**. The Notification Layouts page appears, and displays a table of the existing notification layouts, if any.
- 4. Click the **Create New Notification Layout** button. The **Create New Notification Layout** dialog appears.
- 5. Select the Work Item type to which the new configuration is applicable, or if applicable to all types, select  **unspecific –**.
- 6. Enter a name for this Layout (do not use spaces or special characters) and click **Next**. The Notification Layout Configuration page loads in your browser.

You can now use the embedded XML editor in conjunction with the in-page Help text (scroll down if necessary) to define the layout configuration.

### Find users with disabled notifications

Users can individually opt out of receiving email notifications by checking the **Disable Notifications** option on their user account page. You as an administrator may need to find users who have disabled their accounts.

- 1. If you want to search users of a specific project, log in with administrator permissions for the desired project. Otherwise, log in the administrator permissions for the repository.
- 2. Open the scope you want to search (repository, or the desired project), and in **Navigation** select **User Management Users**.
- 3. You can optionally sort the table of users on the **Disabled Notifications** column by clicking on that column header. Or...
- 4. In the Search field of the User page, enter this search string disabledNotifications:true and click Search.

The table of users is filtered to show only those users who have checked the **Disable Notifications** option in their user accounts.

## **Configure SSL support**

## Install and configure SSL

You may decide that you want to use SSL to provide additional security for Polarion, especially in cases where you are going to set up LDAP access (see, Create User Accounts). This topic discusses how to configure SSL when using the Apache web server installed and/or configured when you

installed Polarion. Everything you need to set up SSL access to Polarion is bundled in the Polarion Windows installation, including default configuration files.

SSL access is configured by standard means in Apache. The only Polarion-specific configuration for SSL is to specify HTTPS protocol for the Subversion repository (described later on).

In-depth documentation on using SSL with Apache is beyond the scope of this Help. For a real understanding of everything described here, and to understand how to actually make an Apache web server secure, it is recommended that you refer to some more sophisticated material on the subject. You will find some links later in this topic that might do for starters.

### Warning

Leave the SSLSessionCacheTimeout value at 300 in Apache's configuration setting to avoid potential broken pipe exceptions.

## **Configure SSL on Linux**

The general apache2 configuration continues to evolve and varies depending on the distribution. See the links below for updated, distribution specific, instructions.

(All commands and edits should be done with sudo/su).

- **Debian Based Distributions**
- **Redhat Based Distributions**
- SUSE Based Distributions

Unless you've disabled the Resource Traceability feature that's now enabled by default, you will have to import a certificate into your Polarion Java TrustStore.

## To point Polarion to your TrustStore, use the following property:

```
-Djavax.net.ssl.trustStore
```

If your organization does not have a TrustStore, importing to the Java Key store will also work. See Import a Certificate to the Java Keystore.

(It must also be done for self-signed certificates.)

### Polarion specific settings

- 1. If you are using a certificate that is not from an authority trusted by Java, import the certificate to the Java keystore, as described in Import a Certificate to the Java Keystore.
  - On Linux, the keytool will be located in the same directory as the version of Java used by Polarion at \$JDK HOME.
- 2. Change the Subversion (SVN) protocol setting in Polarion system properties. Edit file \$POLARION HOME\$/polarion/etc/polarion.properties, and change the following properties, replacing http with https and removing references to port 80, if present:

```
base.url= https://[SERVERNAME] repo= https://[SERVERNAME]/repo/
```

Restart Apache and Polarion.

If you get an error similar to the following...

Failed to access the repository https://[SERVERNAME]/repo. Please check its availability, perhaps Apache is stopped? (svn: E175002: Connection has been shutdown: javax.net.ssl.SSLProtocolException: handshake alert: unrecognized name

Open the file /etc/apache2/sites-available/ssl and add: ServerName [YOUR SERVER NAME].

### For example:

ServerName polarion.dev.ourdomain.com

## **Configure SSL on Windows**

The Open SSL executable is installed at: \$POLARION HOME\$\bundled\apache\bin\openssl.exe

Unless you've disabled the Resource Traceability feature that's now enabled by default, you will have to import a certificate into your Polarion Java TrustStore.

## To point Polarion to your TrustStore, use the following property:

```
-Djavax.net.ssl.trustStore
```

If your organization does not have a TrustStore, importing to the Java Key store will also work. See Import a Certificate to the Java Keystore.

(It must also be done for self-signed certificates.)

- 1. Copy your certificate and keyfile to: \$POLARION HOME\$\bundled\apache\conf\.
- 2. Activate and configure SSL. Open file <code>\$POLARION\_HOME\$\bundled\apache\conf\httpd.conf</code> in a text editor and...
  - Search for the line containing: LoadModule ssl\_module modules/mod\_ssl.so and remove the leading comment marker (#). Uncommenting this line activates the SSL module.
  - Search for the line containing Include conf/extra/httpd-ssl.conf and remove the leading comment marker (#).
- 3. Optionally deactivate unencrypted http traffic. In \$POLARION\_HOME\$\bundled\apache\conf\httpd.conf, find the line Listen 80 and add a comment marker # at the beginning, leaving the line as: #Listen 80.
- 4. Set up the certificate file in SSL. Edit file \$POLARION\_HOME\$\bundled\apache\conf\extra\httpd-ssl.conf and search for SSLCertificateFile. Change the path to point to the certificate file. **Example**:

```
SSLCertificateFile "C:\Polarion\bundled\apache\conf\certificate.pem"
```

5. Set up the key file in SSL. Edit file <code>\$POLARION\_HOME\$\bundled\apache\conf\extra\httpd-ssl.conf</code> and search for <code>sslCertificateKeyFile</code>. Change line path to point to the certificate key file. Example:

SSLCertificateKeyFile "C:\Polarion\bundled\apache\conf\privateKey.key"

- 6. If you are using a certificate that is not from an authority trusted by Java (a self-signed certificate, for example), import the certificate to the Java keystore, as described in Import a Certificate to the Java Keystore.
- 7. Change the Subversion (SVN) protocol setting in Polarion system properties. Edit file POLARION HOME \$\polarion\configuration\polarion.properties, and change the following properties, replacing http with https and removing references to port 80, if present:

```
base.url= https://[SERVERNAME] repo= https://[SERVERNAME]/repo/
```

8. Restart Apache and Polarion.

If you get an error similar to the following...

Failed to access the repository https://[YOUR SERVER NAME]/repo. Please check its availability, perhaps Apache is stopped? (svn: E175002: Connection has been shutdown: javax.net.ssl.SSLProtocolException: handshake alert: unrecognized name

Open the file \$POLARION HOME\$\bundled\apache\conf\extra\httpd-ssl.conf and add: ServerName [YOUR SERVER NAME].

## For example:

ServerName polarion.dev.ourdomain.com

## Additional resources

Here are some additional resources with information on using https://host -

- Apache HTTP Server home: http://httpd.apache.org/
- mod ssl (Apache HTTPd) documentation: http://httpd.apache.org/docs-2.4/mod/mod ssl.html
- Apache SSL/TLS Encryption documentation, FAQ and SSL introduction: http://httpd.apache.org/docs-2.4/ssl/
- OpenSSL home: http://www.openssl.org/
- OpenSSL on Linux: https://help.ubuntu.com/community/OpenSSL

## Import a certificate to the Java Keystore

This section is referenced from previous sections on configuring SSL for Linux and Windows. It is important only if you are not using a SSL certificate that is signed by an authority that is trusted in Java. Use of a trusted certificate is preferred and recommended because use of a Java-untrusted certificate (a self-signed certificate, for example) will cause web services communication to fail with sslhandshakeException. If you opt to use an untrusted certificate, you need to import it into the Java keystore. The general import procedure is described below, followed by examples for Linux and Windows.

### Import a certificate

 Copy the default keystore \$JDK HOME/lib/security/cacerts as \$JDK HOME/lib/security/jssecacerts.

- 2. This will leave the original cacerts file available as a backup. JSSE will use the jssecacerts file, if present, instead of cacerts. Jssecacerts needs to start as a copy of cacerts, which it overrides rather than extends.
- 3. Import the certificate to the jssecacerts keystore using the following command, replacing variables as noted below:

```
$JDK_HOME/bin/keytool -importcert -file $CERT -alias $ALIAS -keystore $JRE HOME/lib/security/jssecacerts -storepass changeit
```

- Replace \$JDK HOME with your actual JDK home path.
- Replace \$CERT with the path to your certificate the you previously installed to the system.
- Replace \$ALIAS with the preferred alias to be used in the keystore.
- Note that changeit is the default password for Java's cacerts file. Check whether it has been changed on your system.
- 4. When prompted, check the certificate and confirm that is should be trusted. (The prompt to verify and confirm the certificate can be suppressed by adding option -noprompt).

## Windows example:

The following command should be written as a single line. It must be run as Administrator. If the Java paths on your system contain spaces, they must be contained in a pair of double (straight) quotes, as shown.

```
"C:\Program Files\Java\jdk-11.0.1\bin\keytool" -importcert -file
C:\Polarion\bundled\apache\conf\certificate.crt -alias labs.polarion.com -keystore "C
```

## Linux Example (CentOS)

This example following command should be written as a single line:

```
/usr/java/jdk-11.0.1/bin/keytool -importcert -file /etc/pki/tls/certs/cert.pem -alias labs.polarion.com -keystore /usr/java/jdk-11.0.1/lib/security/jssecacerts -stc
```

Depending on your operating system and version, additional command parameters may be necessary.

(See https://www.cloudera.com to learn more.)

### **Keytool Commands**

Here are some potentially useful keytool commands:

```
keytool -list -keystore %JAVA_HOME%\lib\security\jssecacerts -storepass changeit
keytool -delete -alias mykey -keystore %JAVA_HOME%\lib\security\jssecacerts
-storepass changeit
keytool -importcert -help
keytool -help
```

## Configure secure access to the repository

Polarion's Subversion repository is accessible via HTTPS protocol when SSL is configured for Apache. To set up secure access:

- 1. Use a text editor to open the system properties file polarion properties (follow link for the location).
- 2. Locate the repo property. By default the value is set to something like repo=http://localhost:81/repo/ Of http://polarion.yourdomain.com/repo/.
- 3. Change the protocol in the repo property from http to https.
- 4. Make sure the login credentials are correct for HTTPS access.
- Restart the Polarion server.

## **Configure Building**

## Basic build management

# ALM REQ QA PRO

#### Note

Availability in **Polarion Requirements** and **Reviewer**:

This feature is also available in **Polarion Requirements**, but it is hidden in the default configuration. The topic can be shown in Navigation by changing the Navigation topics configuration. For more information, see: Configure Interface Views.

This feature is read-only in **Polarion Reviewer**.

Scope(s): Project

This chapter discusses configuration and usage of Polarion ALM's basic build management features. It explains how to quickly get started with several common project build types:

- Simple Maven 2 Project,
- Maven 2 Multiproject,
- Ant Build.
- Build with Makefile,
- External Executable, Batch or Shell Script,
- Continuous Integration,

Additional information for more complex and custom building with Polarion are covered in: Advance Build Management.

Administrators can **Disable builds** or **Calculation Jobs** for security reasons by adding the following properties to the polarion properties file:

- To Disable Builds com.siemens.polarion.security.executingBuilds.enabled=false
- To Disable Calculation Jobs com.siemens.polarion.security.executingCalculations.enabled=false

## **Quick start: Simple Maven 2 project**

Building of a simple Maven 2 project that is structured according to <a href="http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html">http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html</a> is quite simple. Polarion will auto-recognize the Maven 2 project and show it in the Create Build wizard in the Builds topic (in project information scope) either under its name (i.e. <name> from pom.xml) or artifact ID (<artifactId>) if no name is set. No configuration is necessary.

A default build definition called \_default is sufficient for most builds. When you click **Finish** in the **Create Build** wizard, the build will start.

## Launch a build for a simple Maven 2 project

- 1. Open the project you want to build.
- 2. Select the **Builds** topic in the **Navigation** pane.
- 3. In the **Builds** panel of the Content Pane, click **Create Build** to launch the **Create Build** wizard dialog box.

As mentioned above, you should be able to accept the defaults. If you want build reports to be updated after the build, check **Update reports together with the build**.

## Quick start: Maven 2 multiproject

There is essentially no difference between a simple Maven 2 project and multiproject when it comes to building with Polarion.

The only thing which must be assured is that every module in a Maven 2 multiproject is recognized by Polarion ALM.

## Quick start: Ant build

As with Maven projects, Ant builds are also auto-recognized by Polarion - it looks for the file build.xml. The Polarion build artifact's name is either the value of the name attribute of the Ant project, or the simple string ant.

If there are multiple Ant builds in one project, additional steps must be performed to distinguish between them (because Polarion will assign each build artifact the same artifact ID). Suppose the project's structure looks like this:

You can make Polarion recognize the different builds by modifying the

file.polarion/builder/artifacts.xml (for the project, not the global scope):

```
<artifacts auto-recognition="false"> <!-- value "false" is important -->
    <artifact>
        <type>ant</type>
        <location>trunk</location>
        <groupId>com.example.project</groupId>
        <artifactId>normal-build</artifactId>
        <label>Normal Build of Example Project</label>
    </artifact>
    <artifact>
        <type>ant</type>
        <location>trunk</location>
        <antFile>scripts/release build.xml</antFile>
        <groupId>com.example.project</groupId>
        <artifactId>release-build</artifactId>
        <label>Release Build of Example Project</label>
    </artifact>
</artifacts>
```

Setting the auto-recognition attribute to false disables auto-recognition for the entire project (since it is not needed in this case). Because the second Ant build file is not named build.xml and is not located directly under the place specified in <location> (i.e., the called artifact's base location) it is necessary to specify the path to the second Ant build file in <antFile>. Base location serves two purposes:

- 1. All locations are relative to it, and...
- 2. Everything under it (unless resources are used) is downloaded from repository to local disk before the build.

If base location is not explicitly specified, then it is the project's location (or trunk folder if the build artifact was auto-recognized and such a folder exists).

Unless Maven builds, Ant builds (and all other types) do not deploy anything by default to anywhere. Only basic information is deployed to BIR. To be able to deploy more things, a build descriptor must be used. The default build descriptor can be extended with this (contents of the project-level repository file .polarion/builder/descriptors.xml):

```
<descriptors>
    <descriptor>
            <copy dir="dist" includes="*.jar" todir="results" label="jar files"/>
        </deploy>
    </descriptor>
</descriptors>
```

This will deploy all jars which appear in the dist folder to BIR folder results with label jar files. The name, and a link to this deployment, will be visible in the build's detail.

## **Note**

The <descriptor> element could have a name attribute with the name of the descriptor. There could be descriptors with different names for different types of builds. See the advanced topic on build descriptors for more information.

## **Quick start: Build with Makefile**

Suppose that the project structure is like this:

```
trunk
|--- build
|--- build.sh
|--- src
```

Let's also suppose that developers execute builds by going into the build directory and executing make all.

Since it is not possible to call external scripts directly, a simple shell script must be created in the build folder namedbuild.sh:

```
#!/bin/sh
make "$@"
```

Next, the following snippet needs to be added to the project-scope

.polarion/builder/artifacts.xml file:

Now when a build is executed, **trunk** is downloaded to local disk (the location parameter) and make all is executed (by invoking "executable" with correct "arguments") from the build directory (the workdir parameter).

# Quick start: External exe, batch or shell script

This is essentially the same as previous section about Makefile builds. Executables, batch files and shell scripts which are not stored in the repository must be called by some special script which is stored in the repository.

# **Quick start: Continuous integration**

Polarion has two key features for continuous integration:

- Scheduled builds
- Integration builds

Scheduled builds are those builds which run at regular intervals without manual intervention. Integration builds are builds in which only updates of source code committed since the last build are downloaded from repository, and intermediate results of the last build (like compiled classes) are still available.

The following is an example of an integration build that runs every hour. You would define such a build in the jobs configuration file /.polarion/jobs/schedule.xml. (See Configure the Scheduler for more information on this configuration file and working with scheduled jobs.)

```
<job name="Integration Build of Example Project" id="master.build" cronExpression="0</pre>
scope="project:ExampleProject">
    <buildArtifact>com.example.project:normal-build/buildArtifact>
    <buildDescriptorName>_default/buildDescriptorName>
    <localDeploymentSpaceName> default</localDeploymentSpaceName>
</job>
```

# **Configure Work Item Reports**

You can include some information about Work Items in build results by configuring Work Item Reports for builds. The <workitem-report> enables listing of some Work Items in the build preview and notification. A typical usage is to list the Work Items resolved since the previous build. You can configure several Work Item reports, each corresponding to one <workitem-report> element in the configuration. Each report will show as a section in the build preview and notification (below the section "Build Results").

The Work Item reports configuration for builds is available for both repository and for each project.

#### Access the Work Item reports configuration for builds

- 1. Log in with administrator permissions for the scope you want to configure (repository or project).
- 2. If your login does not already take you into Administration from a previous session, enter Administration by clicking the Administration link in the tool view of Navigation.
- 3. In Navigation, expand Building and select Work Item Reports. The Work Item Reports administration page loads in your browser.

## Edit the configuration file:

This configuration is performed by editing the workitem reports.xml configuration file. You can either edit the file in the text editor provided on the administration page, or download it, edit locally, and upload the edited file back to the portal using the controls provided on the administration page.

The configuration file contains extensive comments and an example of a typical configuration in comments. Please refer to the embedded comments for documentation about the available elements and attributes.

# **Advanced Build management**

# Complex and custom building

# ALM REQ QA PRO

This chapter covers Polarion build management configuration and customization in more depth for those who have more complex building setups than those described in Getting Started with Build Management.

#### Note

# Availability in Polarion Requirements and Reviewer:

This feature is also available in **Polarion Requirements**, but it is hidden in the default configuration.

The topic can be shown in Navigation by changing the Navigation topics configuration. For more information, see: Configure Interface Views.

This feature is read-only for users of **Polarion Reviewer**.

# Scope(s): Project

#### **Build Artifacts**

Build artifacts are the smallest buildable entity. They are comprised of resources (usually source code and support files). Multiple build artifacts may share the same resources.

One project may contain zero, one, or more build artifacts. Their definition is stored in the artifacts.xml configuration file.

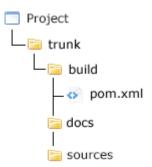
# Access the artifacts.xml configuration file

- 1. Log in with administrative rights for the scope you want to configure (repository or project).
- 2. Enter Administration.
- 3. Open the scope you want to configure (global/repository, project group, or project. (See: Access Projects).
- 4. In the **Navigation** panes, select the **Building** topic.
- 5. Use the appropriate link in the **Configuration** section to download the configuration file to your local system.
- 6. After editing the file, use the controls in the **Upload New [Global][Project] Configuration** section to upload the modified file back to the Polarion repository.

The basic format of the configuration file can be found in the Administration reference topic: Build configuration file format. Please refer to it as necessary when reading this section.

#### Local deployment space

Local deployment space is a place where a build artifact is downloaded before building. Let's consider an example with the following source repository structure:



Assume that pom.xml is written as if sources were in a folder named src in its directory. This definition can be used:

```
<artifact>
   <type>maven2</type>
   <location>trunk
   <resources>
       <resource source="sources" target="build/src" recurse="true"/>
       <resource source="build/pom.xml" target="build"/>
   </resources>
</artifact>
```

Each artifact is identified by its group id (the same convention used with Java package names - e.g. com.polarion.sample) and artifact id a hyphen-delimited name, e.g. sample-demo ). These IDs are usually based on the artifact itself, or are specified in the above configuration. Artifacts also have their label, version and list of source directories (for source-based calculations).

# **Supported Build Artifacts**

This section describes the types of build artifacts supported in Polarion build management. Currently supported types are:

- Maven 2
- Ant
- Shell (batch/executable invocation)

#### Maven 2

```
Type name: maven2.
```

Additional elements:

```
<!-- default is =pom.xml= -->
    <pom>PROJECT_MODEL_FILE_LOCATION_RELATIVE_TO_BASE_LOCATION</pom>
```

Label and source directories are determined from POM file. Note that if POM file is of type pom and it defines modules, then those modules must be build artifacts themselves.

## **Note**

Instead of embedded Maven, you can use external Maven, either the one bundled with Polarion distributions, or any other installation you have. This is not enabled by default during upgrades (only for new installations), but is strongly encouraged to do by placing following into the global /.polarion/builder/build.properties file:

```
polarion.build.maven.executor=maven2
```

#### **Ant**

Type name: ant.

Additional elements:

```
<!-- default is =build.xml= -->
   <antFile>PROJECT FILE LOCATION RELATIVE TO BASE LOCATION</antFile>
   <!-- default is derived from project's location -->
    <groupId>GROUP ID</groupId>
   <!-- default is "ant" -->
    <artifactId>ARTIFACT ID</artifactId>
   <!-- may be omitted -->
    <sourceDirectories>
        <sourceDirectory>
            SOURCE DIRECTORY LOCATION RELATIVE TO BASE LOCATION
        </sourceDirectory>
        ... <sourceDirectory> ...
   <sourceDirectories>
    <!-- default is based on artifact id -->
       <label>LABEL</label>
        <!-- default is 1.0.0 -->
    <version>VERSION</version>
```

#### Shell

Batch/executable invocation

Type name: shell.

# Additional elements

```
<executable>EXECUTABLE LOCATION RELATIVE TO BASE LOCATION/executable>
  <!-- may be omitted ->
  <arguments>ARGUMENT1 ARGUMENT2 ...</arguments>
  <!-- default is executable's directory -->
   <workdir>WORKING DIRECTORY RELATIVE TO BASE LOCATION</workdir>
   <!-- default is derived from project's location -->
  <groupId>GROUP ID
  <!-- default is "ant" -->
  <artifactId>ARTIFACT ID</artifactId>
  <!-- may be omitted -->
   <sourceDirectories>
      <sourceDirectory>
        SOURCE_DIRECTORY_LOCATION_RELATIVE_TO_BASE_LOCATION
     </sourceDirectory>
      ... <sourceDirectory> ...
   <sourceDirectories>
```

```
<!-- default is based on artifact id --> <label>LABEL</label> <!-- default is 1.0.0 --> <version>VERSION</version>
```

# **Build Descriptors (definitions)**

#### **Note**

The term **build definition** may appear in the portal user interface. It refers to **build descriptor** as described in this topic.

Descriptors are configured in the file <code>.polarion/builder/descriptors.xml</code>. You can configure build descriptors in both the global scope, and per project. These are merged if necessary. Here is an example of a build descriptor:

```
<?xml version="1.0" encoding="UTF-8"?>
<descriptors xmlns="http://polarion.com/schema/Builder/Descriptors"</pre>
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 xsi:schemalocation="http://polarion.com/schema/Builder/Descriptors">
 <!-- no name and " default" are equal -->
  <descriptor name="NAME">
   <!-- may be omitted -->
   <artifacts>
      <artifact>
       <!-- if id ends with "*" then the given prefix is searched for;
default value is "*" -->
        <groupId>GROUP ID</groupId>
       <!-- if id ends with "*" then the given prefix is searched for;
default value is "*" -->
        <artifactId>ARTIFACT ID</artifactId>
      </artifact>
      ... <artifact> ...
    </artifacts>
   <!-- may be omitted -->
    <build>
      cproperties>
        <PROPERTY NAME>PROPERTY VALUE
      </properties>
    </build>
    <!-- may be omitted -->
    <deploy>
       <!-- includes and excludes may be omitted,
default value for defaultexcludes is true -->
       <copy|move todir="TARGET DIRECTORY" label="LABEL FOR BUILD REPORT"</pre>
         dir="SOURCE DIRECTORY" includes="INCLUDES PATTERN"
         excludes="EXCLUDES PATTERN" defaultexcludes=
"USE DEFAULT EXCLUDES TRUE OR FALSE">
       ... <copy|move> ...
       <item todir="TARGET_DIRECTORY" label="LABEL_FOR_BUILD_REPORT"</pre>
name="ITEM NAME" />
       ... <item> ...
   </deploy>
```

```
</descriptor>
... <descriptor> ...
</descriptors>
```

# Warning

Usage of <move> or move in <deploy> should be used with caution. It can fail depending on operating system, file system and whether BIR is on the same file system/volume/disk as the data folder. It is also not possible to combine more deployments whose sources are on the same path. Generally, the **move** feature should be used with care and only if necessary due to storage space or performance reasons.

Descriptors are attached to all build artifacts which match a given group and artifact IDs (or all if <artifacts> is omitted). There could be two different descriptors with same name attached to different artifacts.

There is always one default build descriptor named \_default which simply runs a build and deploys it to BIR (linked from the build report) data produced by Maven 2. This default descriptor can be overridden by a descriptor with the same name (or nameless).

In <deploy> tags includes, excludes and default excludes are the same as in Ant (see http://ant.apache.org/manual/CoreTypes/fileset.html for more information).

There is only one deploy item supported for build artifacts of type maven2: default which will deploy all files produced by a Maven 2 build (stored in target directory).

# **Artifacts vs. Descriptors**

It could be said that a build artifact describes what is built, and build descriptor (definition) describes how it is built. Reality is a bit more complicated than this, because build artifacts also define how the build should go. So the more correct view is that a build artifact says what is built (type, identification, repository resources), and what tools are used and their basic configuration (both defined by type-dependent files), while a build descriptor says how these tools are configured (by build properties) and what are the build results (i.e. BIR deployment).

# **Build properties**

Build properties can be defined in (latter overrides former):

- system configuration (in polarion properties configuration file)
- global repository file /.polarion/builder/build.properties.
- project-level repository file .polarion/builder/build.properties.
- properties defined by build descriptor.
- properties defined in Scheduler configuration.

## **Built-in properties**

The following properties are set by the system before a build and cannot be changed:

- polarion.files.repository: BIR location top level folder)
- polarion.build.id: the build ID (e.g. 20070718-1111).

## Standard properties

- polarion.build.shared.config.dir: name of a repository folder relative to the project's location or global root, which will be locally deployed alongside each artifact. Defaults to .polarion/maven. Global is used only if a project scope configuration is missing.
- polarion.build.shared.config.dir.deploy: target name of repository folder (relative to build artifact's local deployment directory; defaults to .config).
- polarion.build.single.artifact.deployment: use if there is a problem with paths being too long for Windows. Setting it to true will deploy to shorter path. However this can be used only with build artifacts which do not contain other artifacts not for Maven multiprojects, for example. Do not use it for calculations under any circumstances.
- polarion.build.shared.local.repository: each build has its own local Maven repository located inside the associated job's working directory, but by setting this property to true builds can use one shared one.

# Warning

Please note that setting this property to true may cause failed builds or calculations, because local Maven repository is not safe for concurrent modifications, and calculations use it by default.

## **Build execution properties**

- polarion.build.maven.executor: specifies which underlying Maven is executed: embedded
  or external installation (default is the Maven installation bundled with Polarion distributions,
  called maven2).
- polarion.build.maven.location.NAME: named location of an external Maven installation (Maven home)
- polarion.build.maven.extra.jvm.options: extra space-separated JVM options (like -xmx) passed to underlying Maven. (Ignored by embedded Maven.)
- polarion.build.maven.extra.options: extra space-separated Maven options passed to underlying Maven. (Ignored by embedded Maven.)
- polarion.build.maven.non.recursive: Maven option --non-recursive (for true), default is nothing.
- polarion.build.maven.offline: Maven option --offline (for true), default is nothing.
- polarion.build.maven.no.plugin.registry: Maven option --no-plugin-registry (for true), default is nothing.

- polarion.build.maven.update.snapshots: Maven option --update-snapshots (for true), default is nothing.
- polarion.build.maven.reactor: Maven option --reactor (for true), default is nothing.
- polarion.build.maven.checksum.policy: Maven options --strict-checksums (for fail) and --lax-checksums (for warn), default is nothing.
- polarion.build.maven.check.plugin.updates: Maven options --check-plugin-updates (for true) and --no-plugin-updates (for false), default is nothing.
- polarion.build.maven.active.profiles: Maven option --activate-profiles. Comma-separated list of profiles to activate.
- polarion.build.maven.inactive.profiles: comma-separated list of profiles to deactivate (ignored by external Maven).
- polarion.build.maven.reactor.failure.behaviour: Maven options --fail-at-end (for fail-at-end), --fail-fast (for fail-fast), and --fail-never (for fail-never), default is nothing.
- polarion.build.maven.plugin.registry.location: location of the used Maven plugin registry (path and name).
- polarion.build.maven.plugin.registry.name: name of the used Maven plugin registry (without path). Default is plugin-registry.xml.
- polarion.build.maven.settings.location: location of the used Maven settings (path and name).
- polarion.build.maven.settings.name: name of the used Maven settings (without path). Default is settings.xml.

# Maven 2-specific properties

- polarion.build.goals: space-separated list of Maven goals/phases to run (defaults to deploy).
- polarion.build.default.deploy.repository.id: default deployment repository ID (usually set in system configuration).
- polarion.build.default.deploy.repository.url: default deployment repository URL (usually set in system configuration).

The default deployment repository is used if the POM file does not define the <distributionManagement> section.

All build properties are reflected as properties accessible by standard ways in Maven (including properties controlling behavior of various plugins like compiler, etc.)

# **Ant-specific properties**

polarion.build.ant.targets: space-separated or comma-separated list of Ant targets to run.
 Defaults to the default specified in the project configuration file.

polarion.build.ant.taskdef.NAME.ATTRIBUTE: value of Ant <taskdef>'s attribute with a given name. NAME is just an arbitrary name used for pairing attributes of one <taskdef>.

# For example:

```
polarion.build.ant.taskdef.1.resource=net/sf/antcontrib/antlib.xml
polarion.build.ant.taskdef.1.classpath=ant-contrib-1.0b3.jar
```

# will produce:

```
<taskdef classpath="ant-contrib-1.0b3.jar" resource="net/sf/antcontrib/antlib.xml"></taskdef>
```

All build properties are reflected as properties accessible by standard ways in Ant.

# **Shell-specific properties**

polarion.build.exec.args: additional arguments to be used when executing.

All build properties are reflected as environment variables.

# Working directory

There are several important directories:

- Working directory of the job related to the build (its location is visible in build's log).
- Base location (in local deployment space = after the download from repository).
- Base directory of Maven 2 (accessible by Maven property basedir).
- Current working directory (called "current user directory" or "cwd" by various platforms).

Base location is either somewhere under the job's working directory or inside the shared local deployment space (if it is an integration build).

Base directory of Maven 2 is always the directory where pom.xml was downloaded (or the build file for Ant builds, or executable for shell builds - Maven is used internally for execution of those). It is under the base location.

If embedded Maven is used, then the current working directory is always the same as the working directory of Polarion (usually directory from which Polarion was run). Otherwise, if external Maven is used (which is the default), then the current working directory is the same as the base directory of Maven 2.

For shell builds, the current working directory can be changed, regardless of which Maven is used, to any directory relative to base location by specifying the <workdir> element in the build artifacts definition.

# **Auto-recognition of Build Artifacts**

Polarion will automatically recognize build artifacts if one of these conditions is satisfied:

clean start (or recognition requested from UI) and there is no artifacts.xml present in the configuration.

• clean start (or recognition requested from UI) and artifacts.xml is present in the configuration with auto-recognition set to true.

Polarion can also auto-recognize artifacts of types maven2 (searches for pom.xml) and ant (searches for build.xml). Files are searched to a maximum of three folders depth, starting from project's location or folder trunk within project's location (if there is one, then it is taken as base location for all artifacts found).

Source directories are automatically determined for maven2, and heuristics is used for ant (heuristics is name-based, not content-based).

If no artifact is auto-recognized, but source directories are found, then special artifact of type <code>auto</code> is used. This type is not intended for building. Rather, it allows source-based reports to work on unknown data.

Because auto-recognition (and detection of changes to build artifacts and related configuration) is a potentially resource intensive operation, there are two configuration options which affect it: polarion.startup.disable.artifacts.auto.recognition and polarion.startup.disable.artifacts.change.detection. Refer to, System Tuning for Advanced Administrators for more information.

# How to build specific types of projects

This section explains how to build Eclipse and Maven 1 projects.

# **Eclipse Projects**

Eclipse projects can be built using the Eclipse PDE Maven Plugin (see <a href="http://mojo.codehaus.org/pde-maven-plugin/">http://mojo.codehaus.org/pde-maven-plugin/</a>) which wraps the Eclipse PDE headless build process automation. For details on usage see <a href="http://mojo.codehaus.org/pde-maven-plugin/usage.html">http://mojo.codehaus.org/pde-maven-plugin/usage.html</a>. In short, one Maven project is created for one Eclipse project (which may contain multiple features and plugins) which must be rigorously structured (all the features in one folder and all the plugins in second folder).

If your repository structure is different than the one required by the Eclipse PDE Maven Plugin, you can adjust build artifact's resources to make the structure upon the build execution. For example consider this repository structure (all under the trunk folder):

```
folderA
|--- plugin1
|--- plugin2
|--- feature1
folderB
|--- plugin3
|--- feature2
pom.xml
```

However Eclipse PDE Maven Plugin requires this:

```
features
|--- feature1
|--- feature2
plugins
|--- plugin1
|--- plugin2
```

```
|--- plugin3
pom.xml
```

That is achievable by changing artifacts.xml to contain:

```
<type>maven2</type>
<location>trunk</location>
  <resources>
    </resources>
</artifact>
```

# **Maven 1 Projects**

Polarion is able to automatically recognize source code from simple Mayen 1 projects whose layout adheres to recommended conventions (with source code in src/main/java. Such build artifacts have their artifact ID (visible in Create New Build wizard) set to auto. The system should be able to compute source code metrics for this artifact, but please note that source code detection is entirely heuristic and may not properly detect source code. It is highly recommended to migrate Mayen 1 projects to Maven 2 and not to rely on this source code detection feature.

There is no direct support for building of Maven 1 projects. However it is possible to call Maven 1 from the command-line as any other tool (see information on build artifacts of type Shell).

## Migrate to Maven 2:

Migration from Maven 1 to Maven 2 is usually very straightforward. Most projects can be converted simply by copying project.xml as pom.xml with a few changes (all these changes apply to elements directly under top-level ct> element):

- **change** <pomVersion>3</pomVersion> to <modelVersion>4.0.0</modelVersion>,
- **change** <id>...</id> **to** <artifactId>...</artifactId>,
- change <currentVersion>...</currentVersion> to <version>...</version>,
- remove <groupId>...</groupId>,
- **change** <package>...</package> to <groupId>...</groupId>,
- add <packaging>jar</packaging>.

For more information see the official Guide to Moving from Maven 1.x to Maven 2.x on Maven's site.

There is also a possibility to convert a Maven 1 project to a Maven 2 project by using the Maven 2 One Plugin. Issue command mvn one:convert in the same directory as project.xml. This will not only convert project definition files, but the plugin also tries to convert any Maven 1 reports to their Maven 2 counterparts. Please note that the conversion is not 100%, but mostly it works. Be sure to update artifacts.xml or issue build artifacts auto-recognition after checking the converted project back in to repository.

# **Branched Artifacts**

There is explicit support for branched artifacts. Artifacts from a branch are handled the same as trunk artifacts. Because Polarion identifies artifacts (from one project) by their group and artifact IDs, and it is fairly common to have branched artifacts with same group and artifact ID but different version, some work-around must be done in order to be able to build trunk and branched artifacts. There are two possibilities:

- Have separate projects for trunk and for every branch.
- Have different group or artifact IDs of trunk and branched artifacts.

# **Maven-specific information**

This section provides some advanced topics related to building with Maven 2.

# Maven profiles

It is sometimes required to distinguish between running the builds on a developer's machine and running them inside Polarion. For this, Maven's concept of profiles can be used (more information is available athttp://maven.apache.org/guides/introduction/introduction-to-profiles.html). There are several ways to define profiles, but here is the use of global profiles defined in C:/Polarion/maven/settings.xml. Just add new profiles to the ones already there):

If developer runs this in standalone Maven with argument -P development then the build will have a property somePathProperty pointing to a folder on developer's machine. If it is run inside Polarion, and build property polarion.build.maven.active.profiles is set to polarion, then the somePathProperty property will point to a folder on the Polarion server.

#### Maven proxy

To minimize connections to the central Maven repository (by other means than by using potentially dangerous polarion.build.shared.local.repository), it is recommended to use Maven proxy (see http://maven-proxy.codehaus.org/). If you have trouble installing the proxy, check http://codeforfun.wordpress.com/2006/05/24/maven-proxy/ and http://codeforfun.wordpress.com/2006/09/11/maven-proxy-details/ for information. Once you have the proxy installed, just edit c:/Polarion/maven/settings.xml and add a new <mirror> section. For example:

```
<settings>
  <localRepository>C:/Polarion/data/maven-repo</localRepository>
  <mirrors>
    <mirror>
```

```
<mirrorOf>central</mirrorOf>
      <id>central-mirror</id>
      <name>Mirror of Central Repo</name>
      <url>http://PROXY HOST:PROXY PORT/PROXY PREFIX/repository</url>
    </mirror>
 </mirrors>
  cprofiles>
    cprofile>
      <id>polarion-bundled</id>
      <repositories>
        <repository>
          <id>polarion-bundled</id>
          <url>file:///C:/Polarion/maven/repository</url>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>polarion-bundled</id>
          <url>file:///C:/Polarion/maven/repository</url>
        </pluginRepository>
      </pluginRepositories>
    </profile>
    cprofile>
      <id>polarion-shared</id>
      <repositories>
        <repository>
          <id>polarion-shared</id>
          <url>file:///C:/Polarion/data/shared-maven-repo</url>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>polarion-shared</id>
          <url>file:///C:/Polarion/data/shared-maven-repo</url>
        </pluginRepository>
      </pluginRepositories>
    </profile>
 </profiles>
  <activeProfiles>
    <activeProfile>polarion-bundled</activeProfile>
    <activeProfile>polarion-shared</activeProfile>
  </activeProfiles>
</settings>
```

For more information about mirrors see <a href="http://maven.apache.org/guides/mini/guide-mirror-settings.html">http://maven.apache.org/guides/mini/guide-mirror-settings.html</a>.

#### Maven repositories

#### One local and two remote repositories are configured:

• Local repository at c:/Polarion/data/maven-repo is used by calculations and can be emptied at will. (Builds each have their own located in their job's working directory. Build property polarion.build.shared.local.repository can be set to true to have this for builds as well, but be warned that it may have undesirable side effects.)

- "Remote" repository polarion-bundled (at C:/Polarion/maven/repository) contains some bundled third party plugins and projects (as well as plugins and projects required for Polarion to work) to minimize connections to Maven's central repository.
- "Remote" repository polarion-shared (at C:/Polarion/data/shared-maven-repo) is fully under user's control and is intended for deployment of projects (this is configured by default to be the default deployment repository see build properties polarion.build.default.deploy.repository.id and polarion.build.default.deploy.repository.url). It is also accessible via http://YOUR\_SERVER\_URL/maven2/.

# Comparison with stand-alone Maven

There is no difference (except for additional configuration which can change the behavior) between running stand-alone Maven 2 tool and running a build with external Maven.

Embedded Maven is significantly different because it is a development version in a frozen state, and is carefully balanced to work with bundled plugins. Using it for builds is discouraged.

#### Under the hood

Maven 2 is a core piece of the Polarion build management system. All builds and calculations are routed through it. Embedded Maven is version 2.1, bundled external Maven is version 2.0 (installed at c:/Polarion/maven/distribution).

If something fails and it is not clear why, then it is recommended to go to the working directory of the build's job, find pom.xml (even if the build was not a Maven build) and see what is inside that could cause something to go wrong.

Plugin registry is at C:/Polarion/maven/plugin-registry.xml, settings are at C:/Polarion/maven/settings.xml. Both files can be freely changed, or other files can be used (see build properties polarion.build.maven.settings.\* and polarion.build.maven.plugin.registry.\*).

# Import Test Case run during Builds

When tests are executed during a standard Maven build, they can be automatically imported as a Test Run when properties in <code>.polarion\builder\build.properties</code> are set. It is possible to fill some Test Run fields from an xUnit file name using regular expressions (regex).

For information, see: Test Management Reference

## **Scheduled Builds**

Builds are scheduled like this (in global .polarion/jobs/schedule.xml):

```
</properties>
</iob>
```

Build is executed with access rights of the system user (defined in polarion properties in the login property). So make sure that this user has repository access to all sources.

# **Integration Builds**

Integration builds are builds which are deployed to one shared local deployment space. This results in a faster deployment phase, because only changes are downloaded from the repository, and the build phase is faster because results of previous build are still in place. Please note that calculations share the same shared local deployment space. Integration builds which must wait for others have status WAITING.

Integration builds are usually stored in data directory in path workspace/polarion-data/deploymentSpaces/ default.

# **Configure Repositories**

# Configure Repositories overview

Although Polarion has its own integrated Subversion repository, which stores Polarion artifacts like Work Items, wiki pages, etc. as well as other artifacts of users, some organizations adopting Polarion want to have Polarion work with multiple repositories. These may either be existing repositories external to Polarion ALM's repository, or new repositories established to work with Polarion ALM.

This chapter provides basic information about multi-repository support and points you to the information you need to set up Polarion to work with repositories in addition to the default one set up during Polarion ALM installation.

There are two Polarion features which can deal with multiple repositories, but they are fundamentally different. You need to understand the basics of each feature before deciding which approach to multiple repositories best meets your needs.

- External Repositories: this feature gives you the ability to link Polarion ALM artifacts stored in the integrated repository with changes (revisions) in source code stored on one or more external SVN or Git repositories. After installation, you can configure Polarion ALM to use one or more external repositories in addition to the SVN repository installed and/or configured during installation of Polarion ALM.
- The Clustering feature enables you to run Polarion ALM on multiple servers, either physical, virtual, or a combination. The topography can be set up to host multiple servers running on separate machines each with its own Polarion ALM repository, and/or multiple machines all accessing a single Polarion repository. (Polarion ALM servers on any node can optionally be configured to access external repositories, as described above.)

Special installation and configuration procedures beyond the scope of Help documentation are required to set up a clustered multi-server environment. These are covered fully in the *Polarion* Enterprise Setup Guide (PDF), available in the Polarion section of SIEMENS' Doc Center.

# **Configure external repositories (GIT, SVN)**

# **ALM**

By default, Polarion uses the Subversion (SVN) repository that was configured by the installation program (Windows) or script (Linux). This "Default" repository is used for all projects by default. The assumption is that development code bases are managed in this same SVN repository. However, if you have code bases managed in another repository, you can configure your Polarion project to enable linking of Polarion Work Items (stored in the default repository) to revisions in the external repository.

Polarion supports revision links with external **Subversion** or **Git** repositories.

When your Polarion project is configured, users can link an external SVN or Git revision to a Polarion Work Item, either automatically by including the Work Item ID in the commit comment, or manually in the **Linked Work Items** table of a Work Item's detail. The user can select the external repository in the drop-down list, and use the **Revision Picker** dialog to select the revision, just as they can when linking to revisions in the default SVN repository. Revisions from an external repository appear in the same **Linked Revisions** table as revisions from Polarion's own SVN repository in the Work Item editor. They are differentiated visually by an icon.

# **Configure the Project**

The **Repositories** topic in project **Administration** enables you to specify one or more external repository(ies) so that revisions there can be linked to Polarion Work Items in the project. It is possible to configure more than one external repository, and external repositories can be of either supported type (SVN or Git).

# Configure an external repository:

- 1. Log in with administrator permissions for the project you want to configure to use an external repository, and open the project.
  - Enter Administration if your login did not already take you into it.
- 2. In Navigation, select  **Repositories** → **■Configuration**
- 3. On the **Configuration** page, if there are no repositories configured yet, the **Add New Repository** button is present. Click it to begin configuring a new external repository.
  - If one or more repositories are already configured, the **Add below** button appears in the **Actions** column of each repository. Click this button on any row to begin configuring a new external repository.
- 4. In the **Provider** list, select the type repository you want to use with this project. When you select repository type, a set of fields specific to the selected type appears below the list control.
- 5. Fill in the fields with data specific to the repository you want the Polarion project to access, noting the required fields marked with a red asterisk (\*).
- 6. Click the **Save** button to save the configuration settings.

Repeat the above procedure to configure additional external repositories.

# Warning

The Default SVN repository is always used by Polarion to store Work Items and other Polarion data. If you have source code for the project in this repository as well as in an external repository, then after adding the external repository(ies) to the configuration, you must explicitly add the Default repository back to the project configuration. If you don't, then users will not be able to link revisions in the default repository to Work Items.

To do this, begin adding a new repository as described in step 4 above, select **Default** in the **Provider** list, and save the change.

#### **Note**

Values for the user name and password for each repository are automatically encrypted and stored in the User Accounts Vault.

# Tip

You can also adjust the following property in the polarion.properties file:

(It determines how frequently (in milliseconds) external repositories are polled for changes.)

The default value is 5000.

com.polarion.repository.pollingPeriod=5000

#### Warning

When using SSH keys on Linux, make sure that you place them in the following folder: /opt/polarion/data/.ssh

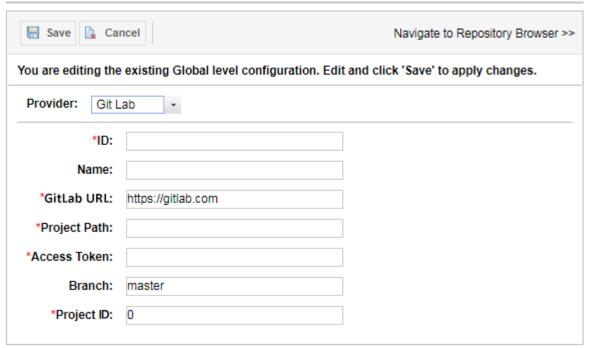
## **Note**

Symlinks are not supported.

# Configure a GitLab repository

- 1. Log on with administrator permissions and enter **Administration** on the project level..
- 2. In Navigation, select **■Repositories → ■Configuration**.
- Click **Add New Repository** or **Add below** if a repository already exists.

# **Repository Configuration**



- 4. Select GitLab from the Provider list.
- 5. Enter a unique **ID**. (The identifier for this repository in Polarion.)
- 6. (Optional) Enter a **Name**. (The Name to display in Polarion for the repository.)
- 7. Enter the GitLab URL.
- 8. Enter a Project Path. (Typically: owner of the project/project slug.)
- 9. Enter the one time Access Token.

To get the access token in GitLab:

a. Select **Settings** under your GitLab profile.



- b. Click Access Tokens on the left.
- c. Enter a Name for your access token.
- d. Tick the api box under Scopes, scroll down and click Create personal access token.
- e. Copy the token that appears under **Your New Personal Access Token** and paste it into the **Access Token** field in Polarion.

10. (Optional) Enter a **Branch**. (The branch that the Connector gets revision data from. Typically the master branch.)

If left blank, the repository does not link any revisions but can still be used as a GitLab repository definition by the GitLab Form Extension.

- 11. Enter a **Project ID**. (The unique numerical GitLab **ID** for the **Project**.)
- 12. Click B Save.
- 13. Add a customized version of the following code snippet to the Work Item form layout so that the Merge Requests section appears in Work Items.

```
<extension id="GitLabExtension" label="Merge Requests" branchNameField="[id of the branch</pre>
name's custom field]"/>
```

The label is customizable and if you don't already have a branch name custom field defined for the Work Item type, you'll need to create one.

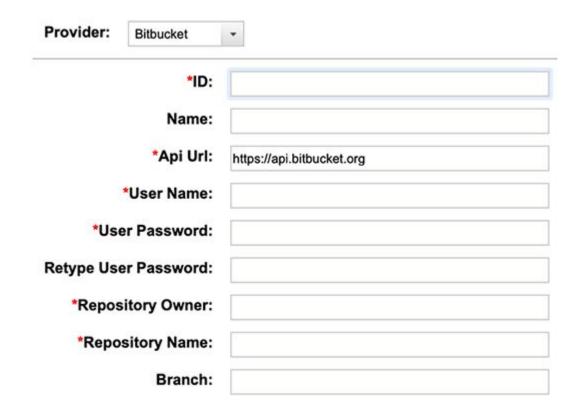
Now you can create and view GitLab merge requests right within a Polarion Work Item.

#### Caution

If there are already revisions indexed by a GitLab repository, then if the branch field of the GitLab repository configuration is cleared, Polarion will not be able to resolve the indexed revisions, and will log errors — every time a Work Item form is opened, for example. To resolve this, an administrator must reindex the Polarion system, which will remove the old revisions from the index.

# Configure a Bitbucket repository

- 1. Log on with administrator permissions and enter **Administration** on the project level.
- In Navigation, select Repositories Configuration.
- Click **Add New Repository** or **Add below** if a repository already exists.



- 4. Select Bitbucket from the Provider list.
- 5. Enter a unique **ID**. (The identifier for this repository in Polarion.)
- 6. (Optional) Enter a **Name**. (The Name to display in Polarion for the repository.)
- 7. Enter the **Api Url**. The API URL to connect to. (https://api.bitbucket.org)
- 8. Enter a **User Name**. (Valid user name of the Bitbucket repository user that will be used for polling.)
- 9. Enter the **User Password**. (Password of the user specified in **User Name**.)
- 10. Enter the **Repository Owner**. (Owner of the repository you want to connect to.)
- 11. Enter the **Repository Name**. (Name of the repository you want to connect to.)
- 12. (Optional) Enter a **Branch**. The branch that you want to index. (Typically, the master branch).
- 13. Click 🖥 Save.

# **Configure Resource Traceability**

## **Resource Traceability configuration checklist**

Resource Traceability gives Polarion users the power to link source code assets semantically. (Link Work Items directly to the source code function, method or variable that implements them.)

This gives managers and auditors a traceable record of all implemented changes and developers a link to who and what prompted a task, in case further clarification is required.

Developers simply add a custom comment to the source code they're developing and it is automatically parsed to any Work Item (Requirement, Change Request, Improvement, Bug Fix etc.) that the code implements.

Administrators can configure how and when this information is parsed in multiple configurations, using multiple parsers, to satisfy all of their technical and time constraints.

Before using Resource Traceability		
✓	Add Resource Traceability properties. to the polarion.properties file for either a Single Server or Cluster setup.	
<b>✓</b>	Configure Repositories.	
	(Supports SVN, GIT or Polarion's default repository.)	
✓	Add Sa Linked Resources to Work Items.	
	Add <extension id="linkedResources" label="Linked Resources"></extension> to each Work Item type that Resource Traceability will be used with.	
<b>√</b>	Configure resource-traceability.xml.	
	Configure Where, When and How Resource Traceability parses data to Work Items.	
✓	Check logs for errors.	
	For Administrators: Check the new log4j-rt log regularly for parsing results and errors.	
✓	Learn the source code syntax.	
	For Developers: The source code syntax used to link where they implemented a source code change to the Work Item that requested it.	

# Warning

If planning to use Resource Traceability with large or multiple repositories, increase Polarion's memory allocation.

# **Add Resource Traceability properties**

# For a Resource Traceability server embedded inside Polarion - The default configuration:

The Resource Traceability server is set to start by default. Administrators that do not wish to use the Resource Traceability feature can disable it by adding the following property to the polarion properties file.

com.siemens.polarion.rt.startRtServer=false

# Customize the following properties to use a different PostgreSQL instance for storing links:

(By default com.polarion.platform.internal PG is used to fetch database properties.)

```
com.siemens.polarion.rt.db.jdbcUrl
com.siemens.polarion.rt.db.username
com.siemens.polarion.rt.db.password
```

# **Use Resource Traceability in a Cluster**

# Warning

To ensure that a cluster setup installs correctly, the property below was added to the shared polarion.properties file. This property should be removed before setting up a Resource Traceability server.

com.siemens.polarion.rt.startRtServer=false

# **Standalone Resource Traceability server:**

To configure a Cluster or Standalone Resource Traceability installation connected to a Polarion Cluster:

This configuration is recommended to ensure high-availability of the Resource Traceability server.

(If the RT node goes down, it can be quickly restarted without having to restart the Polarion application itself.)

#### Adjust database:

- 1. Shutdown **Polarion** and **PostgreSQL**.
- 2. Go to the [POLARION DATA]/postgres-data folder.
- 3. For Windows or Linux installations, open postgreql.conf, comment out the following properties and uncomment the entry (for the same property) exactly below them:
  - max\_connections,
  - shared buffers,
  - work mem,
  - maintenance\_work\_mem,
  - fsync,
  - synchronous\_commit,
  - full page writes,
  - wal buffers,
  - checkpoint segments,
  - effective\_cache\_size,

- max locks per transaction.
- Restart PostgreSQL

To create a database on a new location on a shared storage, please contact Polarion support.

#### Note

To connect a Resource Traceability Server to an external database, the following should be used:

```
com.siemens.polarion.rt.db.jdbcUrl=jdbc:postgresql:
//<databaseLocation>:5433/polarion
com.siemens.polarion.rt.db.username=<username> (e.g polarion)
com.siemens.polarion.rt.db.password=<password>
```

# Adjust the Resource Traceability server's polarion.properties file.

When connecting the Resource Traceability server to a Polarion Cluster:

- 1. Mount the shared storage to the Resource Traceability node. (Required to share the polarion.properties file.)
- 2. Make a copy of your polarion.properties file for Resource Traceability.
- 3. After making the copy, replace the content with the content below and adjust the properties if needed:

```
com.siemens.polarion.rt.polarionUrl=http://polarion-cluster
com.polarion.application=polarion.rt
#Shared folder between the machines that make up the cluster
#default Linux: com.polarion.shared=/opt/polarion/shared
#default Windows: com.polarion.shared=\\\\<shared services host>\\Polarion
com.polarion.shared=/opt/polarion/shared
TomcatService.ajp13-port=8889
base.url=http://rt-hostname
#Control port and host name for shutdown requests
controlPort=8887
controlHostname=rt-hostname
com.polarion.platform.internalPG=polarion:polarion@localhost:5433
```

#### Note

The com.siemens.polarion.rt.polarion URL should point to the cluster address that goes through the load balancer.

# Adjust the virtual memory settings:

- 1. Adjust the Virtual Memory properties so that they fall into the -xms500m -xmx2g range.
  - (These values will vary depending on the number of external repositories, their size and scheduling.)
  - For Windows: in the [POLARION\_HOME]/polarion.ini

- For Linux: in [POLARION\_HOME]/etc/config.sh
- Restart Polarion

# Adjust the Polarion server:

Adjust the Polarion server to work with the standalone Resource Traceability server.

When connecting a Polarion cluster to a Standalone Resource Traceability Server, add the following properties **to each node**:

```
com.siemens.polarion.rt.startRtServer=false
com.siemens.polarion.rt.url=http://rt-hostname
```

#### Note

com.siemens.polarion.rt.url should point to the base.url of the standalone Resource Traceability server.

(For both cluster and single installations.)

Https setup is done like any other Polarion instance. Certificates must also be imported into the truststores of both the Polarion and Resource Traceability servers.

# **Embedded Resource Traceability Server in Cluster Nodes:**

To ensure a high-availability setup, use the Standalone Resource Traceability Server setup above.

# Warning

To ensure that a cluster setup installs correctly, the property below was added to the shared polarion.properties file. This property should be removed before setting up a Resource Traceability server.

```
com.siemens.polarion.rt.startRtServer=false
```

To correctly configure a Resource Traceability cluster, setup **Reader** and **Writer** nodes.

 Reader Node: Can only return links that are stored in the Resource Traceability database for a specified Work Item.

(There is no limit to the number of **Reader** nodes located in the cluster.)

• **Writer Node**: enables configuration updates, collects information from the repositories and stores data, (files, configurations and links), in the database.

(Only a single **Writer** node is allowed in the cluster.)

# **Note**

**Writer** node settings can be left as is, because a Polarion instance starts the Resource Traceability (RT) server by default as a **Writer** instance.

## **Configure Reader nodes:**

Customize the following Properties to use a different PostgreSQL instance for storing links:

(A database on a different node acts like a separate PostgreSQL instance and the properties below should also be provided on the node or instance pointing to the database.)

**e.g.** com.siemens.polarion.rt.db.jdbcUrl=jdbc:postgresql://someurl (e.g. node2):5433/polarion where the URL points to a different server.

By default com.polarion.platform.internalPG is used to fetch database properties.

```
com.siemens.polarion.rt.db.jdbcUrl
com.siemens.polarion.rt.db.username
com.siemens.polarion.rt.db.password
```

All Reader nodes should be configured to send different "write"-requests to the Writer node.

They should also all be marked as **Reader** nodes by setting the following property to 'false'.

```
com.siemens.polarion.rt.dataCollectingEnabled=false
```

The following property should be linked to the **Writer** node's base URL.

```
com.siemens.polarion.rt.writerNodeUrl=
```

Define the same database properties in the following properties for every **Reader** node.

They should be linked to the Database that is used by the **Writer** node.

This enables the Polarion located on a **Reader** node to send a request to fetch Work Item links for its local RT Server instance along will all other requests (e.g. configuration changes) to the **Writer** node.

```
com.siemens.polarion.rt.db.jdbcUrl
com.siemens.polarion.rt.db.username
com.siemens.polarion.rt.db.password
```

## **Configure repositories**

# Things to consider when configuring a Resource Traceability compatible repository:

- Currently only a single branch per GIT crepositoryConfiguration> is supported. The local repository must be switched to the branch defined in the configuration.
- To collect resources from different branches in GIT, create a clone of the repository in the desired branch and set it up as another repository in External Repository Configuration and the resource-traceability.xml file.
- To use an External Repository, configure the repository before configuring the resource-traceability.xml file.

```
( Administration Repositories Configuration ) See Configure External
Repositories for details.
```

#### Changes to a Repository

When considering change to an existing repository, it is essential to understand the potential impact on resource traceability.

If the **root URL** of the external repository is changed, no recollecting of resources is processed: no old records are removed, and no new ones are processed.

• If the **ID** of the external repository is changed, then old records **are** removed and new ones are processed.

For example, if a user changes the URL of the repository without changing the actual repository (e.g. replaces http:// with https://, or the changed URL points to the same repository with same structure as old one, then there is no problem with not recollecting anything. However, if a user changes the URL of the repository and the new repository has different structure, then the administrator should change the ID of the current repository configuration and also change it in resource-traceability.xml in order for everything to be recollected.

# Configure the Resource-Traceability.xml file

- 1. Select the target Froject.
- 2. Click on on the top left of the navigation bar.
- 4. Customize the XML in the provided field using the parameters below:

# Sample resource-traceability.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<resource-traceability>
   <categories>
     <category id="first category" name="Codebase">
        <repositories>
           <repositoryConfiguration id="default" repository="default">
              <branches>
                  <branch id="Release 1.0"</pre>
path="Demo Projects/elibrary/branches/1.0"/>
                  <branch id="Release 1.1"</pre>
path="Demo Projects/elibrary/branches/1.1"/>
              </branches>
                   <paths>
                      <path dir="java folder"/>
                   </paths>
           </repositoryConfiguration>
        </repositories>
        <schedulers>
            <scheduler cron="0 */60 * * * *"/>
        </schedulers>
        <fileParsers>
            <fileParser extensions=".java" id="javaParser"</pre>
name="com.siemens.polarion.rt.parsers.java">
            </fileParser>
        </fileParsers>
      </category>
   </categories>
</resource-traceability>
```

Use the sample xml file above as a structure guide and customize it using the tags below:

## Warning

If a configuration is deleted, all the links associated with it will also be removed!

#### Note

See the <scheduler cron="0 0/60 \* \* \* \* \*" /> tag below for details on how to stop a parser from running a <RepositoryConfiguration> without deleting existing links.

# <categories>

The wrapper tag used for the <category id="first category" name="Codebase"> tag.

For Resource Traceability, a category lets administrators organize one or more <RepositoryConfiguration> tags by similar criteria (e.g. Codebase, Marketing Material etc.).

The configuration file must contain one <categories> tag with one or multiple <category> tags defined for each project.

The resource-tracability.xml file must contain at least one category but multiple <categories> can be defined for each project.

```
<category id="first category" name="Codebase">
```

The <category> tag lets administrators create multiple configurations, based on different criteria, within the singleresource-traceability.xml file.

If parser and/or scheduler are set on the category level then it will apply to all repositories inside that category, unless explicitly set on the <RepositoryConfiguration> level.

Multiple <repositories> can be configured on this level by creating a <RepositoryConfiguration> for each.

Define a unique system **id** and common **name**.

The system id is used by Polarion to identify the category and the name is what appears in the **Linked Resources** section of a configured Work Item.

## <repositories>

The wrapper tag used for the crepositoryConfiguration> tag.

This tag must be used inside a category. Here you can define one or more <repositoryConfiguration>.

## <repositoryConfiguration>

Where Resource Traceability searches for, and processes data.

One or more <repositoryConfiguration> tag can be defined per <category> tag. (At least one must be defined for each.)

```
<repositoryConfiguration id="default" repository="default">
```

Use this tag to define multiple configurations for a single repository by entering a unique name for each configuration in id= and use the id of the repository defined on the Repositories

Administration page ( Administration Repositories) as the repository=value.

#### <br/> <br/> dranches>

```
<branch id="Version 0.9" path="Demo Projects/elibrary/branches/0.9"/>
```

A duplication of resources for version variants or testing.

The behaviour of this tag differs, depending on the Repository type. (Git or SVN)

- For **SVN**, multiple branches can be defined or none at all.

(If none is specified the default "Trunk" repository is used.)

- **GIT** , requires that only a single branch be defined, and currently does not support multiple branches.

(If no branch is specified for a GIT repository then the "Master" branch will be used by default.)

Branch id= The branch id that appears in the Linked Resources section of a configured Work Item.

Path=The absolute location of the branch starting from the root of the repository. (Can include the Project and/or Project Group.)

## <paths>

The wrapper tag for the <path dir="elibrary/trunk/src/main/java"/> tag.

```
<path dir="elibrary/trunk/src/main/java"/>
```

The file and folder locations processed by the parser.

# Note

One or more <paths> can be defined per Repository.

If branches are used, specify, if needed, the specific path location(s) inside the branches that should be processed by the parser.

If not specified the whole branch directory will be processed.

If branches are not used, set the absolute location of the path starting from the root of the repository, including the "*Project*" and/or "*Project Group*".

#### <fileParser>

```
The wrapper tag used for the <fileParser id="javaParser" name="com.siemens.polarion.rt.parsers.java" extensions=".java"> tag.
```

```
<fileParser id="javaParser" name="com.siemens.polarion.rt.parsers.java"extensions=</pre>
```

The parser (Java, C etc.) used to extract the traced content and populate the links that appear in the Linked Resources section of a configured Work Item.

One or more <fileParser> can be defined per <RepositoryConfiguration> or <category> tag...

Define a unique parser system id, the parser's name, and the file Extension it should search for.

#### **Available Parsers:**

```
For Java - <fileParser id="javaParser" name="com.siemens.polarion.rt.parsers.java"
extensions=".java">
```

For C - <fileParser id="cParser" name="com.siemens.polarion.rt.parsers.c" extensions=".c">

For XML - <fileParser id="xmlParser" name="com.siemens.polarion.rt.parsers.xml" extensions=".xml">

#### <schedulers>

The wrapper tag for the <scheduler cron="0 \*/60 \* \* \* \* \*"/> tag.

One or more <schedulers> can be defined per <category> or <RepositoryConfiguration> tag.

```
<scheduler cron="0 */60 * * * * " />
```

When or if the associated parser is executed.

## Tip

To keep existing links but stop the further processing of new links, remove the <scheduler cron="0 \*/60 \* \* \* \* \*"/> tag from the target <category> or <repositoryConfiguration>.

#### **Note**

Use Cron syntax to define the parsing schedule. See the examples of cron Expressions section for details.

(See Cron Trigger Tutorial to view an online tutorial.)

- 5. Click Save once XML configuration is complete.
- 6. The configured Repositories can be run at any time by clicking the Run Config... button on the Resource Traceability Administration page.

(This is a great way to test a new configuration, or bypass the scheduled wait time.)

# Check logs regularly

Administrators should regularly check the log4j-rt-(time stamp) logs in the Polarion\data\logs\main\directory to stay on top of the following:

- What collections occurred,
- What repositories were collected,
- What errors, if any, occurred.

#### Java and C parser behaviour

If the Java or C parsers encounter something that they cannot process, e.g. incorrect comment syntax, they will skip the offending code, log an error and continue parsing the rest of the file.

The Java Parser is optimized for OpenJDK.

# The C Parser is optimized for the C11 Standard.

Java and C Parser Encoding

Use UTF-8 encoding for the Java and C Parsers.

# XML parser behaviour

If the XML parser encounters something that it cannot process, then it will *stop processing the rest* of the file, and log an error.

# XML Parser Encoding

The XML parser automatically detects encoding tags (for example, <?xml version="1.0" encoding="ISO-8859-1"?> but if no heading exists it will use UTF-8.

(If it does and the encoding is not UTF-8 a parsing error will occur.)

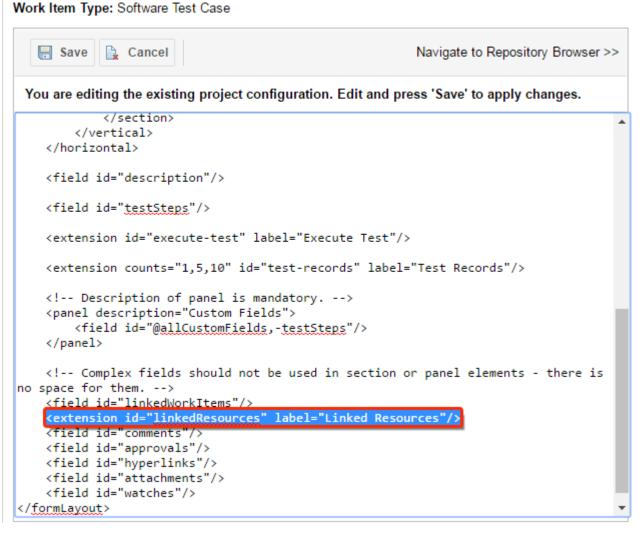
## Add "Linked Resources" to Work Items

- 1. To add Resource Traceability's " Schiked Resources section to selected Work Items.
  - Go to ( **\*Administration** → **\*Work Items** → **\*Form Configuration**.)
- 2. Scroll down to the Form Layouts section.
- 3. Select a Work Item type to add the Resource Traceability feature to and click **Edit**.
- 4. Insert the following line where the new \*Linked Resources section should appear in the Work Item.

```
<extension id="linkedResources" label="Linked Resources"/>
(label="Linked Resources" is the section title that appears in the Work Items.)
```

# Form Layout Configuration

View: Default



- 5. Click B Save.
- 6. Repeat this for any Work Item type that you wish to have Resource Traceability configured for.
- 7. Configured Work Items will now include a \$\frac{\text{\$\congrue}}{\text{\$\congrue}}\$Linked Resources section.

```
Linked Resources
com.polarion.demo.Book.getPublisher() Role: implements
Codebase - default (javaParser)
Release 1 /Demo Projects/elibrary/branches/R1/Java/Book.java - Line 44
Release 2 /Demo Projects/elibrary/branches/R2/Java/Book.java - Line 47
```

## **Note**

This Section always shows information from the "Head" (most recent) revision. (Even in Baseline and Historical Views.)

# **Enable "Linked Resources" READ permissions**

Make sure that the right users have permission to view Schinked Resources in a Work Item.

*Read* rights can be assigned to individual user names **By Permission**, **By Role** or through **Custom Sets**.

# By Permission

- 1. Select the target Project.
- 2. Go to ( **≁Administration** → **≛User Management** → **≛ Permissions Management**.)
- 3. Click the **By Permission** tab.
- 4. Expand the Work Items section and then the Fields section within it.
- 5. Click on Permission to READ field "Linked Resources" and click Edit.
- 6. Make sure that the applicable users have a Check mark  $\boxed{}$  in the box in the **Granted** column beside their username(s).
- 7. Click 🖥 Save.

# By Role

- 1. Select the target Froject.
- 2. Go to ( \*Administration \* \* User Management \* Permissions Management.)
- 3. Click the **By Role** tab.
- 4. Select a Role from the Role drop-down list, expand the **Work Items** section and then the **Fields** section within it.
- 5. Make sure that there's a check mark beside Permission to READ field "Linked Resources" .
- 6. Click 🖥 Save .

#### **Note**

Permissions can also be granted via Work Item Custom Sets.

#### Link a resource

# Resource Traceability: Source code syntax

Developers add a source code comment using the syntax below to customize what information is included with the linked resource.



- Required @wi tells the parsers to include the following line(s) in the Linked Resources section of the specified Work Item(s).
- (Optional).linkrolename (Add how it relates to the Work Item.)
- (Optional) The Project ID name.
- Required The Work Item ID. Can also be the full Work Item URL.

(https://polariondomain.com/polarion/#/project/test/work-item?id=TST-1007&revision=5678 message.)

- (Optional) The revision number. (It will provide a link in Linked Resources directly to the revision that the source code change refers to.)
- (Optional) An additional commented message from the developer that will also appear in the Linked Resources section of the Work Items.

# Linked resource example



1 Link Role	6 Branch
2 Category	7 Revision
3 Repository	8 Message
4 parrow	This reference can be linkable and bring users directly to the code that implemented the Work Item change.
Parser	If administrators set the View Traceable Resource Local URL.  (*Administration ** ** Repository )
6 Element	10 Position

To add a linked resource like the example above (for Java or C parsers), add the following commented code where the change was implemented.

```
/**
 * @wi.implements elibrary/EL-115:164 Returns author of this book
*/
public String getAuthor() {
 return this.author;
}
```

## Multi-line message

There is no limit to the number of commented lines for a developer message.

Just start the message with a "{" and end it with "}".

```
/**
* {@wi.implements elibrary_alm/EL-122 messages can take up
* several lines in the code. Just surround the whole parsing syntax
* with {}
* Anything commented after will not be included.
```

## com.polarion.demo.Library.add(Book) Role: implements

Codebase - default (javaParser)

Release 1.1 /Demo Data Development/elibrary\_alm/branches/1.1/Java/Library.java - Line 34 messages can take up several lines in the code. Just surround the whole parsing syntax with {

## Supported elements

Developers can add the Resource Traceability feature to comments for the following elements:

## For Java: (.java is the supported extension.)

- method
- constructor
- field
- class
- interface
- enum
- static block

For C: (Supported extensions - .c, .h and additional custom extensions depending on how the parser and project preferences are configured.)

- function
- function prototype
- global variable
- struct
- union
- enum

## Where to place the comment

Where the comment should be placed, varies depending on the parser used.

#### For Java and C:

If using a Java or C parser, the Resource Traceability syntax links the element that directly follows it in the source code.

(public String getAuthor() in the example above.)

#### For XML:

For XML, add the comment inside the target element as in the example below.

```
<plugins>
  <plugin>

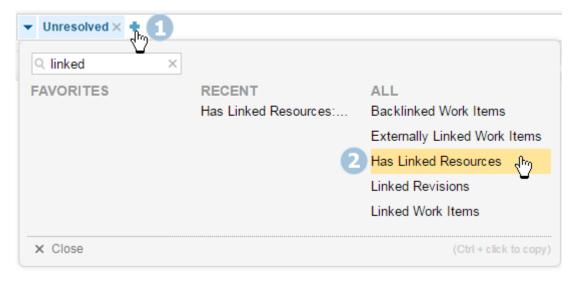
✓ | <groupId><!-- @wi EL-188 -->org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-plugin</artifactId>
    <!--version>2.9</version-->
   <configuration>
      <excludes>
        <exclude>${exclude.tests.pattern}</exclude>
      </excludes>
    </configuration>
  </plugin>
 <plugins>
   <plugin>
  <!-- @wi EL-188 -->
     <groupId>org.apache.maven.plugins</groupId>
     <artifactId>maven-surefire-plugin</artifactId>
     <!--version>2.9</version-->
     <configuration>
       <excludes>
         <exclude>${exclude.tests.pattern}</exclude>
       </excludes>
     </configuration>
   </plugin>
```

The second example will link to the Plugin instead of the groupId.

#### Search for Work Items with Linked Resources

Users can search for Work Items with or without Linked Resources.

#### Work Items with Linked Resources:



#### Work Items without Linked Resources:



# **Configure Connectors**

# **Configure Connectors overview**

While Polarion ALM provides solutions that encompass all aspects of the application life cycle, it may not always be possible or realistic for organizations to adopt Polarion ALM solutions for the full spectrum. For example, your team or department might want to use Polarion ALM for requirements management, but other teams or departments are currently unable or unwilling to adopt the Polarion ALM solution applicable to their function. For example, development teams might want to keep their present tool for task and issue tracking, or testing teams might want to keep on with their current test management and testing tools.

If we identify sufficiently widespread demand for interoperation with Siemens or third party systems or tools in order to help prospective customers remove barriers to adoption of a Polarion solution within their organizations, we may develop *Connectors* for these third-party tools and systems. Connectors are integrated directly into the Polarion ALM platform rather than implemented as extensions. Connectivity and interoperation is configurable directly in platform's Administration interface, in the Connectors topic, and technical support is provided according to the customer's subscription plan.

The list includes connectors for such popular solutions as:

- Siemens Teamcenter
- Atlassian JIRA (version 5.0 or later)

- HP Quality Center or HP ALM (versions 11 and 12)
- IBM DOORS
- MATLAB Simulink
- Microsoft TFS
- Perforce
- Salesforce

Some connectors may require you to download them from the Polarion ALM Extensions portal and install them per bundled instructions before they can be set up in Polarion ALM administration.

## Overview of functionality

Connectors provided either unidirectional or bidirectional synchronization of data between Polarion and a third-party system. The synchronization process generally works in the following way:

- 1. Load all items that match a query and other parameters from Polarion ALM.
- 2. Load all items that match query and other parameters from other system.
- 3. Determine what needs to be done with the items:
  - Create "pairs" of items that have been synchronized before by looking up the connection information stored in folder [POLARION]/data/synchronizer.
  - If the "partner item" is found in the queried items, create a pair to be synchronized.
  - If an item is connected, but the "partner item" was not found, try to load the partner item by ID to create a pair to be synchronized.
  - If loading the item by ID fails, consider the item as deleted.
  - Consider all items that are not connected according to information in [POLARION]/data/synchronizer as new items.
- 4. Create new items, synchronize paired items and propagate deletions according to the connector configuration.

In summary, if an item matches the query on either side (Polarion or third-party system) it will be synchronized.

The following sections in this chapter explain how to configure a connection to third-party systems, define synchronization pairs, map data interchange between Polarion and another system, and use features of the connectors.

## Import an IBM DOORS project

Use the Connector for DOORS to import a complete IBM DOORS project into Polarion. The connector works with all Doors 9.x versions running on Windows. (It will still work if Polarion is run on a Linux machine.)

#### **Note**

#### Limitations:

Before importing, read the Import limitations and recommendations section.

The connector will import the following:

- All modules selected for import as a Live Document,
- All requirements in the modules,
- All links between requirements in the same or different modules,
- All the configured requirement attributes, including tables.
- DXL attributes are also imported and can be mapped to Polarion fields.

Once imported, the connector can be used to continuously import updates made within the DOORS modules.

## Tip

To ensure that the synchronization runs smoothly, make sure that there is a strong network connection between DOORS and Polarion.

1. Click on the on the top left and go to Administration.

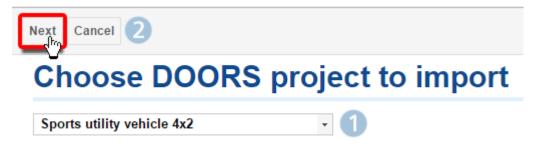
- 2. Open the **DOORS** Client.
- 3. Open the **DXL Interaction Window** in DOORS. (**Tools DXL** )
- 4. Click on the script on the Establish connection to DOORS screen in Polarion and copy (Ctrl **+C**) it to the clipboard.

# Establish connection to DOORS

- Open the DOORS Client.
- 2. Open the DXL Interaction Window (Tools > Edit DXL)
- Press CRTL-C to copy the script below, paste it into the DXL input text box and click Run to execute the script.

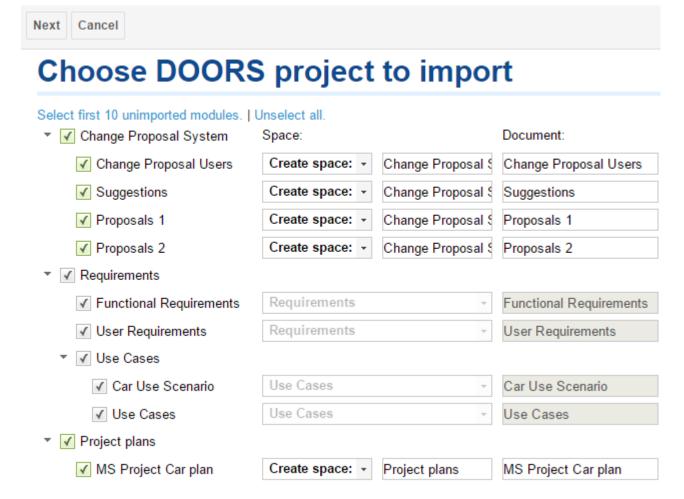
```
oragma runLim,0
OleAutoArgs args = create
OleAutoArgs args2 = create
OleAutoObj http = oleCreateAutoObject("WinHttp.WinHttpRequest.5.1")
 oid sendResult(string commandId, string result) {
clear(args)
      clear(args2)
put(args, "POST")
put(args, "http://ea-latest.labs.polarion.com/polarion/doorscon
0a7b1e4b-11e33294-6b0fbe04/commands/" commandId "/result")
                                                                                            nector/rest/connections/347c1a93-
      OleAutoArgs args2 = create put(args2, result) oleMethod(http, "open", args) oleMethod(http, "send", args2)
Regexp line = regexp '
bool connected = true
vhile(connected) {
put(args, "GET")
put(args, "http://ea-latest.labs.polarion.com/polarion/doorsconnector/rest/connections/347c1a93-
Da7b1e4b-11e33294-8b0fbe04/nextCommand")
      oleMethod(http, "open", args)
oleMethod(http, "send")
      int status
oleGet(http, "status", status)
      connected = status == 200
      oleGet(http, "responseText", response)
print "RESPONSE:" response "\n"
      line response
      string commandId = response[0:end 0]
string resultUrl = "http://ea-latest.labs.polarion.com/polarion/doorsconnector/rest/connections/347c1a93-
 a7b1e4b-11e33294-8b0fbe04/commands/" commandid "/result"
      string command = "pragma runLim,0\nstring resultUrl = \"" re
       sendResult(commandId, eval_(command))
```

- Paste (CTRL +V) the script into the DXL Input text box in DOORS and click Run.
- 6. The DOORS client will connect to the Polarion server and send the required connector data.
- Select the target DOORS project and click Next.



8. Select all DOORS modules to configure and import.

Previously imported modules have a black check mark. These modules will be re-imported but not re-configured.



Click the check box again so that they turn green to re-configure them.

Remove the check mark on any modules you do not wish to re-import.

#### Note

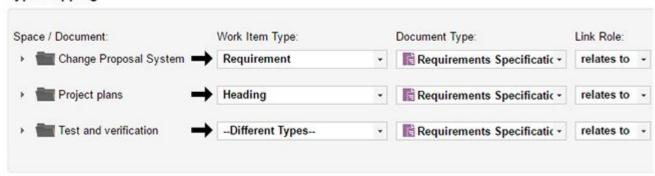
## Maximum number of modules that can be configured per import:

The maximum number of modules that can be configured in a single import is 50. (But there is no limit to the number of pre-configured - black check marked modules - that can be imported). If more than 50 modules need to be configured, simply run the connector multiple times.

9. Configure the Work Item type to be used when importing and configure how the DOORS attributes are mapped to Polarion attributes.

# Choose DOORS project to import

## Type Mapping



10. The **Create missing attributes and values** check box will automatically create all missing custom fields and enumeration values.

## Attribute Mapping



- 11. Click **Start Import** to import all selected modules to Polarion.
- 12. The import status can be checked in the Monitor.

# **DOORS** import limitations and recommendations

- DOORS only uses a single CPU core per thread so you should only run a single Connector session at any given time (to a single Polarion Project), and merge the data once its complete.
- Picture objects are not supported but pictures inserted as OLE objects are.
- If a DOORS module's structure can't be mapped to a document flow, Requirements may not
  appear at the expected location. For example, in DOORS you can create the following structure.

#### **Example:**

## Heading

#### Requirement

But a document flow would recognize it as;

#### Example:

## Heading

## >Requirement

The connector would therefore preserve the parent child relation by re-positioning the requirement

## Requirement

## Heading

Headings that are hierarchically below a Requirement can't be inserted into the document during import.

Example:

## Requirement

## >Heading

- Module properties are not imported, only object properties are.
- DXL attributes will be imported as static data into Polarion so will not act as they do in DOORS.
- The thumbnails generated for imported OLE objects are not always accurate. (Especially when the conversion is executed on Linux.)

## **Note**

Image Magick must be installed and configured to see the thumbnails of imported OLE Objects. See the Include thumbnails for OLE DOORS and RegIF imports section for details.

## **Memory Settings when Importing Large DOORS Modules**

Synchronizing large DOORS modules that contain a lot of OLE objects consumes a lot of memory. If you're planning to import them, assign at least 4 GB to Polarion.

If you see an error like "Expected 5 objects (top-level) from Doors but received 4" in the synchronizer log it's probably because Polarion has run out of memory and you'll need to assign more. (You can confirm this by looking for java.lang.OutofMemoryException entries in the main Polarion log.)

## **Linked Data applications**

## **Linked Data applications overview**

This topic covers how to configure external Linked Data applications to work with Polarion.

Polarion's external Linked Data support lets users link to objects from other tools, like Teamcenter.

#### The Linked Data can be used to:

Create links for Work Items or LiveDocs from objects that reside on external Linked Data enabled tools. (Act as a Linked Data consumer.)

Linked Data consumers can connect to Polarion using the following rootservice URL:

http://polarion-server/polarion/oslc/rootservices

(Polarion provides the OSLC RM and CM domain that enables Linked Data consumers to link to Polarion. Delegated UIs are provided.)

• Link Polarion Work Items *to* objects that reside on external Linked Data enabled tools. (Act as an Linked Data provider.)

(This is usually done by creating or selecting a remote object in a delegated UI provided by the external Linked Data application.)

Administrators can define linking rules in the Linked Data Semantics and Linked Data Mapping configuration sections.

## **Example:**

- 1. A user selects a **Requirement** Work Item that is considered a rm:Requirement according to the Linked Data Mapping configuration.
- 2. They select the **is implemented by** link role, as the backlink for the **implements** role.
- 3. The **implements** role is mapped to cm:implementsRequirement based on the Linked Data Mapping configuration.
- 4. The reverse for that is rm:implementedBy based on the Linked Data Semantics configuration.
- 5. The Linked Data link type rm:implementedBy can be used from rm:Requirement to cm:ChageRequest according to the Linked Data Semantics configuration.
  - (So only dialogs that can be used to select cm: ChangeRequest remote objects are shown.)
- 6. After a user selects a cm:ChangeRequest the available link roles are also filtered so that only those valid for use with rm:Requirement and cm:ChageRequest are displayed.

Once and Linked Data Friend is added, and a Linked Data Association is established, an la icon appears in the Linked Work Items section of a Polarion Work Item.

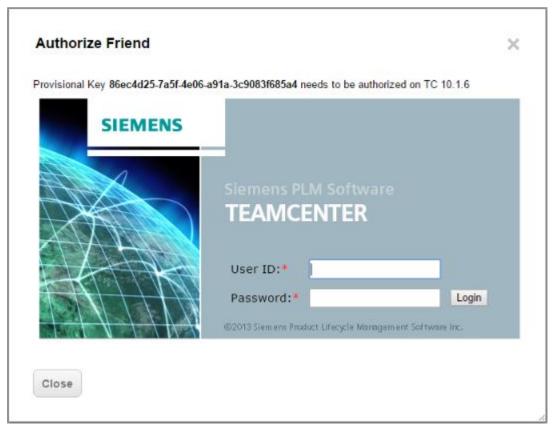
Users can then Link Items to and from external Linked Data applications.

## Add server as a "Friend"

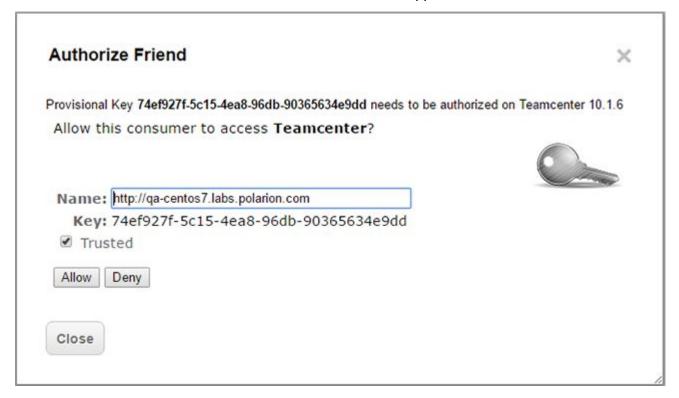
- 1. As an Administrator, click on the top left of the **Navigation** Bar.
- 2. Click Administration.
- 3. Click **Global Administration**.
- 4. Click on the S Linked Data topic on the Navigation Bar.
- 5. Click on the **S** Linked Data Friends sub-topic.
- 6. Click \* Add Friend .
- 7. The dialog below appears.



- 8. Enter the following information and click **NEXT** .
  - **URI**: <OSLC Server URL>/lis/oslc/rootservices
  - Name: Enter a name that will help you to identify your external Linked Data server,
  - OAuth Secret: Enter any random text.
     This OAuth file is secure and only available for authorized users. (The @admin group by default.)
- 9. A pop-up dialog appears with the login screen of the external Linked Data application.



- 10. Log into the external Linked Data application.
- 11. Authorize Polarion as a "Friend" of the external Linked Data application:

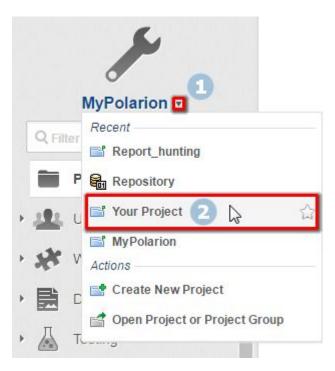


- a. **Name**: Enter the URL of the Polarion server that you want to add as a friend.
- b. Tick the Trusted box.
- c. Click Allow.
- 12. Now it's time to enable the linking of Polarion Work Items with the external Linked Data application's items.

## **Enable item linking**

Once an external Linked Data application and Polarion are configured to communicate with each other, an *association* between the two must be created before being able to link items back and forth.

1. Select a target Project and enter the Project level Administration.



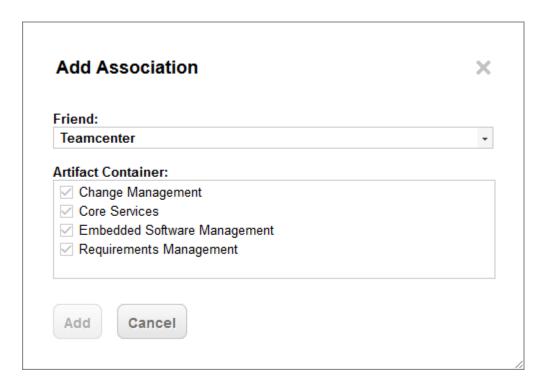
- 2. 🏕 Administration 🕶 🚳 Linked Data 🕶 🚳 Linked Data Associations.)
- 3. Click Add Association and select the external Linked Data server to connect to the selected Polarion Froject.

## **Note**

If connecting to SIEMENS' Teamcenter for the first time of a session, an additional pop-up will open with the Active Workspace login.

Enter your Teamcenter Credentials.

4. Select the external Linked Data services to associate Polarion with.



- Select **Change Management** if you want to link to Teamcenter items specific to Change Management (for Example: PR, ECR, ECN).
- Select Core Services if you want to create generic links to any Teamcenter item type.
- Select Embedded Software Management if you want to link a release in Polarion to Teamcenter.
- Select Requirements Management if you want to link to Teamcenter requirements.

#### Tip

Select all to have access to all the integration functions.

- 5. Click Add.
- 6. An association between the external Linked Data application and Polarion is made.
- 7. Items can now be linked between Polarion and the External Linked Data application.

## **Configure Linked Data mapping**

**Linked Data Mapping** is used to map custom Work Item and link types to semantic types defined in **Linked Data Semantics**.

Linked Data Mapping can be defined on both the *Global* and *Project* levels.

## **Note**

Project level Linked Data Mapping overrides Global Level Linked Data Mapping for the targeted project.

## **Project level**

(Select Project → F Administration → ©Linked Data → © Linked Data Mapping)

#### Global level

(Select Project → FAdministration → FGlobal Administration → ©Linked Data → Linked Data Mapping)

#### There administrators can:

- Map Work Item Types to Linked Data types.
- Map Link Roles to Linked Data link types.

#### **Configure Linked Data semantics**

Linked Data Semantics are defined on the Global Administration level and are described on the Linked Data Semantics Administration page:

(Select Project → FAdministration → FGlobal Administration → ©Linked Data → © **Linked Data Semantics**.)

#### There administrators can:

- Determine linking rules based on semantic Linked Data types.
- Define how backlinks are created.
- Define the relationship between Linked Data domains and semantic types.
- Define prefixes used in the Linked Data Semantic and Mapping configurations.

## Warning

When adjusting the configuration in Linked Data Semantics it is essential to make sure that the external Linked Data application understands the adjusted semantics.

Adding a custom resource or link type will not work if the external application does not understand the configured custom types.

(Polarion's default configuration only maps the default Work Item Types.)

See the Semantic configuration page to categorize custom Work Item types as a Change Request or Requirement.

# **Configure synchronization**

## Configure synchronization overview

This topic covers how to configure a connection to a supported system like Atlassian Jira, HP ALM or DOORS via a Polarion Connector.

#### You will need to:

- · Configure a Connection.
- Configure a synchronization pair.
- · Configure Polarion settings,
- Configure the settings of the connected system,
- Adjust some general settings,
- Map Workd Item types, fields and values to types, and the fields and values of the third-party system.

## Data backup

Before you begin using any Connector(s), be sure to have your system backup administrator include the Connectors data folder in routine backups. Losing Connector data will result in items being recreated rather then updated when systems are synchronized.

For details, please refer to: Connectors Data.

#### qiT

- 1. Be sure to read this entire section before configuring and creating a synchronization pair, so you become familiar with all the capabilities.
- 2. Spend some time planning out your data synchronization between Polarion and the third-party systems. As already stated, ensure you have a backup process in place, in the event you need to return to a previous state.

## **Configure connections**

A **Connection** refers to a specific third-party server containing the data you want to replicate. Objects from the third-party system are identified based on their ID and the connection that was used to synchronize them.

## Tip

- If you specify multiple connections for the same third-party system you will duplicate items.
- Only a single synchronization can run with a specific connection at any given time.

#### Create a new connection

- 1. Open the Polarion project that corresponds to third-party's project that you want to synchronize data with. (You will need administrator permissions for this project.)
- 2. Open Administration, and in Navigation select Connectors to access the Connectors page.
- 3. In the **Connections** section of the page, click \*Add Connection in the Actions column.
- 4. In the Create new connection to dialog, select the third-party system to connect to and click Next.

(The list contains the systems currently supported by Polarion Connectors.)

- 5. In the Create New Connection To [SYSTEM] dialog, specify the Connection ID that identifies items from a specific server and the fields required by the third-party system.
  - (Use an ID name that helps you identify that unique connection.)
- 6. Click **Create** to finish configuring the Connection.

## **Create links between different Projects**

Because items are identified by their ID and the connection that creates the links between projects needs the same connection, use a global connection to synchronize linked items to different Polarion projects. (This ensures that the links between the items are recreated in Polarion.) For example, to import links from defects in Project B to requirements in Project A. In this case, defining the mapping of the link parameter for the defect type is sufficient.

If the items are not in one sync pair, it may be necessary to execute the synchronization twice, because links cannot be created when there is no target for them yet. The items must be created during the first sync, and then on next sync they will be linked.

## Create a global connection template

Connections can be configured globally for an entire repository. When created in the global scope, Connections become Connection Templates, which can be reused in different projects. The login credentials for Connection Templates are not stored in the User Account Vault. Users can enable Connections defined in the global scope in any project by supplying login credentials. You can create entirely new Connection Templates, or save any existing project-scope Connection configurations as global Connection Templates.

To create an entirely new Connection Template, follow the same steps as outlined previously in Creating a New Connection, but instead of opening a project, open the Repository.

## Reuse an existing connection as a global connection template

- 1. Open the Connection for editing in the Connectors topic of Administration. Use the drop-down control by the **Save** button, and select **Save As**.
- 2. In the Save Connection Configuration dialog, select the create new configuration option, and in the Scope field, select Global.

3. Specify an ID for the new Connection configuration and click **Save**.

## **Configure / Create a synchronization pair**

The **Sync Pair Configuration** page provides settings for Polarion and the third-party server connections, data propagation, and data mapping.

#### Note

#### Before creating a Synchronization Pair:

Make sure that you've created a Connection before you create and define Synchronization Pairs.

A Synchronization Pair defines how a specific type of information is exchanged and synchronized between Polarion and the target third-party system configured in the Connection. Polarion includes several Synchronization Pair templates that are pre-configured with the commonly required settings, needed to facilitate the configuration of new Synchronization Pairs. You can optionally save Synchronization Pairs that you create and configure, as templates.

## Create a synchronization pair

- 1. In the **Synchronization Pairs** section of the page, click **Add Sync Pair** in the **Actions** column.
- 2. In the **Template** field, select the template for the type of synchronization you are creating. The list of templates is filtered according to the setting of the **Connect to** field, so you may need to set that field in order to select the right template.

#### **Note**

If you haven't created a synchronization pair before this, then no templates will be available.

- 3. Specify a unique identifier for the new Synchronization Pair in **Sync Pair ID**. This ID can be specified in Jobs that run this Synchronization Pair. Example: **Jira Sync 1**
- 4. Select a Connection in the **Connect to** field. The list in this field is populated with the names of all existing Connections.
- 5. Click **Create** to create the Synchronization Pair configuration.

The **Sync Pair Configuration** page for the new Synchronization Pair loads and displays settings and data mappings.

The sections that follow will cover these settings.

## **Configure Polarion settings**

In the **Polarion Configuration** section, set the following:

• **Account vault key** (*Required*): The authentication key from the User Account Vault that contains a user name and password that the Connector can use to access the Polarion server.

You may want to configure a new key specifically for use with Connectors, or even different keys for different Connectors if you connect Polarion to more than one third-party system. For details about Account Vault keys see: Configure the User Accounts Vault.

(The Account Vault configuration requires global administrator permissions.)

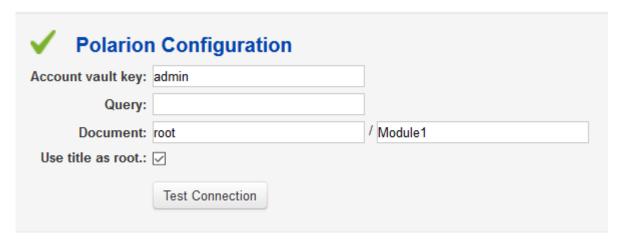
Query (Optional): But generally recommended. Enter Lucene query syntax to filter the Work Items that will be synchronized with the third-party system. For example, maybe only **Task** type items will be synced with JIRA. If that's the case, then you would specify a simple query like type:task. For Task items that are Open or Finished, you would need a more complex query: type:task AND (status:open OR status:done).

#### Warning

Once an item has been synchronized, it will continue to be synchronized until it does not match the query configured for Polarion and the query configured from the third-party system.

- Live Document (Optional): Specified as space name/Document name. If specified, only Work Items from the specified Document in the specified space will be synchronized. The Document items are filtered according to the filter (if any), in the Query field.
- **Use title as root** (Optional): Check this option if the title of a Document should be treated as the root element in the synchronization.

(The title itself will not be synced but all synced items will have it as an ancestor.)

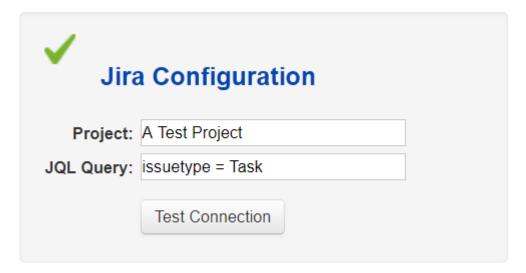


When you've set the desired properties, test the settings using the **Test Connection** button, a green check mark appears if the configuration is valid as specified. A descriptive error message appears if a setting is incorrectly configured. This connection, and the one to the third-party system described below must succeed before you can proceed to complete the Synchronization Pair configuration.

## Configure the settings of a connected system

To the right of the Polarion Configuration section there is another section with connection settings for the target third-party system. The fields vary depending on the system, but should be similar enough to Polarion's settings to recognize. (It is expected that you are familiar with the third-party system and understand the relevancy of the fields). Label semantics are taken from the third-party

system. After you specify these settings, test the connection with the section's **Test Connection** button. This connection, and the one to the Polarion server described above, must succeed before you can proceed to complete the Synchronization Pair configuration.



## **General settings**

The settings in this section let you control the propagation of new items and deleted items between Polarion and the third-party system.

- Propagate new items: Controls how new items are synchronized.
  - o **Bidirectional**: New items created in Polarion will propagate to the third-party system and vice versa.
  - o **Left to Right**: New items of the compatible mappedtype(s) created in Polarion will propagate to the third-party system, but new items created there will not propagate to Polarion.
  - o **Right to Left**: New items created in the third-party system will propagate to Polarion, but new items created in Polarion will not propagate to the third-party system.
  - o **Disabled**: Prevents all propagation of new items between the two systems. It is likely to be less commonly used than the other options.
    - (Only if there's a case where the Synchronization Pair should only be applied to items created before the Sync Pair configuration.)
- Propagate deletes: This setting works much like Disabled except that it controls how deleted items are synchronized.

General Settings		
	Dir. C. I	
Propagate new items:	Bidirectional	~
Propagate deletes:	Disabled	~

## Type and Field and value mapping

After you've tested both connections as described in the previous section, the Mapping section appears in the lower part of the page. There are two mappings to complete:

- Type mapping: Maps one or more artifact types of your Polarion project to their equivalent artifact type in the third-party system's project.
- Field mapping: Maps one or more Work Item fields of your Polarion project to their equivalents on the third-party system, and specifies the direction and precedence of synchronization.

First, map the Polarion Work Item type(s) you want to synchronize to their equivalents on the third-party system. Create a row in the Type Mapping section for every Work Item type you want to synchronize.

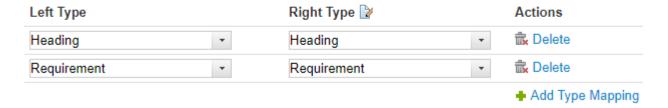
## To define a Type mapping

- 1. Click Create New in the Actions column of the Type Mapping section.
- 2. In **Left Type**, select a Polarion Work Item type in the list. This type will be synchronized with a third-party type specified in the Right Type column.
- 3. In **Right Type**, select a third-party artifact type in the list. (The third-party system's equivalent of the Polarion type specified in the Left Type column, so that the fields/attributes of these two types can be synchronized.
- 4. If more types need to be synchronized, create a new row in the **Type Mapping** section using Create New.

#### Tip

Make sure that you map all types that might be synchronized in accordance with the queries defined in the Polarion and third-party configurations. If you only want to sync specific types, you must configure the queries to filter your target types. For Polarion set the Query field in the Polarion Configuration section and have it filter for all types defined in the Type Mapping section. For example, if only Tasks are to be synchronized, specify a query for the *Task*Work Item type.

# Type Mapping



After mapping the type(s), you can configure how fields of Polarion Work Items are mapped to fields or attributes of the equivalent artifact(s) on the third party system. You do this in the Field Mappings section. You can specify field mappings applicable to all types, and field mappings applicable to specific types. The latter is especially needed if you want to synchronize one or more type-specific fields.

Field mappings also specify the synchronization direction, and in the case of bidirectional synchronization, the precedence.

## **Customize a Field Mapping**

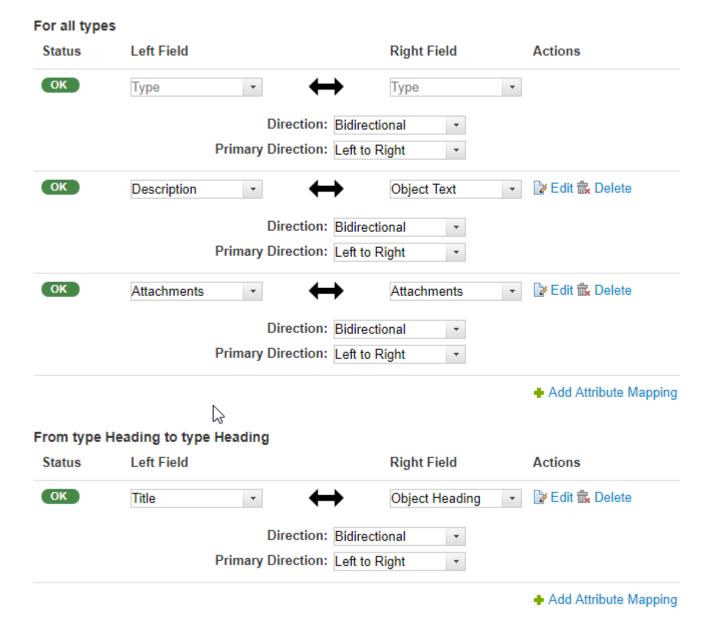
- 1. Select a value in the **Left Field** list. The list contains Polarion Work Item fields, and the one you select is to be matched with a field of the third party system: *Title*, for example.
- 2. Select a value the **Right Field** list. The list contains names of fields or attributes of the third-party system. The field you selected in Left Column will synchronize with the selected field on the third-party system. For example, if the third-party system is JIRA and you selected *Title* in the previous step, then you would probably select *Summary* in this field.
- 3. Select a value in the **Direction** list. This specifies if the synchronization is bi-directional, or only from Polarion to the third-party system (Left to Right), or only from the third-party system to Polarion (Right to Left).
- 4. If you selected **Bidirectional** in the previous step, you should review and set **Primary Direction**. This is the direction that will be take precedence if there is a conflict. For example if the *Title* field is changed in Polarion and its equivalent is also changed in the third-party system between synchronizations, there will be a conflict when the next synchronization occurs.
  - Choose which direction should take precedence. For example, if you select **Left to Right**, the *Title* value in Polarion will overwrite the changed value of the equivalent field or attribute in the third-party system during the next synchronization.
- 5. Click **OK** when you finish the mapping. To define additional field mappings, click **Create New** in the **Actions** column and specify mappings for the fields, direction, and precedence.

If you want to map fields that are configured only for a specific Work Item type (a custom *Task Category* field for the Task type, for example), specify the mapping in the last subsection of **Field Mappings**.

## Tip

- **Very Important!** Do not try to synchronize to the **Type** attribute since type is already mapped via the configured **Type** mapping. Overwriting the **Type** using a field mapping will cause unpredictable behavior.
- The UI does not display which attributes are read-only, but you can check the sync execution log. It displays error messages when any attempt to overwrite a read-only attribute is made.
- When synchronizing rich-text attributes like the **Description** in Polarion and Jira or **Object Text** in DOORS, you should also consider synchronizing attachments. (They might be referenced or embedded within the synchronized rich-text and will otherwise be missing.)

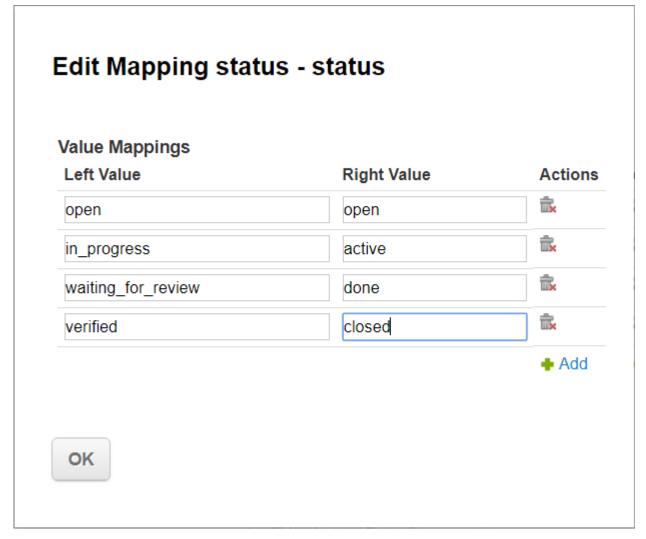
# Field Mappings



Within field mappings, you can also specify **value** mappings. This is required when you need to synchronize values like *users* in a field like **Assignee**, or an *enumeration* value in a field like **Severity**. Click the **Edit** link in any field mapping to display a dialog in which you can set value mappings. Click **Add** in the dialog to add a new value mapping pair. Specify the Polarion value in the **Left** field, and the third-party value in the **Right** field. You can create multiple value mappings for any mapped field. The value that will be used in Polarion (left side) is the **ID** of the enumeration option, category or user. The value that will be used for the third-party (right side) depends on the system that you are synchronizing to.

## Tip

If you are unsure about what value is used in the third-party system, start with synchronizing the value to Polarion without mapping it. Polarion will set the value to whatever is used in the third-party system and you can then define the value mapping based on that.



When you have finished all the mapping, click **Save** at the top of the page to save the Synchronization Pair. You should now be able to synchronize data between Polarion and the configured third-party system. You can do this either manually, as described in the next section, or by setting up a scheduled Job to run the synchronization automatically at an interval of your choosing.

## **Execute a Synchronization Pair**

After you have created a connection and added a Synchronization Pair for it, you are ready to synchronize data between to two systems.

To explicitly invoke synchronization manually, click the **Execute** link for the Synchronization Pair on the **Sync Configuration** page. The progress and log of the synchronization can be seen in the 4-Monitor topic of project Navigation. (For testing, it can be useful to open this topic in a different

browser tab before executing the synchronization.) Look for the job named **Synchronization of [SYNC PAIR NAME]** (where *[SYNC PAIR NAME]* is the name of the Synchronization Pair executed).

As already mentioned, you can automate synchronization by creating a scheduled Job to execute the Synchronization pair at an interval of your choosing. You create jobs in the Scheduler topic of Global administration. For more information, see Configure the Scheduler.

## **Job Configuration**

In the Scheduler, you need to create a new <job> element. Since Synchronization Pairs are configured in the project scope, it is essential to specify the project ID in the scope attribute.

## Example 1:

- Use synchronizer as the value for the id attribute.
- Use any meaningful string in the name. This is the name that will appear in the Scheduler's list of jobs.
- In the above example, replace MyProjectID with the actual ID of the project that the Synchronization Pair to be executed is configured in.
- The cronExpression attribute is optional. It's only used if you want the job to run automatically at some defined interval.
- The <syncPair> element, replaces the [SYNC PAIR ID] of the actual ID of the Synchronization Pair that will be executed by this job. To avoid errors, open the Synchronization Pair configuration in another browser tab and copying the configured ID.

## Example 2:

## **Configure frequent synchronization**

When teams work concurrently with Polarion and another tool, such as Jira or Microsoft Team Foundation Server (TFS), it can be desirable to run the synchronization between these systems frequently so that the teams see changes in a timely manner. The scheduler in Polarion can be configured to run a scheduled job that in turn runs Sync Pairs. The job can be configured to run Sync Pairs as often as desired, within certain limits — see Caution below.

Because Sync Pairs typically have more items in scope than necessary, and synchronization time increases with the number of items, the scope should be restricted when a Sync Pair is run frequently by a job. You can limit the scope by using a system-specific query expression. For example, the expression <code>updated:\$today\$</code> limits the Sync Pair to Polarion items that have been changed on the current date. A similar query must also be used for the remote system.

## Caution

The frequency of the job running a Sync Pair must be larger than the running time of the Sync Pair when the maximum number of items are in scope. For example, in a scenario where 500 items are changed daily, and the running time of the Sync Pair for these 500 items is 3.5 minutes, the frequency of the scheduled job that runs the Sync Pair should be at least 5 minutes. Otherwise, there could be a number of jobs stacking up as newly triggered jobs wait for the previous one to finish.

## Reuse Sync Pairs as global templates

Synchronization Pairs can be configured globally for an entire repository. When created in the global scope, Sync Pairs become Synchronization Pair Templates, that can be reused in different projects. Each Synchronization Pair Template must use an existing Connection defined in a Connection Template. You can create entirely new Synchronization Pair Templates, or save any existing project-scope Synchronization Pair configurations as a global Synchronization Pair Template.

To create an entirely new Synchronization Template, follow the same steps as outlined in Create Synchronization Pair, but instead of opening a project, open the Repository.

To reuse and existing Synchronization Pair as a global Synchronization Pair Template:

- 1. Open the Synchronization Pair for editing in the \*\*Connectors topic of \*\*Administration. Use the drop-down control near the Save button, and select Save As.
- 2. In the Save Synchronization Pair Configuration dialog, select the Create new configuration option, and in the Scope field, select Global.
- 3. Specify an ID for the new Synchronization Pair configuration and click **Save**.

## System specific instructions

#### **DOORS**

## Warning

Before importing, read the Import Limitations and Recommendations section.

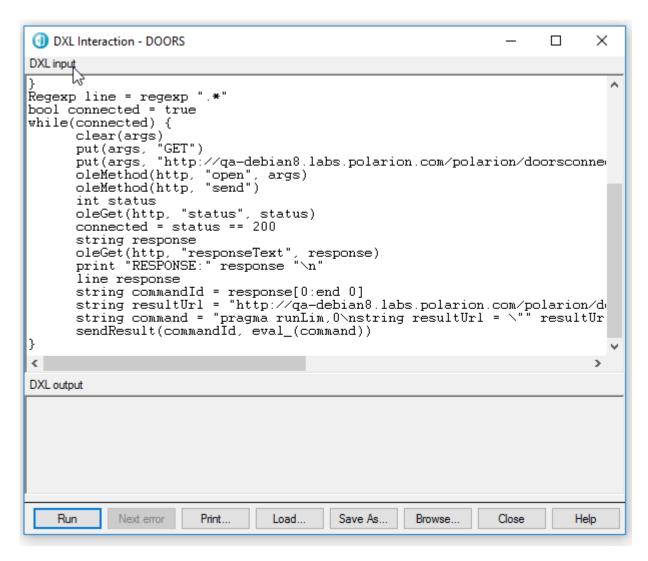
#### Connecting to DOORS:

The connection to DOORS only requires the ID used to identify the target DOORS database. The connection is established from the DOORS client to Polarion by running a DXL script.

Click Connect to activate the connection and run the DXL script in a DOORS client (Tools - Edit) connected to the DOORS database containing the module you want to synchronize.

## **Note**

- While the script is running, the DOORS client will be locked until you close the connection.
- Running the script from a Module can lock up the sync. (The connector might attempt to close the DOORS module.)



If you are planning to schedule the synchronization, run the DOORS client on a workstation where the connection can be kept active.

(See Schedule Synchronization with DOORS for details.)

#### Creating a Sync Pair for Doors and Advanced Type Mappings

## **General Setup:**

When synchronizing with DOORS you typically want to synchronize Live Documents in Polarion to Modules in DOORS and include the structure in the specifications. This will allow you to easily

control the mapping to Polarion Work Items, since they will be in the context of the Live Document you have chosen to synchronize to.

To synchronize the structure, one sync pair must be defined per Module/Live Document pair. Make sure that you configure the sync pair so that it points to a specific Live Document in Polarion and Module in Doors. (See Configure a Synchronization Pair for details.)

## There are two ways you can configure a DOORS sync pair:

## 1. Use the Module ID;

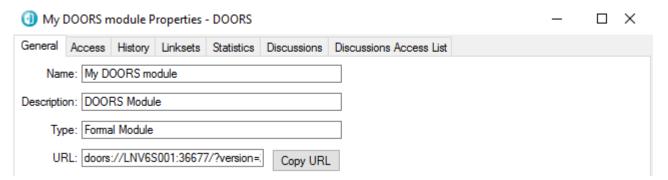
a. Open the DOORS module you want to synchronize, go to **Tools DXL**, then insert following script and run it:

```
Module m = current
print qualifiedUniqueID(m)
```

b. Put the value you get after execution of the script in the appropriate field in the Doors Sync Pair in Polarion.

## 2. Use the Module URL;

- a. Choose the target module in Doors, then right-click to open the context menu and choose **Properties**.
- b. In the opened General tab click Copy URL beside URL.



c. Paste the value in the appropriate field in the Doors Sync Pair in Polarion.

## Tip

Check the **Use Title as Root** option to sync Work Items below the first title in Polarion because it is usually a better representation of a DOORS Module's structure. If you have additional content below the first title, depending on the synchronization pair settings, the sync operation may attempt to create objects in DOORS to synchronize this Live Doc.

You can set the Creation and Hierarchy directions in the following ways:

- **Right to Left**: To replicate requirements from Doors to Polarion.
- Left to Right: To send new Work Items from Polarion to Doors.
- Bidirectional: To synchronize new requirements.

## **Type Mappings:**

Before beginning the mapping process, spend some time determining how you want the data mapped between DOORS and Polarion. The Synchronization *will not* create Polarion Work Item types and fields automatically. (They should all be created before you start mapping.)

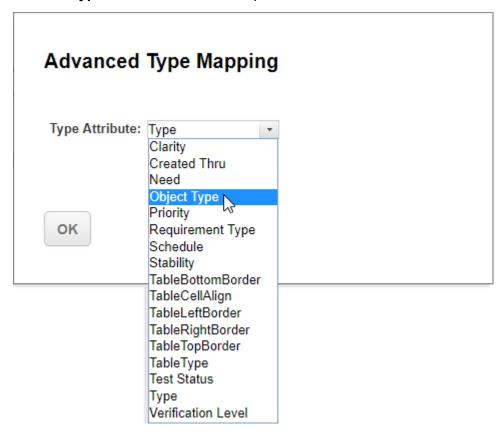
The DOORS connector will treat objects without **Object Text** as **Headings** and other objects as **Requirements**.

Since DOORS does not differentiate between different types of objects in a module, it's common practice to define the attribute type in DOORS.

These types can then be mapped via the Advanced type mappings dialog.

## To access Advanced type mapping:

- 1. Click mext to Right Type.
- 2. Select a **Type Attribute** from the drop-down box.



3. All values of the selected attribute should now be available for type mapping.

## Note

Mapping to the Work Item type heading will impose some limitations on the way that the structure can be represented in Polarion.

## Tip

You can always choose to use a Work-item of type called something like **Doors heading** to replicate the DOORS structure verbatim inside of Polarion.

Field mappings will vary depending on your needs, but consider following DOORS specific suggestions:

- When syncing any rich text attributes, you should also include attachments because the rich text fields might include embedded objects.
- VERY IMPORTANT! Avoid syncing back Object Heading for non-heading Work Item types because Polarion will automatically populate the heading and returning these auto generated headings might not be desirable.

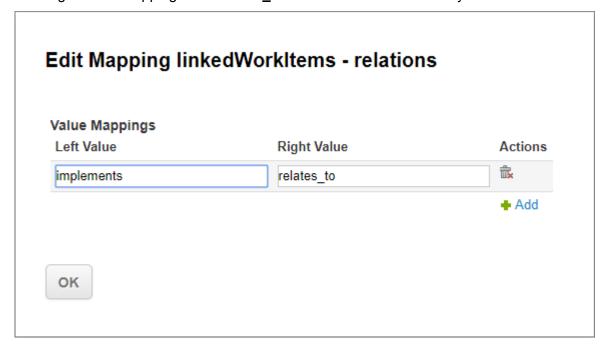
The connector will use the name of a DOORS option for Value mappings. Define Value mappings accordingly, especially when syncing an attribute to Doors. (Attempting to update an attribute with an invalid value will fail).

## **Mapping Links:**

Links can be synchronized from DOORS to Polarion by defining an attribute mapping from Relations to Linked Work Items.



By default, the link created in Polarion will use the **relates to** link role. It can be configured by defining a value mapping from **related to** in DOORS to the link role you want to use in Polarion.



Different roles can be used per Object or Work Item type by defining the attribute mapping for links in the **Type-specific Mapping** section.

#### Limitations:

The same limitations as a DOORS Import apply.

Bidirectional synchronization of rich text can lead to loss of Polarion data because DOORS' ability to store rich text is very limited.

Currently it is not possible to synchronize links added in Polarion to Doors.

When adding .rtf files containing OLE objects in Polarion, syncing them to Doors will result in the reversion of an attachment with an adjusted name (<Doors Attribute> <Index>.rtf)

## **Scheduling Synchronization with DOORS:**

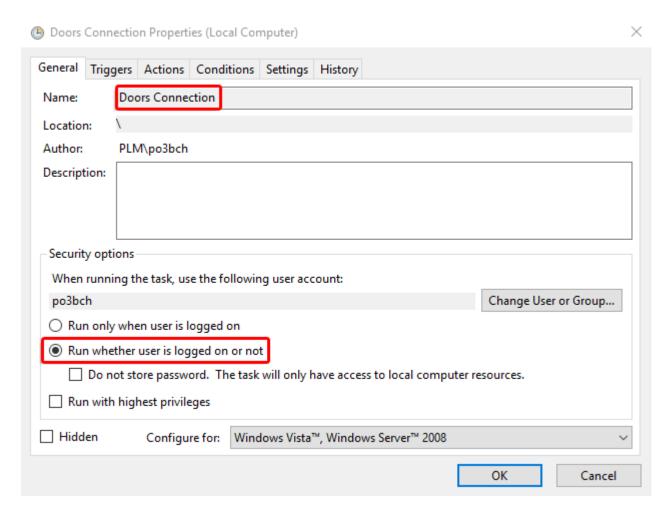
When configuring synchronization with DOORS as a scheduled job, you must keep the DOORS client running in the background. It must be done on a workstation that can connect to the Polarion server and the DOORS server.

(The DOORS server must also have the DOORS client installed on it.)

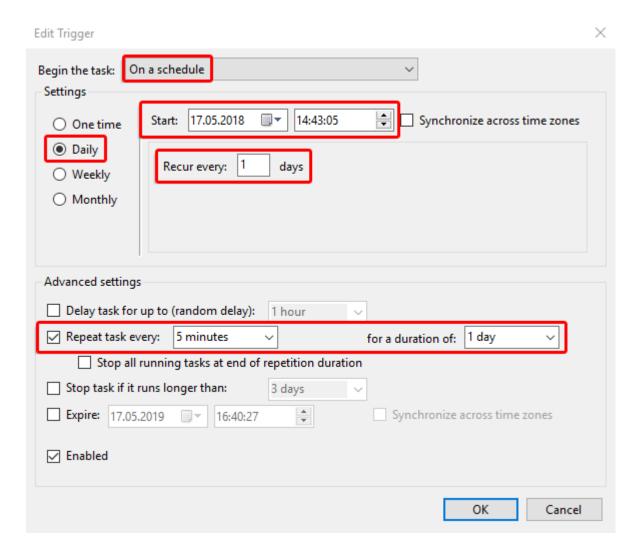
The DOORS client can be executed as a scheduled job to ensure that the client is started automatically. (Just in case the workstation is restarted or the Doors client crashes.)

#### Add the scheduled DOORS connection:

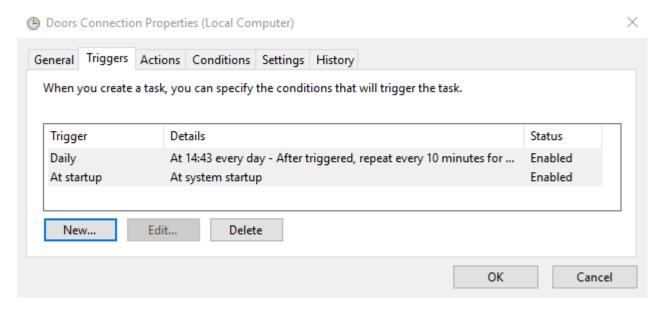
- 1. Create a c:\doorssync folder.
- 2. Create a connect.dxl file and populate it with the connection script.
- 3. Open the Task Scheduler Windows application.
- 4. Click \* Create Task.



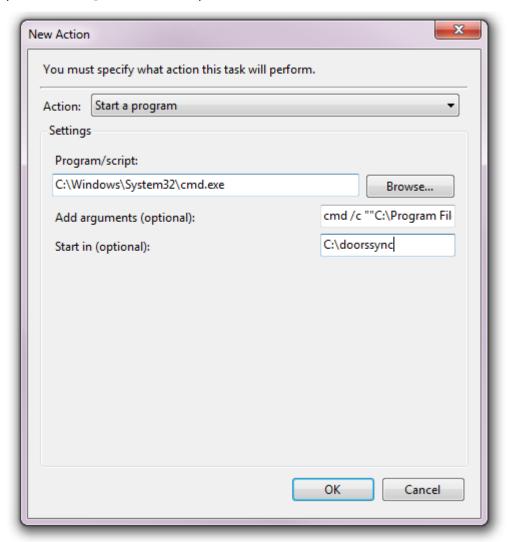
- 5. Name the task "Doors Connections" (for example), and set it to **Run whether users is logged** on or not.
- 6. Click on the **Trigger** tab, click **New**, select **At Startup** from the **Begin the task** drop-down menu and click **OK**.
- 7. Click **New** again on the **Trigger** tab, select **On a Schedule** from the **Begin the task** drop-down menu, click on **Daily** set up a time like the screenshot below and click **OK**.



8. Both triggers should be listed and appear as *Enabled* in the **Trigger** tab.



- 9. Click on the Actions tab and click New....
- 10. Select **Start a program** in the Action drop-down box.
- 11. Under Program/script: click Browse... and select the cmd.exe file. (C:\Windows\System32\cmd.exe).



12. Add the following to the **Add arguments** field:

(Replacing USER and PASSWORD with the credentials of the doors user you want to use for sync.)

```
cmd /c ""C:\Program Files\IBM\Rational\DOORS\9.6\bin\doors.exe" -b connect.dxl -user
"USER" -password "PASSWORD" -W nowait 1> doors.out 2> doors.err
```

13. Add c:\doorssync to the Start in field:

#### **HP QC or ALM Database**

#### **Connection settings**

When configuring a connection to HP ALM or Quality Center, you can optionally specify a direct connection to the SQL database using JDBC. While this connection is optional, it is not possible to access (read/write) links from tests to requirements, and between requirements, without it.

To successfully connect, you need to specify a value in the **JDBC Connection String**. This value can be derived from the Connection String in the DB Servers tab of HP ALM's Site Administration. When you enter the string into Polarion, replace mercury with jtds (for a Microsoft SQL database), or oracle:thin (for an Oracle Database).

## Sync Pair settings

**Filter (HPQC)**: A filter in HPQC can be used like a Work Item query to filter entities in HPQC. The syntax for a filter is: field\_id[filter]. For example a filter to have only entities with ids between 5 and 10 would be:id[>=5 And <11]. In the filter block the following operators can be used:

- Comparison operators:<,>,<>,>=,<=,</li>
- Logical operators:and,or,not,
- Literals: A string that represent the value of an expression. If a value contains spaces, it must be contained within single or double quotes. Literals can contain the '\*' wild card.
- Pairs of parenthesis.

Multiple filters can be combined using the AND or OR operator. For example the queryid[>3] AND title['foo ba\*'] is a valid query to filter out all entities with an ID larger than 3, and a title starting with foo ba.

**Parent ID (HPQC)**: The ID of an entity in HPQC, that will be used to define a root element in HPQC. Every entity from Polarion will be put below this root element in HPQC, and only entities that are below this root element will be transferred from HPQC to Polarion. The parent ID can only be used for the "Requirement" HPQC entity type, since its the only element that supports hierarchies.

#### Jira

**JQL Query (JIRA)**: A JQL (Jira Query Language) can be used like a Work Item query to filter entities in JIRA.

For the JQL Syntax please consult the official JIRA

documentation under https://confluence.atlassian.com/display/JIRA/Advanced+Searching.

Jira Formatting - (What's preserved and omitted when a Jira project is synchronized):

When synchronizing rich-text from Jira to Polarion the connector will use the conversion (to html) provided by Jira. This will accurately render the rich-text in Polarion for most Jira formatting. (Including all of the formatting listed below.)

When synchronizing rich-text from Polarion back to Jira only the formatting listed below will preserved. (All other formatting will be removed.)

Character	Formatting Preserved
+	Underline
*	Bold
_	Italic
-	Strike-through
(/)	Icon check
(x)	Icon failed
(i)	Icon information
??	Citation
Λ	Superscript
~	Subscript
{quote}	Block-quote
{color}	Text color

(Characters placed before and after the target text.)

The following can also be synchronized with Jira:

- **Tables**
- Images and image references

(If you want to make an image reference on the Polarion side, the attachment link should have same label as its file name.)

#### **Note**

### Formatting omitted:

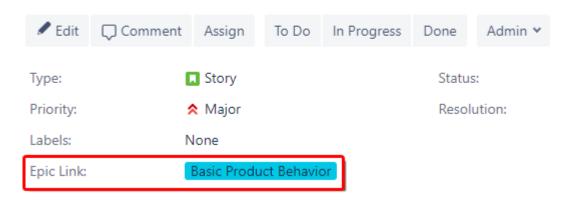
- All other Jira formatting that is not listed in the above table.
- All formatting done within Microsoft Word.
- Formatting changes within a single word. (For example, Polarion)
- Formatting that begins in the middle of one word and ends in another. (For example, polarion alm tool.)

### Sync Jira Epic Links with Polarion

Assigning User Stories to Epics is common practice in Agile projects. Polarion and Jira do this in different ways:

- In Polarion it's usually done by using a link role like **Parent** or **Implements**.
- In Jira, it is done using the **Epic Link** field.

## **Password Regex**



### (The Epic Link field in Jira.)

If you are synchronizing both Epics and User Stories between Polarion and Jira, you'll probably also want to synchronize the links between Epics and Stories.

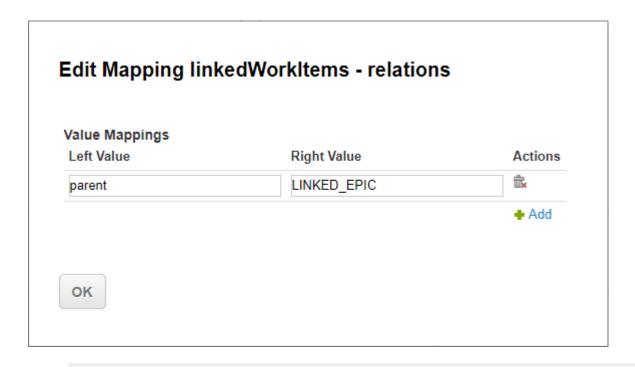
### To synchronize the Jira Epic Link attribute with Polarion Work Item links:

1. Add a field mapping in the type-specific **Field Mapping** section for *User Stories*.

### From type User Story to type Story



- 2. Click **Edit** on the right, then click **Add**.
- 3. Enter the link role used in Polarion as the **Left Value**, **LINKED\_EPIC** as the **Right Value** and click OK.



### Warning

Make sure that the link role is configured in Polarion and **LINKED\_EPIC** is in all caps.

4. Click Save at the top.

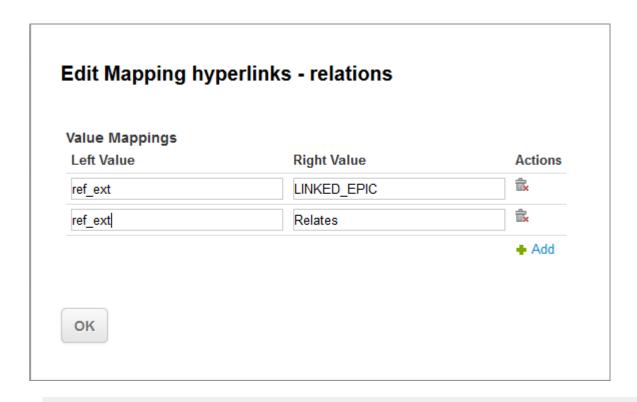
### Sync Jira Links to Hyperlinks

You can also synchronize Jira links (both regular and Epic) to hyperlinks in Polarion:

1. First, define an Attribute Mapping from Links to Hyperlinks. (Unidirectional – From Right to Left.)



2. Then define a Value Mapping for each link type that you want to synchronize with Polarion, mapping the link type to a link role defined in Polarion.



If you don't map a link, no hyperlink will be created for that link type.

### **Microsoft TFS**

### **Note**

The Microsoft TFS (Team Foundation Server) Connector requires an additional license.

### Limitations

- The file name of an image must not contain the ampersand (&), backtick (`), tilde (~), or hash sign (#) characters when inserting an image into a TFS rich text field.
- An image inserted into a TFS rich text field, must have a different name than existing Work Item attachments.
- If attachments have identical file names, only the attachment that was added first will be synced.
- Naming an attachment image.png may cause issues if multiple images are copied and pasted into a TFS rich text field.
- The propagation of item deletions is only supported from TFS to Polarion. (Not from Polarion to TFS.) If synchronization is configured bi-directionally, TFS deletions are propagated to Polarion but Polarion deletions are not propagated to TFS. This results in the recreation of deleted items in Polarion.

Currently, we do not support syncing Emojis, Headings and Code styled text from or to TFS rich text fields.

#### Warning

If a TFS project is renamed, the connection database must be updated to prevent the creation of duplicates.

(The TFS connector has a single connection database per Connection, Collection and Project Name tuple/combination.)

Please contact support for more details.

### Supported versions

- Azure DevOps (Cloud)
- TFS 2015 or newer (On-Premises)

### **Connection settings**

To connect to a TFS server, you need the URL of the server and a personal access token or username and password. Access tokens are the preferred authentication mechanism unless you are trying to connect to TFS 2015. You can create new access tokens in the **Security** settings of your Profile in TFS. The token must be configured with **READ** and **WRITE** access for Work Items.

### Sync pair settings

You need to configure Project Collection and Project to configure a sync pair. The WIQL Query can be used to limit the scope of the synchronization. The expected content is the "where" part of a WIQL query. (See Microsoft's Documentation for details.)

### Sync Microsoft TFS links with Polarion

Both Microsoft's TFS and Polarion use directed links. This lets you group items in parent child relationships like the following:

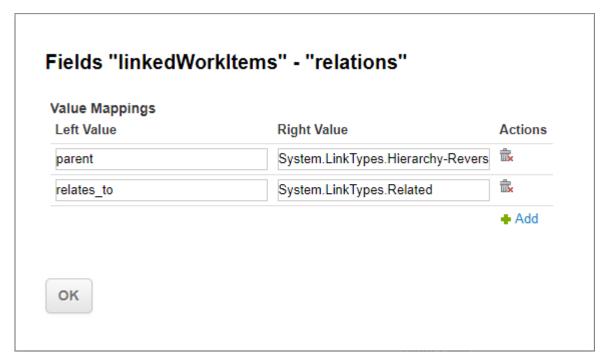
- Item A has a directional link of is the parent of item B.
- Item B has a directional link of **has parent** back to item A.

#### Tip

When mapping Polarion link roles to Microsoft TFS link roles use the **ID**, for example parent, of the Polarion link role and the ID of the reverse direction (**System.LinkTypes.Hierarchy-Reverse**) of the Microsoft TFS link role.

# From type Change Request to type Feature





Unlike Polarion, Microsoft TFS also uses undirected links, where two items can share the same **Related** role. When they are mapped to Polarion, they are assigned a direction: from the TFS item with the lower **ID** number to the TFS item with the higher **ID** number.

### Tip

When mapping Polarion link roles to undirected TFS link roles, use the **ID** of the Polarion link role, for example **relates\_to** and the **ID** of the undirected TFS link role (**System.LinkTypes.Related**).

### **MATLAB Simulink**

### **MATLAB Simulink overview**

This topic covers general steps for configuring a connection to a supported MATLAB Simulink ("MATLAB") system via the Polarion Connector for MATLAB Simulink.

You will need to:

Download the Polarion Connector for MATLAB Simulink.

- Install the Polarion Connector for MATLAB Simulink.
- Configure a connection between your Polarion server and your MATLAB installation.
- Publish MATLAB elements to Polarion Work Items.

### **Prerequisites for installation**

It is assumed that you already have an installation of a Polarion application lifecycle management product installed, configured, and running.

To use the Polarion Connector for MATLAB Simulink, you need the following:

- MATLAB Simulink version R2013a or later, and Simulink Verification and Validation.
- Any version of Polarion ALM, Polarion Requirements, or Polarion QA (latest version is recommended).

The Polarion PRO license is supported. The Polarion Reviewer license is not supported (it does not allow users to create new Work Items).

The connector may be integrated with a customer's instance of Polarion SaaS, but this requires customization of the instance by Polarion IT Services at additional cost.

#### **Connector overview**

Experienced MATLAB users know that it has the capability to link model elements and subsystems in diagrams to external requirements specifications maintained in Microsoft Word. Polarion leverages this capability to enable linking of MATLAB model components to requirements managed with Polarion application lifecycle management products. This can be a significant advantage for MATLAB users who design components in companies and industries with stringent compliance mandates.

Polarion's technology is designed from the outset to facilitate wide and deep traceability through every engineering phase. In the case of MATLAB and Polarion, MATLAB model elements and subsystems can be linked to granular requirements specifications maintained in Polarion's exclusive LiveDoc documents. Process is efficiently managed across disciplines and stakeholders by means of workflow, which can be customized to suite the customers exact needs (if these fall outside the parameters of Polarion's robust default workflows for software and systems engineering projects). Inevitable change during the product lifecycle is easily managed and, most importantly, automatically tracked so that at any given time it is possible to trace who changed what, when, and why.

Traceability extends onward into testing. Test cases are linked to requirements in Polarion, which in turn link to MATLAB components, so it is easy to understand the extent of test coverage for all components and what is tested (and how). A comprehensive historical record of testing activity is automatically maintained as well.

### **How Traceability is Created**

Creating this robust traceability is like the old joke about eating an elephant one bite at a time. By creating appropriate links every time an artifact is created, you end up with forensic levels of traceability that can satisfy the most exacting requirements.

For example, suppose the process starts with business cases created in a business case specification document. When requirements engineers create e.g. a functional requirement, they link it to the relevant business case(s) with appropriate link role(s) which describe the nature of the relationship.

Then when testers create a test case. they link it to the requirement(s) it verifies, again with an appropriate link role (e.g. "covers" or "verifies").

When engineers use MATLAB to design some hardware component, they have the option of linking model elements to existing requirements in a Polarion project, or creating and linking to new requirements in Polarion. Such new requirements can either be written up by the modeling engineer, or assigned to a requirements engineer for completion. As requirements are completed, they can be assigned to testers (automatically via workflow) who them create and link test cases. The end result is traceability from a MATLAB model element to a requirement (or multiple requirements), and from the requirement(s) to test cases, which later on will be linked to Test Runs to record manual or automated test executions.

This guide explains how to set up a connection between a MATLAB installation and a Polarion project hosted on a Polarion server, and how to link model elements to existing or new requirements (or other artifacts) in a project hosted on a Polarion server.

#### **Installation procedure**

#### Install the Polarion Connector for MATLAB Simulink:

- Extract content of the ZIP distribution archive file in an empty folder on the computer hosting MATLAB.
- 2. Start MATLAB and switch to the folder (e.g. type 'cd <folder path>')
- 3. Run the startup.p file.

### Note

MATLAB might warn you: "MATLAB cannot run this file because <some file> shadows it in the MATLAB path". You may select option **Change Folder** or **Add to path** to continue.

When you execute the installation program for the first time, it can display additional instructions on how to extend your existing environment (if you already have startup.p files in your paths).

### **Connect MATLAB to Polarion**

This section explains how to set up a connection between MATLAB and your Polarion server, and a project on that server. It is assumed that you already have a project created in Polarion to manage the requirements (and possibly test cases and testing as well) for the MATLAB project you want to connect and link to. If you do not have such a project, you should create one before going further. See the Create Projects for details.

It is further assumed that you have the connectivity scripts for Polarion set up and configured in your MATLAB installation.

It is not necessary that requirements pre-exist in the project, as you will be able to create these in Polarion as you being linking elements of your model in MATLAB to Polarion.

If you have existing requirements in Microsoft Word, you can import these into Polarion to take advantage of the traceability benefits mentioned earlier. Consult the User Guide topic *Importing Microsoft Word Documents* in Polarion Help.

The connector for MATLAB supports efficient work with one Polarion project. If the MATLAB solution should be connected to several Polarion projects (or even servers with different URL or user credentials), it could lead to inconveniences with reconfiguring settings each time... when particular block needs to be revised, for example.

### **Connection Setup Procedure**

Before you begin, you might want to consider having your Polarion administrator create a dedicated user account to be used solely for the purpose of MATLAB accessing Polarion, with its credentials stored securely in Polarion's User Accounts Vault. This user account must have permissions to create and modify Work Items on the Polarion server in the project accessed by a MATLAB connection. For more information see Configure the User Accounts Vault.

### Set up a connection to a Polarion server and project:

- 1. In the MATLAB menu, choose **Tools Polarion Settings** to open the **Polarion Settings** dialog.
- 2. Specify the Polarion server's URL. For example, http://polarion.yourorg.com
- 3. Specify the user name and password to be used by MATLAB to access the Polarion server. The specified user must have permissions to create and modify Work Items on the Polarion server for the project specified in the **Project ID** field of the dialog. The credentials (and other settings) are stored in persistent variables of the local MATLAB installation.
- 4. Specify the ID of the Polarion project to be accessed by this Connection. Note that the ID may not be the same as the project name that appears in the Polarion user interface when you are logged in. The ID is shown in the **ID** field of the **Project** topic of Polarion Administration when you are logged in to a project. (You must have global or project administrator permissions to access it.)
- 5. **New Work Item Type** field, specify the ID of the Work Item type that should be created when a MATLAB user takes the option to link a model element in MATLAB to a new Work Item in Polarion.
  - Here again, you need the ID from the project configuration, not the Name that appears to end users in the user interface. You can find this in the project Administration: expand **Work Items**, select **Types**, and copy the value in the ID column. For example, the ID for type with the name *Requirement* is requirement.
  - The default value is <code>requirement</code>, but you can specify any type configured in the target Polarion project. For example, you might have a type <code>simulation block</code> (an aggregate type later decomposed into individual requirements).
- 6. The **Search Query** field is optional. Here you can enter Lucene query syntax for a query to isolate some subset of all Work Items when a MATLAB user takes the option of linking a model element to an existing Work Item in Polarion. This becomes the *default* query for the "Link Block/(Sub)System" action (described later). The query can be refined as needed when invoking the action.
  - For example, suppose elements from some model need to be linked to Requirement items in Polarion. Entering type:requirement will cause only Work Items of the requirement type to be retrieved when a user invokes linking to an existing Work Item. The query can filter the list even more. For example, maybe the links will be to unresolved requirements planned to version 2: NOT

HAS\_VALUE:resolution AND type:requirement AND targetRelease:2.0

7. Click the **Save** button to save the configuration.

Settings will be used for all subsequent sessions. To work with a different server or project on the same server, reopen the Polarion Settings dialog and modify the settings as needed.

### **Note**

If the user did not execute **Tools Settings**, then the Polarion context menu will offer only the option to open the Settings dialog.

When the connection is successfully completed, a context menu item **Polarion** appears on each type of element in MATLAB models.

#### **Publish MATLAB Elements to Polarion**

You should now have a connection configured and working between MATLAB and a project on a Polarion server. This chapter will explain various options for "publishing" model elements in MATLAB to requirements in Polarion. (Note that although we say "requirement", because it is the most common use case, the target item can be of any type configured in the project on your Polarion server.)

Publishing means creating a link to a Work Item in Polarion... either a pre-existing item, or on you create during the process of publishing. Once published, user can easily navigate from elements and subsystems in the MATLAB model to Work Items in Polarion, and vice-versa. You can publish:

- A single selected element to a single Work Item (existing or new).
- Multiple selected elements to multiple Work Items (existing or new).
- An entire subsystem (to a single Work Items that may later be decomposed into sub-items).
- An entire diagram to a single Work Item.

### Publishing to a new Work Item

You can select a MATLAB element and "publish" it to create a new Work Item in the configured Polarion project.

- 1. Select an element in a MATLAB diagram. The element must be one supported by MATLAB's Requirements Management Interface (RMI).
- 2. On the element's context menu, select **Polarion Publish Block/(Sub)System**. (If the element is not supported by RMI, the Polarion menu item visible, but all actions are disabled.)
  - A new Work Item of the type specified in the configuration for this connection is created in Polarion and displayed in a dialog that now pops up.
- 3. Click on the Work Item listed in the dialog and wait for your web browser to open a new instance and open the new Work Item in the Polarion web interface.
- 4. Optionally edit any data fields of the new Work Item... Title, Description, etc. and save the Work Item.

### Tip

Polarion users can now follow the link in the **Hyperlinks** section of the Work Item back to the element in MATLAB.

You can optionally link the same MATLAB element to multiple Work Items in Polarion. For example, you might link to two requirements: functional and non-functional.

The Polarion context menu in MATLAB now has an item Open Linked Work Item, which leads to a dialog in which you can click any listed item to open it in Polarion.

#### Publishing to an existing Work Item

You can select a MATLAB element and "publish" to link it an existing Work Item in the configured Polarion project.

1. Select any element in the MATLAB model and on its context menu, select **Polarion Link** Block/(Sub)System with Existing Item.

The Select Polarion Work Item dialog opens and lists all items found by the query specified in the MATLAB to Polarion connection settings. Note that the configured guery string is shown, and that you can modify it in this dialog to retrieve a different set of items.

- 2. Locate the target Work Item in the dialog, and select it.
- 3. Click Link Selected.

A hyperlink to the MATLAB model is added to the existing Work Item in the **Hyperlinks** section.

### **Bulk publishing to Polarion**

You can publish multiple elements of a MATLAB diagram to Polarion in order to create, say, requirements for them in a single operation. For example, you could publish two selected subsystems to create Requirements in Polarion that cover both of them.

- 1. Select multiple elements in a MATLAB diagram... subsystems for example.
- 2. Right click any selected element and on the context menu select: **Polarion Publish** Block/(Sub)System.
- 3. Confirm that you want to publish multiple selected items.

New Work Items are created for all elements selected and these are listed in the Following Work Items Were Linked dialog.

You can click any listed item to open it in Polarion in your web browser. Each item has the same title as the element in MATLAB, and also contains an image of the element (e.g. subsystem or block), as well as a hyperlink back to the element in MATLAB.

#### Publish an entire diagram to Polarion

1. Right-click on the diagram background, and on the context menu choose: **Polarion Publish** Block/(Sub)System.

A new Polarion Work Item is created in Polarion with the same title as the MATLAB diagram.

- 2. Navigate to the Work Item in Polarion in your browser by clicking on the item in the dialog.

  The Work Item's Description field contains an image of the entire MATLAB diagram.
- Edit the data of the new Work Item as desired and save it.
   Users can follow the link in Hyperlinks to the diagram in MATLAB.

### Tip

- Use the element context menu item **Remove Highlighting** in MATLAB to prevent the image in Polarion from showing highlighted elements.
- If you modify the diagram in MATLAB, use diagram context menu item **Refresh Published Diagram(s)** to update the image in the Work Item on the Polarion server.
- The diagram image is created/updated as an attachment to the Work Item, which
  is versioned in the repository. Older revisions of the image are accessible in the
  attachment history.

### **Use the Requirements Management interface**

MATLAB is designed to work with external requirements management tools. Polarion natively integrates with MATLAB, so you can access Polarion-managed diagrams/elements and modify their properties in MATLAB.

Right-click on any diagram element linked to Polarion, and choose **Requirements** on the context menu. The context menu indicates if the element is linked to any requirements. You can click any requirement beginning with Polarion: to navigate to the linked Requirement in Polarion.

Click **Add-Edit links** to see the link properties. You can use this dialog to unlink any element from Polarion (if an element was linked to the wrong existing Work Item by mistake, for example).

## **Configuration properties**

( ★Administration → Properties)

This where administrators can add and edit advanced configuration properties while the server is still running.

The properties can be set to affect the **Project** or **Repository** levels.

(Values that are not set on the **Project** level are inherited from the **Repository** level.)

### **Property inheritance**

#### Note

When viewing the **Properties** page for a selected **Project**, only properties that are set for that Project will appear.

(View the Properties screen on the **Repository** level to view all properties that are inherited by all Projects.)

### **Currently Supported Properties:**

testManagement.useOldExecutionView=true

Use this option to use the classic execution extension on Test Cases instead of the revamped execution view introduced in Polarion 18.

#### **Note**

#### Classic test execution view will become obsolete:

To give current testers some time to transition to the new Test Execution View workflow, we will still support the classic execution extension for a few more releases.

**Recommendation**: Get to know the improved workflow outlined in the Executing a Test Run Using Polarion section.

## **System maintenance**

### Low resource notifications

Polarion monitors Java system memory and disk storage resources and automatically notifies the Polarion administrator by email when on or both of these are approaching failure levels. The actual algorithm is fairly complex. The main points you need to know as an administrator are:

- A warning message is sent if the available free disk space is running low. By default, notifications are sent when free disk space:
  - is less than 5%
  - is less than 1 GB
  - less than 100 MB

The free disk space limit is configurable. See configure free space limit, below.

If Java is not able to recover enough heap memory during garbage collection, condition may indicate that Polarion is running out of memory, and the administrator may want to increase the -xmx setting in polarion.ini (Windows) or etc/config.sh (Linux) and restart the server.

Therefore, a warning message is sent when Java heap space is less than 5%.

#### If Polarion is installed as a service:

If Polarion installed as a service, as it is in the default Windows installation, you will also need to uninstall then reinstall the service.

#### Uninstall and reinstall the Polarion service.

- 1. Launch a command prompt window as an administrator.
- 2. Type cd Polarion\polarion and hit Enter
- 3. Type net stop polarion and hit Enter
- 4. Type service -uninstall and hit Enter
- 5. Type service -install and hit Enter
- 6. Type net start polarion and hit Enter

The Sending of these administrator notifications can be disabled by setting com.polarion.platform.monitoring.notifications.disabled=true in the system properties file polarion.properties (Follow the link if you need the location).

The email address(s) for the recipient(s) of these notifications can be set by specifying a comma-delimited list of email addresses in the system property:com.polarion.platform.monitoring.notifications.receivers

The sender can be set in the property com.polarion.platform.monitoring.notifications.sender, the email subject prefix can be set in com.polarion.platform.monitoring.notifications.subject.prefix.

#### Configure free space limit

The system property <code>com.polarion.platform.monitoring.notifications.freeSpacePercent</code> controls when notifications about low free disk space will be sent. Default value is 5 (i.e. 5%), meaning that when the free disk space on the disk where polarion data is stored is lower than 5% of the disk's capacity, notifications will be sent to the Polarion administrator.

It is not enough to simply set this property - monitoring of system resources must be configured as well. The following properties should be set:

```
com.polarion.platform.monitoring.notifications.disabled=false
com.polarion.platform.monitoring.notifications.receivers=<valid@email>
```

## Prevent server clogged with long-running transactions

Some processes, poorly optimized custom reports in particular, can run for hours and clog up valuable server resources. The problem is compounded when users get impatient and refresh the view in the hopes that it will speed up the process. At best, these multiple parallel runs slow down the server. At worst they can completely deplete the server's resources and result in an out of memory error that requires reindexing the server. To prevent clogging of the server, Polarion provides functionality that interrupts the processing of UI requests that take longer than a specified amount of time.

Only "read-only" user requests are affected. Requests with write operations are not.

The system properties described below prevent the clogging scenario by "killing" any user UI requests a few seconds before the connection timeout defined by the Apache configuration expires, and they display the following error:

Your request [id: REQ < NUMBER>] took too long to process and was stopped to avoid clogging the server.

Possible reasons it took so long: High server load, too much processed data or non-optimal scripts.

The interrupted request will also be logged in the log files. The following exception indicates such a request interruption: com.polarion.core.util.exceptions.OperationTimeoutException

The information in the exception and stack trace might indicate why the request is taking so long to process. Use the request identifier REQ <NUMBER> to match the request and log record.

### Applicable system properties

The following two system properties can be used to prevent long running transactions.

com.siemens.polarion.platform.threadKillingMode=disabled

#### Possible values:

- killAll: All long, read-only UI requests will be killed.
- reportOnly: Only requests related to Rich Text or Wiki Pages (read-only) will be killed.
- disabled: Only logs the long-running transactions and doesn't interrupt them. (The default value.)
- com.siemens.polarion.platform.threadKillingTimeout=590000

Sets the time period (in milliseconds) after which a request can potentially be killed.

The default value is 590000 (9 minutes, 50 seconds: 10 seconds before the default Apache timeout). If the value is 0 or less than 0, the default value will be used.

- Set this property with a time period *less than* the Apache ProxyPass timeout value configured in the polarion.conf or polarion-cluster.conf files.
- UI requests that make any write operation are marked as *non-interruptible* and will continue to be processed in the background until they are complete.
- Custom extensions invoked by UI requests that operate outside the Polarion API may also be considered *interruptible*.

(If you think that they might exceed the set timeout period and be affected by a termination of the request, then disable this feature or call the preventThreadKilling() method from the ITransactionService class within the custom extension.)

### **Activities index cleanup**

Administrators can schedule an automated cleanup of the activity stream storage area.

The default setting runs the Activities Index Cleanup on a weekly basis.

(Every Sunday at Midnight.)

### Benefits:

- The bigger the index, the slower it is to read and write to it.
- In a clustered environment, the index is located in a shared storage environment. If never cleaned, it will grow indefinitely.
- In a single-instance environment, the activities index is cleaned with every reindex.

### **Configure Activity Cleaner:**

By default Activity Cleaner job keeps a maximum of **5000 activities** that are not older than **90 days** for each source.

These limits can be configured granularly per source and type by editing the following lines in the **polarion.properties** file.

```
In (C:\Polarion\polarion\configuration\) by default.
    maxActivitiesLimit=5000
```

maxActivitiesLimit = The number of activities that are kept by Activity Cleaner.

(Can be set individually for particular sources and types, or globally for all sources.)

If no property is specified, the limit is controlled **per source by default**.

### A search for the value is done in following order:

1. **Source.type** system property:

```
com.polarion.activity.(sourceID).(type).maxActivitiesLimit
```

### 2. **Source** system property:

```
com.polarion.activity.(sourceID).maxActivitiesLimit
```

## 3. The system property for all **Sources and Types**:

```
com.polarion.activity.maxActivitiesLimit
```

### 4. The default Source/Type limit:

```
Returned by; IActivitySource.getHistoryLimitCount(type)
maxActivitiesAge=90
```

maxActivitiesAge = How long, in days, that activities are preserved by the Activity Cleaner.

(Can also be set individually for particular sources and types, or globally for all sources.)

If no property is specified, the duration is controlled **per source by default**.

### A search for the value is done in following order:

1. **'Source.type'** System Property:

```
com.polarion.activity.(sourceID).(type).maxActivitiesAge
```

2. 'Source' System Property:

```
com.polarion.activity.(sourceID).maxActivitiesAge
```

3. The System Property for all **Sources and Types**:

```
com.polarion.activity.maxActivitiesAge
```

### 4. The default Source/Type limit:

```
Returned by; IActivitySource.getHistoryLimitDays(type)
```

See Configure the Scheduler for more information on scheduling maintenance jobs.

## Configure the Scheduler

Polarion comes with a default set of scheduled jobs. Scheduler allows you to configure scheduled jobs such as builds, dashboard updates, system cleanup, etc. You can alter the properties of scheduled jobs, such as the day of the week when the job should run. You can also add additional scheduled jobs to the default set, or remove jobs from the default jobs set. You can optionally invoke any job explicitly in the **Monitor** topic.

#### **Define and modify Jobs**

This configuration is available only for the global (repository) scope. If you are not familiar with the basics of the different scopes, you may want to review: The Administration Interface. The configuration is specified in the OScheduler topic of Administration.

Reference information including cron expressions and examples for configuring jobs is embedded on the Scheduler configuration page (scroll down if necessary to see it). The same information is available in the Administration Reference: Scheduler.

### Access the Scheduler configuration:

- 1. Log in with administration rights for the repository.
- 2. Enter the Administration Interface (click the \*Administration) link in the tool view of Navigation).
- 3. Select Repository in the **Open Project or Project Group** dialog.

```
(See: Access Projects.)
```

4. In the Navigation pane, select the OScheduler topic.

The **Scheduled Jobs Configuration** page loads in your browser. This page has an embedded XML editor which displays the XML code of the current configuration.

- 5. Edit the XML code as needed to make the changes or additions you want. Refer to the embedded help text that appears below the embedded XML editor.
- 6. Click the **Save** button to save the changes to the Scheduler configuration. Changes are immediately in effect it is not necessary to restart the server.

### Spawn multiple jobs

Jobs can spawn other jobs. The ID for such a job is multi.job. Jobs to spawn are referenced by their names from schedule.xml. It is possible to run jobs either sequentially (default) or in parallel, and possible to terminate sequential execution if a job fails. Default is to ignore the failure. The job reports failure if at least one of the executed jobs fails.

The following listing shows the multi.job scheduler configuration:

It is possible to fail the job if at least one of the executed jobs fails. Default is to finish successfully no matter what. Aborting will always fail the multi.job.

### **Extended configurability:**

- JOB NAME may contain execution expression. Execution expression consists of:
  - o Job name
  - Sequential operator ,

- Parallelization operator |
- Operator precedence is (from highest to lowest): |,
- Seguential operator will execute job groups in seguence. Failures coming from these jobs are ignored and not delegated.
- Parallelization operator will execute job groups in parallel. Failures coming from these jobs are ignored and not delegated.

Example: J1, J2 | J3 | J4, J5 will execute three groups of jobs in parallel, first group consists of sequence J1,J2, the second group is a single job J3, and the last group is a sequence of J4 and J5.

#### Jobs in a Clustered setup

Scheduled jobs should be reviewed for a cluster, and convenient Node selectors (i.e. the **node** attribute of <job> elements) should be specified depending on the nature of the job. The following default jobs should have node="\*" specified for them in a clustered setup:

- Index Checker
- Suspend DB History Creator
- Resume DB History Creator

#### Run scripts via Jobs

You can define a job that runs a custom script. The job ID in the scheduler must be script.job. Groovy (groovy) and JavaScript/ECMAScript (js) scripts are supported. Script files must be stored in the **scripts** folder of the installation's file system.

The job has the following required parameters:

- scriptName: Name of the script to run
- scriptEngine: Name of the scripting engine to use (groovy or is)

In the script you will be able to access following variables:

- logger: A com.polarion.platform.jobs.lLogger you can use to log messages for the job
- scope: A com.polarion.platform.context.lContext that repesents the scope of the job.
- workDir: A java.io. File pointing to the working directory of the job.
- jobUnit: The com.polarion.platform.jobs.lJobUnit that runs the script.

Every property is defined inside the properties> element of the job. The property in the example below would be accessible as variable with the name "myProperty" containing the value "myPropertyValue".

```
<job name="Script Job" id="script.job" cronExpression=""</pre>
disabled="true" scope="system">
    <scriptName>test.js</scriptName>
    <scriptEngine>js</scriptEngine>
    properties>
```

### Run scripts directly:

The internal script engine that executes scripts via jobs, and also script-based workflow conditions and functions for Work Items and Documents (ScriptCondition, ScriptFunction) can run a script without installing it into the scripts folder. To run a script directly, direct your web browser to:

[POLARION SERVER]/polarion/scripting (where [POLARION SERVER] is replaced with the actual domain of your server).

The page provides an interface that you can use to input and run a script. The variables visible to the script are the same as for script jobs. (Actually, the script is run **as** a job). The system property <code>com.polarion.scripting.servlet.enabled</code> must be enabled in the system configuration file polarion.properties in order to run scripts directly. By default, only administrators are allowed to run scripts this way, as it is potentially dangerous.

### Start and stop the Scheduler

As mentioned, Jobs are defined (scheduled) and run in the global scope. You can start or stop the Scheduler from the Scheduler topic in Administration.

### To invoke one of these operations:

- 1. Log in to Polarion with administrator permissions for the repository.
- 2. Enter the Administration Interface.
- 3. Select **Repository** in the **Open Project or Project Group** dialog to work in the global scope.

```
(See: Access Projects.)
```

A button appears at the top of the Scheduler page, just under the main heading. If the Scheduler is currently running the button label is **Stop Scheduler**. If the Scheduler is currently stopped, the button label is **Start Scheduler**.

4. Click the button for the operation you want to invoke. When the Scheduler is stopped, configured scheduled jobs will not run. When the Scheduler is started (i.e. running), the configured jobs will be run according to the configured schedule.

### Schedule Jobs to run immediately

You can run any scheduled job immediately from the Monitor topic. The Scheduler must be running. (See Starting and Stopping the Scheduler.) Check the job(s) you want to run in the **Scheduled Jobs** section and click the **Execute Now** button to launch the job(s) you selected.

#### **Note**

Running jobs uses server system resources, and some types of jobs can be very resource intensive. Administrators generally schedule jobs to run at times when user demand is low. Be aware that explicit running of jobs has the potential to degrade performance for end users and use discretion accordingly.

### **Execute system commands and utilities:**

You can use the **execute.command** job to immediately run some system command or utility. For example, the following will run the PDT diagnostic utility.

```
<job id="execute.command" name="PDT" scope="system">
    <command>$[com.polarion.home]/diagtool/run$[shellExtension]/command>
    <arguments>
        <argument>-sendResults</argument>
        <argument>no</argument>
        <argument>-resultsDir</argument>
        <argument>$[jobDir]</argument>
    </arguments>
</job>
```

### Disable execute.command Job execution:

Administrators can add com.siemens.polarion.security.executeCommandJob.enabled=false to the polarion.properties file to disable all execute.command jobs for increased security.

When set to False, users will not be able to execute execute.command jobs and will be prompted with a security warning if they attempt to do so.

#### **Note**

The Polarion Diagnostic Tool is also an execute.command job and will also be disabled.

#### View Jobs in the Monitor

You can see the status of all Jobs in the **Monitor** topic. This topic is available in the **Navigation** pane when you exit from **Administration** and return to project or portal content topics.

#### To access the Monitor:

- 1. In the Open Project or Project Group dialog, select either Repository or a Project (you must have the relevant permissions).
- 2. If you are in **Administration**, click the link in the top panel of **Navigation** to return to the portal Home or the project.
- 3. In the **Navigation** pane, select **Monitor**.

### Clean up temporary folders

Jobs create temporary folders, which in turn store job logs. At some point, you will want to clean up these folders. There is a job jobs.cleanup that deletes these temporary folders and their data. The job can be scheduled to run periodically, or launched explicitly. You can optionally use the preserveAgeHours to specify how long to preserve temporary file after jobs start. The value of the parameter is hours (see the following example). If the parameter is not specified, the job defaults to 24 hours.

### Example:

```
<job cronExpression="0 55 * ? * *" id="jobs.cleanup"</pre>
name="Cleanup of Temporary Files" scope="system">
```

In this example, temporary folders and data are preserved for 36 hours.

### **Backup Polarion data**

This section discusses the system location of data that administrators should be sure to include in regular backup operations.

#### References to installation locations.

For the purposes of discussing the location of various folders that administrators may wish to include in backups, this section will define some variables and their meaning on different operating systems. This is because the installation structure differs somewhat on Windows vs. Linux.

#### Variables and equivalent paths:

This section documents the system variables and their locations of different operating systems.

Variable: POLARION HOME

Description: Location of Polarion binaries.

Windows: C:\Polarion

• Linux: /opt/polarion

### **Note**

The above paths are the defaults. You can configure the root location during installation.

Variable: polarion\_data

Description: Root for Polarion data files.

Windows: %POLARION HOME%\data

• Linux: /var/lib/polarion

#### **Note**

The above paths are the defaults. You can configure the data files location during installation.

### **Subversion repository**

An essential component to back up is the Subversion repository, located in <code>%POLARION\_DATA%/svn</code>. The default storage method is file-based and so does not modify existing files, only adds new ones. This is very convenient for incremental back-up methods.

#### **Project builds**

Project builds are located in %POLARION DATA%/bir/projects.

The contents can be very large. Developers and project managers should know what is most critical to back up. It is usually not necessary to back up all the nightly builds for example. It also makes sense not to back up the demo project builds (installed by default by the Windows installer program).

#### Reports

Reports are generally not critical to back up because they can be recreated any time, with one **exception:** the trends.

To back up the trends you need to collect all \*trend\*.xml files (e.g. workitems-trend-data.xml) in the structure beneath the folder %POLARION DATA%/RR. Backing up the entire folder will most likely be a significant waste of backup space (up to 99%).

If you use SCRUM calculation, you should back up scrum-data.xml because it holds the historic values for all past sprints.

#### **User-modified data**

The following data may or may not need to be backed up depending on whether it was modified after Polarion was installed.

### Polarion configuration file:

It may make sense to back up the root Polarion configuration file polarion properties (follow link for location). This file can be recreated by reinstalling Polarion and filling in all the original arguments. However, if the file was modified after the installation to customize the system configuration, it is a good idea to back it up.

### Modified installation structure:

If the original folder structure of the installation was modified - to add more Maven plugins or Apache modules, or if some graphics or templates were modified - it is a good idea to back up such modifications in case it is ever necessary to reinstall Polarion (on new hardware, for example).

#### Connectors data

If any Connectors are configured for any projects (in Administration Connectors), include the folder %POLARION DATA%/synchronizer in your backups. The folder contains a database that stores the relation between Work Items and the items in the connected third-party systems, and also baseline data that is used to detect which fields were changed. Losing that data would mean losing the relation between Work Items and the items in the other systems, and when the items are synchronized they will be recreated instead of updated, thereby corrupting the data integrity of both systems.

### Backup tips for the Polarion server

From best to worst scenario:

- 1. Do a snapshot of suspended virtual machine in which Polarion server runs (in case virtual machines like VMWare are used).
- 2. Stop Polarion server, do the backup, run server again.

- 3. On Windows, do not use backup software which opens files in write mode, because that may cause corruption of index-related files and would require a re-index operation afterwards)
- 4. If running on Windows and backup software does open files in write mode, configure the job "Live Plan Refresh" not to run during the backup (this job is most likely to work with the index and stands the greatest chance of index corruption resulting)

When doing recovery from a backup of the running system (i.e., not a snapshot of the virtual machine, or the server was not stopped during the backup), it will most probably be necessary to run a re-index operation.

#### **Index and Reindex**

Polarion stores all data in the Subversion repository. While this approach brings many advantages (history, branching, etc.), compared to traditional database approach it does entail some performance trade-offs. Polarion uses advanced caching and indexing techniques to achieve acceptable performance levels. In most standard situations, the caching and indexing behaves transparently and users do not need to take care of it. However, the caching and indexing logic imposes certain restrictions and requirements for what users can and cannot do. This section explains these limitations in more detail.

### **Index Synchronization**

The following events initiate index updating:

- Commits to the Subversion repository (including changes initiated by Polarion).
- Polarion start-up

The index contains information about location of all entities (Work Items, etc.) in the repository and their history, so the term "index" refers to more things then just the index used by the Search feature. Rather it is the entire indexing scheme that enables optimal performance.

During the index update the Polarion checks all changes made to repository since the last indexed revision and update the index accordingly.

Changing sort order of an enumeration, dates of time points, etc. which are already referenced will cause inconsistency in the sorting. Some referrals will be sorted by to the old sort order, and some by the new. A complete index rebuild solves this problem.

#### **Note**

If all you need to do is refresh lists of enumerations because of changes to some enumeration(s), you do not necessarily need to run a full re-index. For more information see: Enumerations and Database Issues.

#### When to repair the index:

Symptoms of a "broken" index tend to be the following: some Work Items are not up-to-date, some are missing, history cannot be viewed or shows invalid or truncated information.

#### First remedies:

You may not need to completely rebuild the index if you observe the above symptoms.

- 1. Try an index refresh on the repository, or on a specific project. This corrects problems with information updating.
  - Go to the Administration interface and open the Repository (if you want to refresh the entire repository), or a project (if the problem seems to affect only that project). Then, in Navigation, select Maintenance. In the Index section, check the index(s) to refresh and click Refresh Index. (Note that the indexes listed varies depending on whether you are working in Repository or project scope).
- 2. If number 1 above does not fix all symptoms, try running the Check Index button. The index check job should bring back Work Items that are missing from view.

If the above steps do not fix the symptoms, then you will need to rebuild the index, as described in the next section.

#### How to rebuild in index:

The procedure may sometimes be referred to as "index rebuild", "rebuild index", "index synchronization", or "index rebuilding".

There are two important things for an administrator to keep in mind before initiating an index rebuild:

- The Polarion server must be shut down before commencing an index rebuild.
- Depending on the size of the repository, reindexing time can range from a few minutes to several hours. End users will not be able to use the system during the reindexing process.

If possible, you should plan to perform any re-index operation at a time when system usage is low, and downtime is not critical for end users.

### **Configure CPU usage for indexing:**

It is possible to configure how many CPU cores ("workers") are used in these phases of the indexing process. There is one basic system property - com.polarion.startup.workers - that can be configured to define total number of workers for all phases of reindex that can run in parallel. Administrators can also specify different numbers for individual phases in instances of the system property com.polarion.startup.workers.phase#, where # can be any of the following:

- **3** For Context Initialization (phase 3),
- 7 For Data Indexing (phase 7),
- 8 For calculating calculated fields (phase 8),
- 10 For Attachment indexing (phase 10 AttachmentIndexer job),
- 11 For History Indexing (phase 11 DBHistoryCrator job).

By default, the number of CPU cores used for the above phases is total cores, divided by 2, plus 1. For example, on a CPU with 4 cores, the default usage is (4/2) + 1 = 3 cores.

### Maximum workers for all phases is limited to 8:

In the default setup, the jobs are configured to run only between 19:00 and 08:00 (and they can be stopped and started by scheduled jobs) so in most cases there should be no need to redefine the

number of workers. Adjusting the number of workers may be worth doing for the jobs running after Polarion start-up that may cause performance to slow.

To explicitly set core usage for all index phases and/or the configurable index phases listed above, you must add the following lines to the system properties file polarion.properties:

Total workers for all phases:

com.polarion.startup.workers=[N] (where [N] is an integer value that should not exceed the total available CPU cores).

Example: com.polarion.startup.workers=8

Workers for specific phases:

com.polarion.startup.workers.phase#=[N] (where # is the phase, and [N] is an integer value specifying the number of workers to be used for the specified phase.

### **Examples:**

- 0 com.polarion.startup.workers.phase3=2
- 0 com.polarion.startup.workers.phase10=3

#### Note

The location of the polarion.properties file can be found in: System Properties File Location.

### Launch the index rebuild scripts:

You perform the reindexing by running the appropriate reindexing script for your operating system:

- Windows: C:\Polarion\polarion\reindex.bat,
- Linux: \$POLARION HOME\$/bin/polarion.init reindex,

Progress of the reindexing operation is reported in the console and written to a log file.

### Warning

Do not interrupt the index rebuild process. The process writes messages to the console and the <a href="log4j-startup-TIMESTAMP.log">log4j-startup-TIMESTAMP.log</a> file. Check this file for the status of the process. As mentioned, with large repositories the index rebuild can take several hours, and it may seem that nothing is happening for long periods.

Reindex must be restarted if interrupted. If you believe there may be a problem with the process, contact Polarion technical support before attempting to kill the reindex rebuild process.

### **Understanding the Re-index process:**

The reindexing scripts perform the following operations:

1. Back up of the polarion-data folder. (The folder is renamed to polarion-data.saved.)

(Path: \$POLARION HOME\$/data/workspace/polarion-data.)

- 2. Start the Polarion service. (As mentioned previously, the service should be shut down before running the script.)
- 3. Regenerate the index in the polarion-data folder. As previously noted, this can take considerable time during which users will not be able to access the system.

### **User impact:**

Users cannot use the system while the system indexes are being rebuilt. They can, however, use the system while the historical database is being reindexed.

The embedded historical SQL database, which replicates the Subversion repository to facilitate complex gueries, is indexed in the background. Although users can use the system, some features are either not available, or do not display information until the database reindex is complete. A message is displayed at the top of the Navigation panel warning users that database indexing is in progress. When the message is present...

- Baseline (Time Machine) feature is not available.
- Reports that contain SQL queries on the historical database do not work and display error messages.
- Burn-up and burn-down charts in Plans do not work and display error messages.
- Test management form extensions do not work.

## **Use the Polarion Diagnostic Tool**

Polarion comes with a self-diagnostic utility, Polarion Diagnostic Tool (PDT). If your system is not performing as it should, you can use this tool to run comprehensive diagnostic tests and communicate the results to the technical support team by email or FTP upload. PDT checks if Polarion is running in a cluster and gathers configurations from shared folders.

The utility is bundled into all product distributions. It is located in the diagtool folder under the root of your Polarion installation. Complete documentation (diagtool.pdf) covering setting up and running the PDT utility is included in that folder.

### Warning

If administrators disable all execute.command jobs the Polarion Diagnostic Tool (PDT) is also disabled.

## Recover disk space

Over time your server's disk storage can accumulate a large amount of data that does not need to be kept. For example, if your process involves nightly or other frequent builds, these can eventually consume a lot of disk storage, but the don't really need to be kept indefinitely. Old log files can also consume disk space. This section discusses ways to clean up your server's disk storage and recover disk space.

### Clean up old Builds

Polarion comes pre-configured with the build.clean job for cleaning up builds and related things like local deployment space.

To have the job delete the local deployment space directory, set the <code>localDeploymentSpaceName</code> to the name of the local deployment space. For example:

To have old builds deleted from BIR, three job parameters must be set:

- maxBuildAgeDays,
- minSuccessfulBuildsCount,
- buildArtifacts.

The job deletes builds of all build artifacts that match at least one of the selectors from buildArtifacts, but only those builds that were started at least before maxBuildAgeDaysdays. At least the number of builds specified in minSuccessfulBuildsCount are preserved provided at least that number of successful builds exist (i.e. builds with status ok). Build selector format is GROUP\_ID:ARTIFACT\_ID, where each part can include a suffix \* matching any string. For example:

See also: Configure the Scheduler.

#### Clean up old log files

You can configure and run the **logs.cleanup** job to delete log files with specified names that are older than a specified date. (See Configure the Scheduler for more information on working with scheduled jobs.)

#### Scheduler configuration:

The following example shows the basic structure of the log cleanup job configuration.

cronExpression= Lets you schedule an automatic cleanup of old log files.

If the cronExpression="" field is left empty and/or disabled= set to "true", log cleanups will need to be triggered manually.

(Via the OScheduler.)

Cron Expression Example: "0 15 01 ? \* SAT" values would schedule the log cleanup every Saturday at 1:15 am.)

(Recommendation: Larger enterprise organizations should also consider using an external central monitoring application for saving and archiving logs.)

- **FOLDER** is where to find log files,
- **\_PATTERN**\_ is a file name prefix regular expression,
- **\_FOLDER\_** and **\_PATTERN\_** are concatenated with | (pipe),

File name prefix is matched against all files in the folder, those matching the same prefix are treated together, From this set, all files older than MAX AGE OF FILE IN DAYS are removed with the exception of the newest one which is always kept.

Example file pattern #1: C:/Polarion/data/logs/main|log4j-[^2]\*2

#### Job result:

- C:/Polarion/data/logs/main/log4j-20090525-1057-23.log (will be kept),
- C:/Polarion/data/logs/main/log4j-20090523-0457-87.log (will be deleted if older than specified number of days),
- C:/Polarion/data/logs/main/log4j-20090525-1057-23.log.2009-12 (will be deleted if older than specified number of days),
- C:/Polarion/data/logs/main/log4j-startup-20090525-1057-23.log (will be kept),
- C:/Polarion/data/logs/main/log4j-startup-20090525-1057-23.log.2009-12 (will be deleted if older than specified number of days),

Example file pattern #2: C:/Polarion/data/logs/apache|.\*\.log,

#### Job result:

- C:/Polarion/data/logs/apache/access.log.2009-05-25 (will be kept),
- C:/Polarion/data/logs/apache/access.log.2009-05-20 (will be deleted if older than specified number of days),
- C:/Polarion/data/logs/apache/error.log.2009-05-20 (will be kept).

See also: Configure the Scheduler.

### **Maintain Subversion (SVN)**

This section provides guidance for administrators on maintaining and updating the Subversion component of the Polarion system to ensure optimal performance. The information here is mainly about repository format and protocol. Once you have these in place, see: Optimize Subversion.

### Use the latest repository format

Use latest repository format with default configuration for best performance. To migrate the SVN repository to a new format:

- 1. Dump repository via command: svnadmin dump -M 1024 repository\_path > dump\_file.out
  - The in-memory cache (-M) option was introduced only in SVN 1.9.
  - This operation may take hours and you need to make sure that all revisions will be dumped, either by taking your SVN offline, or doing incremental dump and load when switching the repository.
- 2. Create new repository in new storage format via command synadmin create repo path.
  - By default this creates new repository storage in newest format with default (and recommended) settings
  - It is advisable to secure access to repository via the configuration in repo path/conf/synserve.conf.
  - Ensure that there are the same permissions and owners set for the new repository as for the old one.
- 3. Load the old repository contents from the dump to the new repository via command: svnadmin load -M 1024 repo\_path < dump\_file.out
  - The in-memory cache (-M) option was introduced only in SVN 1.9.
  - This operation may take days for large repositories with hundreds of thousands of revisions, so make sure to utilize the in-memory cache and fast storage (SSD) for this operation.
  - Doing incremental dump and load when switching the repository can help you to limit the downtime.
  - Pack the repository after import.

### Pack the Repository

To be sure that caches are optimally utilized, your should pack the repository regularly. Packing weekly is a good interval for many installations.

To pack the repository, execute via crond: svnadmin pack repository path.

### **Advanced administration**

## System and virtual machine memory

Issues around server memory allocation and Java virtual machine are covered in the PDF installation guide documents bundled with Polarion distributions, and also available on SIEMENS' Doc Center.

- **Windows Installation Guide**
- Linux Installation Guide
- Polarion Enterprise Setup Guide (for users running Polarion on multiple physical and/or virtual servers with one or more Polarion repositories in the topography.)

Administrators of Windows installations should follow the recommendations in the section Adjusting Server Memory Allocation in the Windows Installation Guide.

Administrators of Windows or Linux installations should note the section Java Virtual Machine **Memory Limit** in the installation guide for their respective OS.

Administrators of clustered server installations should note the recommendations in the installation guide for the operating system on their servers, well as any relevant instruction in the Polarion **Enterprise Setup Guide.** 

The information in these documents can help avoid issues such as startup failure (where the Polarion service does not start) and Java Out of Memory (OOM) errors.

## Optimize the PostgreSQL database

Beginning with version 2015 SR2, Polarion integrates the PostgreSQL database in all new installations. This system component is used to improve performance on complex system queries. Automated installers for Linux and Windows attempt to modify default PostgreSQL settings to optimize it for operation with Polarion. Assuming the process succeeds, PostgreSQL should perform adequately in most cases. An administrator may optionally adjust the settings described later in this section to better match the specific environment.

If PostgreSQL optimization fails during automated installation, or if Polarion is installed manually (as can be the case with Linux in particular), the administrator should modify PostgreSQL settings as described below to optimize it for Polarion.

The PosgreSQL configuration file is named postgresql.conf, located on path /opt/polarion/data/postgres-data/ on Linux, or C:\Polarion\data\postgres-data\postgresql.conf on Windows. Open the file in any text editor and set the properties as shown in the following listing:

```
max connections = 80 # should be less than 10 * number of CPUs
shared buffers = 2GB # should be 10% - 15% of total system RAM
work mem = 10MB # should be 10MB - 100MB
maintenance work mem = 200MB
fsync = off
synchronous commit = off
full page writes = off
wal buffers = 256kB  # should be more than size of common transaction
```

```
checkpoint_segments = 32

effective_cache_size = 4GB  # should be approx 1/3 of RAM for PostgreSQL

max_locks_per_transaction = 100  # specific for Polarion

# Optimal planner performance setting

# For HDD, keep default setting. Otherwise, uncomment the applicable setting below:

# For SSD:

# random_page_cost = 1.5

# For SAN:

# random_page_cost = 2.0
```

After saving the changes, restart PostgreSQL.

### Warning

On Linux systems, check that the value of the <code>shared\_buffers</code> parameter in <code>postgresql.conf</code> does not exceed the available shared memory configured by the kernel parameter SHMMAX. Adjust <code>shared\_buffers</code> as necessary, keeping in mind the recommendation in the listing above.

#### **Prevent excessive Test Run records**

If you have very large manual Test Runs, the default system configuration can result in excessive numbers of Test Run records in the historical database, as it uses the default referencing of structures via foreign key. You can remedy this situation by setting the system property com.polarion.platform.sql.useNewStructTables=true in the polarion.properties file. Setting this property ensures that artificial primary keys (PK) are generated for Test Run records, resulting in far fewer records in the historical database. After setting this property it is necessary to re-index the system, which will rebuild the database.

#### **Enumerations in the Polarion database**

By default the database table containing the enumeration options is not populated because this data rarely needs to be obtained from the database. However there can be times when the complete enumeration options definition is required in the database. (Like when the database is the primary source of this data, for example, if you are building an external report on top of the Polarion database using the The Business Intelligence and Reporting Tool (BIRT) extension.)

If this is the case, you may not only need to load the unique IDs stored with the objects, but also the names of the enumeration options.

To populate the enumeration options table in the Polarion databases, set the following property in the polarion.properties file:

```
com.polarion.platform.sql.populateEnumOptionsTable=true
```

Once this property is enabled, a job that populates the enumeration options table, in both the HEAD and historical database, is automatically run after each reindex.

#### Keep the Enumeration table up to date:

Unlike Work Items or LiveDocs, the enumeration options are not automatically updated in the database table when there's a configuration change.

To ensure the consistency of the enum options table with the actual configuration, the **Refresh** Database button triggers a job that re-synchronizes a particular enumerations configuration to the database.

### **Periodically refresh Enumerations:**

You can schedule the polarion.jobs.refresh.enumeration job to run regularly as a part of your system maintenance. When it is run without any additional parameters, all enumerations are refreshed. By default, the polarion.jobs.refresh.enumeration job only starts once after a reindex.

For more information on scheduled jobs, see: Configuring the Scheduler.

### **Object-based Enumerations:**

Object-based enumerations are not populated in the enum table and if required, must be collected from other tables.

For additional information, see: History Indexing Job and User Impact.

## Speed up the movement of Work Items to and from Documents

Enable In-memory Object Maps to reduce the wait time while moving Work Items to and from Documents by 95%.

Doing so will increase the amount of RAM used and the size of the object-maps folder.

(RAM usage will vary but will be about 3x the size of the object-maps folder.)

#### How it works

It optimizes Object Map implementation to minimize disk access. It is based on a map kept in memory and saved to disk asynchronously.

Saves are made each time the map is asked to do so. (Provided no additional save requests have been made for 10 seconds.)

While saving, all data is flushed onto the disk and data in memory can still be modified.

### **Enable in-memory Object Maps**

To enable **In-memory Object Maps**, add the following property to the polarion properties file:

com.siemens.polarion.persistence.objectMaps.inMemory=true

#### Warning

- Any time the "True/False" value is changed a reindex must be performed.
- A reindex is also required if Polarion is shutdown incorrectly.

#### Reduce the size of the Object Map folder

On some production systems, depending on the configuration, the **Object Map** folder can get quite large (10+ gigabytes).

A **Defragment object maps** job can be run that greatly reduces the size of this folder.

Copy the sample script below to the O Scheduler.

### Warning

Set it to run during off-hours:

Some resources might be temporarily locked while the defragmentation job is run. To ensure that users are not affected, run the **Defragment object maps** job over night on production systems.

### Sample defragment object map script:

```
<job cronExpression="0 0 6 ? * *" disabled="false" id="object.maps.defragmentation"
name="Defragmentation of object maps" scope="system"/>
```

The sample script above will run the **Defragment object maps** job everyday at 6 a.m.

### **Database compatibility (H2, PostgreSQL)**

Polarion replaced the integrated H2 database with PostgreSQL - in new installations only from version 2015 SR2, and in existing installations from version 2016. The database is used for complex query processing, such as are often needed for reports. Users of prior versions may have some report queries formulated for H2 that do not work with PostgreSQL due to its more strictly ANSI-compliant syntax. To mitigate such problems, Polarion provides a database compatibility mode.

Compatibility mode is enabled via the system property com.polarion.platform.sql.h2CompatibilityMode. When set to *true*, Polarion attempts to adjust queries formulated with H2 syntax "on the fly" so that they work with PostgreSQL. Compatibility mode handles situations where a problematic query cannot be updated to be PostgreSQL compliant - reports in History and Baselines that fail due to missing or incompatible functions, for example.

If, after your system is updated to use PostgreSQL, you find that some queries are not working, enable database compatibility mode. Open the system configuration file polarion.properties in a text editor and make sure the following is present: <code>com.polarion.platform.sql.h2CompatibilityMode=true</code>. Restart the Polarion server to effect the change.

Issues handled in compatibility mode include:

- Error: Subquery in FROM must have an alias
- Usage of H2-specific functions that result in errors such as: function curdate() does not exist.

Even when database compatibility mode is enabled, it is still possible that some reports may need to have their queries modified manually.

### Map H2 Functions to PostgreSQL

This section explains how to map queries that use H2-specific functions to PostgreSQL. (You can find a listing of all H2 functions at:http://www.h2database.com/html/functions.html.)

Here is an example of an error, thrown when the proper *curdate* function is not defined: "ERROR:

function CURDATE() does not exist Hint: No function matches the given name and argument types. You might need to add explicit type casts."

#### To map a function from H2 to PostgreSQL:

- 1. In your installation's file system, navigate to [POLARION HOME]/polarion/plugins/com.polarion.platform.sql x (Where "x" is the version number.)
- 2. Copy the file h2 functions mapping.sql, paste it to folder [POLARION HOME]/polarion/configuration, and open it using a text or SQL editor application.
- 3. If the function does not already exist in this file, add your own definition of the function from H2. For example:

```
CREATE OR REPLACE FUNCTION POLARION.CURDATE()
returns date AS
$$ SELECT current date $$ LANGUAGE SQL;
```

Function definitions must always end with a semicolon (;) to be successfully deployed, and always prefix the names of mapped functions with the schema name polarion, otherwise Polarion will not always take them into consideration.

- 4. Enable the property com.polarion.platform.sql.h2CompatibilityMode=true in the polarion.properties configuration file.
- 5. Restart the Polarion server.

#### Tip

For easy debugging before deploying a function to the h2 functions mapping.sql file, try to manually create it in the PostgreSQL database.

Use the psql utility to connect to the polarion database as the polarion user: psql -p 5433 -U polarion polarion.

For information on creating PostgreSQL functions, see http://www.postgresql.org/docs/current/static/sql-createfunction.html. To check the definition of a function had in H2, see <a href="http://www.h2database.com/html/functions.html">http://www.h2database.com/html/functions.html</a>.

## PostgreSQL database Start-Stop-Restart

Polarion includes the embedded PostgreSQL database to process complex gueries more efficiently than is possible with the Lucene query engine. There may be times when an administrator needs to start, stop, or restart the database.

On Windows, shortcuts are provided in the folder [POLARION HOME] \polarion shortcuts. On Linux, the following scripts are available:

#### CentOS, RHEL, Debian or Ubuntu

service postgresql-polarion start

```
service postgresql-polarion status
service postgresql-polarion status
service postgresql-polarion restart
```

#### SUSE/SLES 11

```
rcpostgresql-polarion start
rcpostgresql-polarion stop
rcpostgresql-polarion status
rcpostgresql-polarion restart
```

#### SUSE/SLES 12

```
service postgresql-polarion start
service postgresql-polarion stop
service postgresql-polarion status
service postgresql-polarion restart
```

### **Optimize Subversion**

Polarion uses a Subversion (SVN) repository as its main data storage. This section provides guidance for administrators about optimizing Subversion for best performance. Some information is specific to Windows or Linux, and some is specific for a Subversion version currently running in a Polarion installation. Before implementing any of the optimizations below, be sure to see: Maintain Subversion.

### Subversion 1.7 and 1.8 cache memory

**Windows:** Both Apache and synserve use threads to serve incoming connections, so the allocated cache is shared among all connections. You should:

- Assign at least 256 MB cache allocation via SVNInMemoryCacheSize 262144 for Apache.
- Assign at least 256 MB cache allocation via -M 256 for synserve.

**Linux:** By default, both Apache and synserve use new processes to serve incoming connections, so the allocated cache is specific for every connection. Make sure you have enough memory for recommended cache size multiplied by the anticipated number of parallel connections to SVN. For example, given a recommend cache of at least 128 MB as synserve cache, and if there are 10 parallel requests to SVN at the same time, at that moment the processes consume 10 \* 128 MB memory, or 1280 MB. Review your cache memory to be sure the assigned cache for SVN will not consume too much memory when parallel connections are made.

- Assign at least 128 MB cache allocation via SVNInMemoryCacheSize 131072 for Apache.
- Assign at least 128 MB cache allocation via -M 128 for synserve.

Because a process is spawned for each connection and there might be dozens of such connections with highly concurrent use of Polarion, the total amount of memory occupied by synserve may be high. The following external resource may be helpful: http://svnbook.red-bean.com/en/1.8/svn.serverconfig.optimization.html

#### General recommendation for Subversion 1.7+ on any environment:

Use MaxKeepAliveRequests 10000 for Apache's mod\_dav\_svn.

# **Subversion 1.8 with 18.8 repository format:**

Utilize the revision properties packing feature with the related cache settings below:

- Enable caching of revision properties via SVNCacheRevProps on for Apache.
- Enable caching of revision properties via " --cache-revprops yes" for synserve.

#### Subversion 1.9

- Use dynamic cache allocation via SVNInMemoryCacheSize 0 in Apache.
- Use dynamic cache allocation via -M 0 in synserve.

# **SVN** security when using synserve

Make the following adjustments to enhance security of any server running synserve:

- Limit access to the synserve server to only those machines that actually read the data. Use parameter --listen-host 127.0.0.1 for the synserve server.
- Set -R for synserve to open access for read-only
- Configure repo path/conf/synserve.conf to disable anonymous access to the repository via the svn protocol and force authentication through the passwd file for read access.

#### Subversion access by system user

On Polarion installations with large amounts of content, there can be significant performance benefits from setting up alternate access to the Subversion repository for the system user, especially for system indexing/reindexing.

The system property reposystem can be used to provide an alternative repository URL to be used for read-only access by the system user. If this property is not set, the system user uses the normal access URL as specified by the repo system property.

Either of 2 protocols can be used when specifying the URL value in the reposystem property:

- The file:// protocol can be used when the SVN repository resides on the server computer's local drive. This is the simplest alternative, as nothing other than the reposystem property needs to be configured. See warning below.
- The svn:// protocol is faster, but it requires start-up of the sunserve server (part of the Subversion installation). With this protocol, if the repository resides on the server's local drive then the following can be used: synserve -d --listen-host 127.0.0.1. In this case, the repository is readable by anyone from the server computer (also true for the file:// protocol).

# Warning

In a production system, you should always set up and use the svn:// protocol for system user access. See **Installing synserve as a Service** below.

Polarion does not currently support the file:// protocol with Subversion 1.9 repositories. (That version is bundled with Windows distributions, but installers create the repository using version 1.8 repository format.) If you manually update to version 1.9 before full support by Polarion is declared, use only the svn:// protocol (which is in any case the recommended protocol for a production system).

# **Examples:**

```
repoSystem=file://C:/Polarion/data/svn/repo
--OR--
repoSystem=svn://localhost/C:/Polarion/data/svn/repo
```

If the repository referenced via svn:// protocol is remote, then it will most likely be necessary to set up authorization for **svnserve**. You can find information about this in the **svnserve** documentation published online at: http://svnbook.red-bean.com/en/1.8/svn.serverconfig.svnserve.html.

#### Install synserve as a service:

The **synserve** server can be installed as a service on Windows and Linux Platforms. The main steps are:

- 1. Set up the **svnserve** service using the recommended settings for your specific version and environment (see examples below).
- 2. Configure Polarion to use svn protocol for the system user to access SVN by setting the following in the pp file: repoSystem=svn://localhost/
- Restart the Polarion server.

**Windows Example:** The following is an example of how to set up the synserve service on Windows. The following should be written as a single command-line command with a space in place of each line break in the example:

```
sc create PolarionSvnserve binpath= "\"C:\Polarion\bundled\svn\bin\svnserve.exe\"
--service -R -r C:\Polarion\data\svn\repo -R -M 0 --listen-host 127.0.0.1
--log-file C:\Polarion\data\logs\svnserve.log" displayname= "PolarionSvnserve"
depend= Tcpip start= auto
```

#### To start the service on Windows if it is setup using the command above:

```
sc START PolarionSvnserve
```

**Linux Example:** The following is an example of how to set up the synserve service on Linux using Subversion 1.9.

```
svnserve -d -R -r /opt/polarion/data/svn/repo --log-file /var/log/svnserve.log -M 0 --pid-file /var/run/svnserve.pid --listen-host 127.0.0.1
```

#### **Note**

The Linux command above will only start the process a single time. (Good for testing purposes.) To install synserve so that it runs after a reboot, see the install instructions specific to your distribution.

# Manage Polarion licenses

Your Polarion server may host more than one license type: named and concurrent, for example. For named user licenses, you need to specify the user IDs of all users who will utilize each named license. For concurrent licenses, you may want to limit the number of users who can access the portal concurrently (the maximum is controlled by the license), and/or you may want to define groups of concurrent users with different limits. You may also wish to check the current usage of your license(s).

You can manage licenses and view current license usage in the Polarion portal. The configuration is only accessible in the repository scope.

# Access license management

- 1. Log in with administrator permissions for the repository.
- 2. If you are not in the **Administration** interface after you log in, click the **Administration** link in the tool view of Navigation.
- 3. If you are in a project after you log in, click **Global Administration** at the top of the **Navigation** panel.
- 4. In **Navigation**, click on the **License** topic. The License page loads.

The portal page contains embedded Quick Help providing more detailed information on how to use the various license management features.

#### Specify license assignments

The License page provides a plain text editor in the **Assignments** section which enables you to edit the underlying configuration file, which is named users in the file system. Please refer to the Quick Help documentation on the License Assignment and Usage page of Administration for specifics.

When you change any text in the online text editor, the **Save** button is enabled which saves changes made in the portal to the configuration file.

Users not named in the license configuration are treated as concurrent users, and will utilize the default concurrent license specified in the configuration. If the default concurrent license is not specified, then these users will use the "highest" license having an unlimited number of concurrent users. If all are limited, then these users will use the "lowest" license with the highest number of concurrent users. If all licenses have zero, then these users will not be assigned a license and will not be able to log in.

Optionally, you can configure that all users who log into Polarion are automatically assigned a concurrent add-on license together with their basic license. If no concurrent add-on license is available, none will be assigned and users can still log in with the basic license. (See the embedded Quick Help on the License page in global Administration.)

#### **Note**

The license assignment configuration can be changed only once in 20 minutes.

### View license usage

The License page provides sections that display statistics on how users are currently consuming the licenses available on your Polarion system. The numbers are current as of the moment you invoked the page. Refresh your browser to see any change.

For information about the usage details displayed on the page, see: License Usage Reference.

#### **Update and reactivate licenses**

If your organization has purchased new, or obtained updated licenses for your Polarion installation, you can update the installation from the License page in Administration. Use the **Reactivate Polarion** button to begin the process. You will be taken to the first screen of the activation application, where you will find instructions for activating your license(s) online or offline.

A confirmation dialog appears on each online activation, before the activation request is actually sent. You must confirm that no running Polarion instance is activated by the License Key you are activating.

# Configure single sign-on (SSO)

By default Polarion uses form-based authentication and validates user credentials via Apache against a remote LDAP server or local passwd file. Additionally, Polarion supports token based authentication using Kerberos v5, Teamcenter Security Services or SAML 2.0.

# SSO configuration settings

See the **Configuring Single Sign-on (SSO)** section in the Windows or Linux Installation guides for detailed configuration instructions.

#### Warning

When using Single Sign-on with Polarion, users cannot modify their password in Polarion because the password is managed by the directory service.

#### **Note**

#### Fallback for Kerberos:

See the Windows and Linux installation guides to configure a fallback Kerberos authentication.

#### E-signatures with SSO

Electronic Signatures require that users provide their user credentials to file a signature. Therefore Polarion provides an authentication fallback in the SSO authenticated environment to enable Electronic Signatures. See the Signing Documents section for details.

#### Authentication via credentials for external web services clients

Some external clients may not be able to login via a token in an SSO environment. Administrators can optionally enable Web Service fallback authentication by credentials for users with external clients.

(e.g Someone accessing Polarion through a VPN who was not authenticated in the Windows domain.)

- 1. Go to the **LDAP Configuration** page.
  - ( Administration Global Administration User Management LDAP Configuration.)
- 2. Make sure that the LDAP Server Connection Settings are configured. (See the Windows and Linux installation guides for details.)
- 3. Tick the Allow for Web Services box in the Alternative Authentication via LDAP in a Single Sign-on Environment section.

# Alternative Authentication via LDAP in Single Sign-on Environment



4. Click 🖥 Save

#### Log out with SSO

Even in LDAP or Kerberos SSO configurations, Polarion still lets users log out of Polarion while remaining signed into the greater SSO environment.

This allows organizations with concurrent Polarion licenses to make the most of them while still taking advantage of a single sign on environment.

#### With Kerberos:

With a Kerberos Single Sign-on configuration, when users log out of Polarion, the SSO session is not invalidated.

#### With Teamcenter Security Services:

Polarion detects whether a user is also logged into Teamcenter Security Services and if no other TSS applications are active, Polarion will also log them out of Teamcenter Security Services.

#### Adjust pop-up login dialog size

By default, when using SAML or Teamcenter Security Services, the internal login is done via a new browser tab.

To disable this behavior, see Polarion Setup Single Sign On in the Polarion section of SIEMENS Doc Center.

Some 3rd party authentication screens may be larger than the default pop-up dialog that Polarion provides. Administrators can change its size by adding the following properties to the polarion.properties file.

```
com.siemens.polarion.ui.internalLoginWidth=NNN (Where "NNN" is the number of pixels.)
```

# The dialog **Width** in pixels.

```
com.siemens.polarion.ui.internalLoginHeight=NNN (Where "NNN" is the number of pixels.)
```

The dialog **Height** in pixels.

# Repository access management

# Repository access management overview

# **ALM**

Polarion utilizes an integrated Subversion repository to manage all data artifacts. Access to Subversion is controlled by the Subversion access file. Polarion provides a web based client with an interface to facilitate administrators who need to control repository access directly in Subversion. This topic explains how to locate and use the client. Subversion administration and access file management is a topic beyond the scope of Help. Basic knowledge of the SVN access file is assumed. For information about Subversion repository administration, see <a href="http://svnbook.red-bean.com/nightly/en/svn.reposadmin.html">http://svnbook.red-bean.com/nightly/en/svn.reposadmin.html</a>.

The Access Management feature enables you to edit data in the SVN access file online, in either repository scope or single project scope. It is an administration feature, so you must have administrator permissions for the scope you want to work with.

# Warning

#### Permissions change when creating a new Subspace within a Space:

If you create a Subspace within a Space, or Move a Space/Subspace from another Space, they do not inherit the SVN (repository) restrictions or permissions from their new parent. If you want to restrict access to a newly created Subspace, you have to remove the SVN access rights and create a custom rule for permissions (see Configure user Permissions - Define permissions for Artifact Sets).

# **Access the client**

- 1. Log in with administrator permissions for the repository or project you want to work with, and open it (See: Access Projects).
- 2. If you are not in Administration after logging in, click the **Administration** link in the tool view of **Navigation**.
- 3. In **Navigation** (topics view), expand **User Management** and select **Access Management**. The Access Management page loads in your browser.

#### **Use the Access Management client**

The Access Management page has 2 panes, top and bottom. The top pane enables you to browse through the repository structure as it is defined in the access file, and review the current Role and

User assignments for folders and files. Keep in mind that this pane is a viewport into the SVN access file... what you see is what is currently in that file.

The upper pane displays a string that shows the repository path to the currently selected folder or file. Some or all of the folder names in the path are clickable depending on your permissions:

- If you have global (Repository) administrator permissions, the all folder names in the path are clickable enabling you to navigate all the way up to the repository root.
- If you have project administrator permissions, the full path to the project folder is displayed, but you will not be able to navigate higher than the project root folder.

The lower pane of the Access Management page provides detailed information about access rights currently defined for the folder or file selected in the top pane. In the lower pane you can:

- Add, edit, or remove Role assignments for the selected folder or file. Click the / (Edit) icon in the **Roles Assignment** section in the lower pane of the Access Management page.
- Add, edit, or remove User assignments for the selected folder or file. Click the / (Edit) icon in the **Users Assignment** section in the lower pane of the Access Management page.

#### Note

Assigning access by **Role** is generally considered better practice than by **User**.

- Review Users and Roles that currently have read and write permission for folder or file currently selected in the upper pane.
- Review Users and Roles that currently have read-only permission for the folder or file currently selected in the upper pane.

By default, files in a folder have the same access permissions as the containing folder.

### **Edit Role assignments**

In either scope, you can edit Role assignments. This is generally considered preferable than assigning access to individual users.

- 1. Open the repository or project you want to manage and navigate to **Luser Management Access Management** in the **Navigation** pane.
- 2. In the top pane of the **Access Management** page, browse to the repository folder or file for which you want to edit Role assignments and select the item. The selected folder or file's SVN access details appear in the lower pane.
- 3. In the Roles Assignment section, click (Edit). The table of currently assigned Roles becomes editable. Each row of the table corresponds to a Role and the permissions currently assigned to it. The last row enabled you to specify a Role and its permissions to add to the currently selected folder or file. Columns correspond to user role and permissions. Check boxes correspond to enabling (checked state) and disabling (cleared state) each permission.

# Tip

**Deny All** overrides the other permission settings and revokes all permissions for the selected folder or file.

# Add Role assignments

- Make sure you are editing the correct folder or file select it in the top pane of the Access Management page. If the Role Assignments table is not editable, click the (Edit) icon on the section header.
- 2. In the last row of the table, click the ♣ icon to add a new row. Select the Role you want to add to the access file from the list in the **Roles** column. (For information about configuring Roles, see: Configure User Roles.)
- 3. Check the boxes corresponding to the permissions you wish to assign to the added Role. The options are:
  - Read Check to assign read-only permission, clear to revoke it.
  - Read & Write Check to assign read and write permission, clear to revoke it.
  - Deny all Check to deny all permissions, clear to revoke denial.
- 4. If you want to add another Role and set its permissions, click the → icon in the **Actions** column to add another row to the table. Select the Role and assign permissions as described above.
- 5. Click the **Save** button in the lower pane's toolbar to save changes from the GUI to the SVN access file.

# **Remove Role assignments**

- Make sure you are editing the correct folder or file select it in the top pane of the Access Management page. If the Role Assignments table is not editable, click the 

   ✓ (Edit) icon on the section header.
- 2. On the row(s) of the Role(s) you wish to remove, click the in the **Actions** column. The row becomes highlighted. (If you make a mistake, click the **Cancel** button on the lower pane toolbar and start over.)
- 3. Click the **Save** button in the lower pane's toolbar to save changes from the GUI to the SVN access file.

#### Edit user assignments

As previously mentioned, it is generally considered preferable to assign Subversion access by Role rather than by User. The most common reason to edit User assignments is in situations where some "clean-up" of the access file is desirable, removing User assignments that may have found their way into the file at some time or another. Of course, if you wish to assign access to some specific User(s), you can do so.

- 1. Open the repository or project you want to manage and navigate to **User Management Access Management** in the **Navigation** pane.
- 2. In the upper pane of the Access Management page, browse to the repository folder or file for which you want to edit User assignments and select the folder. The selected folder's SVN access details appear in the lower pane.
- 3. In the **Users Assignment** section, click (Edit). The table of currently assigned Users becomes editable. Each row of the table corresponds to a User and the permissions currently assigned. The last row enabled you to specify a user and permissions to add to the currently selected folder or file.

# Remove user assignments from the Subversion access file

- Make sure you are editing the correct folder or file select it in the upper pane of the Access Management page. If the **User Assignments** table is not editable, click the 

   ✓ (Edit) icon on the section header.
- 2. On the row(s) of the User(s) for whom you wish to remove SVN access, click the in the **Actions** column. The row becomes highlighted. (If you make a mistake, click the **Cancel** button on the lower pane toolbar and start over.)
- 3. Click the **Save** button in the lower pane's toolbar to save your changes to the SVN access file.

#### Add user assignments

- Make sure you are editing the correct folder or file select it in the upper pane of the Access Management page. If the **Users Assignments** table is not editable, click (Edit) icon on the section header.
- 2. In the last row of the table, select the User to which you want to assign permissions from the list of in the Users column. (For information about configuring Users, see: Configure User Roles.)
- 3. Check the boxes corresponding to the permissions you wish to assign to the selected User. The options are:
  - Read check to assign read-only permission, clear to revoke it.
  - Read & Write check to assign read and write permission, clear to revoke it.
  - Deny all check to deny all permissions, clear to revoke denial.
- 4. If you want to add another User and permissions to the table, click the → icon in the Actions column to add another row to the table. Select the User and assign permissions as described above.
- 5. Click the **Save** button in the lower pane's toolbar to save changes from the GUI to the SVN access file.

#### The Tilde exclusion marker

Subversion access management enables use of the tilde character (~) as an exclusion marker. Prefixing a user name or role with this character causes Subversion to apply the access rule to users *not* matching the rule. Polarion displays access rules as they are written in the access file, so prefixed items may appear in the Polarion access management user interface if they exist in the access file.

The system property <code>com.polarion.accessmngmt.useTildaFeatureForRoles</code> in the polarion.properties file controls whether or not Polarion's Access Management should offer tilde-prefixed roles. The property is set to <code>true</code> by default.

You can find more information about this marker, along with some examples, in the Advanced Access Control Features section of the web page at http://svnbook.red-bean.com/nightly/en/svn.serverconfig.pathbasedauthz.html.

#### **Access control for Documents**

Controlling access to Documents can be an issue in some organizations. While administrators cannot specify access control on specific Documents, it is possible to control access to Document folders. Document folders are found under the modules folder, under the project folder. If, for example, you wanted to set access control on the "User Requirements Specification" Document in the eLibrary example project, you would set the desired permissions on the folder elibrary/modules/Specification/User Requirements Specification.

When access management restricts a user from accessing a Document, the Polarion user interface does not show the Document to the user in Navigation.

#### Note for Global administrators

If your user account has global (i.e. repository-scope) administrator permissions, but you are editing Role access in the project scope, then the appearance of some items in the Roles list is somewhat different than for administrators having just project-scope administrator permissions. Global administrators will see Roles prefixed with a project ID: myproject-project\_user, for example. A project administrator would see the same roles as a project user.

# Remove unused Excel import configurations

If some Excel Import Configurations become disused or obsolete over time, they can be removed via the Repository browser. Import configuration may be save either in the global/repository scope or in the scope of some project. Deleting an Import Configuration does not require administrator permissions, but the user performing the removal must have permissions to delete content from the repository in the relevant scope.

The Repository Browser path to Excel Import Configurations is:

.polarion/tracker/import configurations/excel

# **Configure the Dashboard**

# **ALM**

Scope(s): repository, project

The Dashboard topic is implemented as a page in the integrated Wiki. You can customize the layout and content of this page using the macros discussed in this section. (Details on macro syntax are provided in the embedded Wiki Syntax Help in the Polarion portal).

#### Macros for layout

You can organize the content of the dashboard into table and or column structures using these macros:

- {table} Creates a table structure,
- {section} Defines a page section,
- {column} Divides a section into columns,
- {style} Enables embedding of CSS styling into the Dashboard (and other wiki pages).

Again, please refer to embedded syntax help for syntax and usage examples of these macros.

#### Macros for information

There are several macros which roll up and display information from the system in the Dashboard. The most important ones are:

- {fact}
- {linechart}
- {piechart}
- {report}

# Marco: {fact}

This macro renders some useful values from factbase. These values are stored for example in the workitems-data.xml file under the data folder of the Polarion installation. For example, the file for the repository level is located in: %POLARION HOME%/data/RR/default/.reports/xml/workitems-data.xml.

This xml file contains computed values that can be rendered as numbers or graphs: UNRESOLVED-BY-ASSIGNEE, OPEN-BY-SEVERITY, STATUS and so on. Values are populated by reports (see: Configure Reports.

#### Macros: {linechart}, {piechart}, {report}

The {piechart}, {piechart}, and {report} macros render information from HTML and XML files generated automatically by system calculations. The main thing you need to understand about them is that the path in their report-path is a relative path in the PR directory.

For example, if a macro is executed in project elibrary then the path is relative to the reports for that project.

Reference information about fact bases and values can be found in: Calculations and Facts.

#### **Update Dashboards**

There is a calculation named all that calculates all metrics, charts etc. in a given scope. The default system configuration has a job named All Calculations that invokes this calculation. Schedule or call this job if you want to be sure that everything is freshly calculated.

Each section of the Dashboard contains an **Update** button which can be used to recalculate and refresh specific information displays in the Dashboard.

Depending on what you have configured to appear in the Dashboard, you may only need run some calculations, as opposed to all calculations as described above. You should configure the specific reports that run the calculations for the data you are displaying in the Dashboard. For information, see: Working with Report Configurations.

# **Customize XML export templates**

# ALM REQ QA

Work Items can be exported to XML (see: Exporting to XML. Export is accomplished by means of XSL templates (XSLT). You can create your own XSL templates to transform exported Work Items to another XML document, a HTML document, a plain text file, or PDF or RTF. When creating your own XSLT, refer to the Polarion XML Export Schema.

You can access the templates via the **Administration** interface: **Work Items Export Templates**.

You can also use the integrated Repository Browser to access export templates. Templates are located on repository path/.polarion/tracker/export\_templates/. You must use this tool to access the XSL templates.

#### XSL template naming convention

Each of the exporters has a unique set of templates. Which template is used for which exporter is determined by the name of the template:

- Transformed XML Export: \*-xml.xsl
- HTML Export: \*-html.xsl
- Text Export: \*-txt.xsl
- RTF and PDF Export: \*-fo.xsl

#### Where to deploy XML export templates

Upload your custom XML export templates to the export templates folder in the repository: .polarion/tracker/export templates.

# **Customize Excel export templates**

# **Customize Excel export templates overview**

Excel export templates enable you to set up different export scenarios for users of the Excel round-trip features. Templates specify which Work Item field are exported for round-trip, and the default type for new Work Items created via Excel round trip.

Several export templates used for exporting Work Items to Microsoft Excel format are provided in the xlsx: Microsoft Excel section of the Export Templates topic of Administration.

( Administration → Work Items → Export Templates → xlsx: Microsoft Excel.)

The repository folder is /.polarion/tracker/export templates/excel.

You will probably want to create you own Excel export templates derived from the Polarion default export templates.

The following sections discuss the characteristics of an Excel export template, and describe several commonly needed customizations:

- How to make a column read-only
- How to add a column
- How to add a calculated column

# **Default Excel export templates**

#### Polarion provides three default Excel export templates:

- Basic.xlsx Template with some basic column-to-field mapping. A good starting point for many customizations.
- Empty.xslx Contains the required tables (describe later), but nothing else. No columns are defined in the template sheet, and there is no default column-to-field mapping. If used for round-trip export, the export dialog fields selection is pre-filled with all columns that are shown in the Table view of Work Items.

This template is best used as the basis for new templates developed "from scratch".

TimeReport.xlsx - This template defines a work time report. It serves as a good example of how to define calculated columns in your custom Excel export templates.

Start by downloading the Excel export templates and checking to see how much is already implemented that you can use. The Empty.xlsx template is the most basic template with minimal data. The Basic.xlsx template has some typically used fields and columns pre-defined. The Time Report.xlsx template has some examples of calculated columns. Create a copy of one of these export templates to use for your customizations.

#### **Export template structure**

Work Items are exported to an Excel table (a named region of the worksheet) named WorkItems. Your template must contain this table, which should contain only a header and one normal row - the "template" row (see figure below). In Polarion's pre-defined export templates, this table is found in

Sheet 1. Any column in this template sheet that has the a column label corresponding to a Polarion Work Item field ID is automatically mapped to the corresponding field. This includes custom fields. You can add columns as needed. If you need to have column names that do not correspond to field IDs, use the mapping table on the **Polarion** sheet.

The pre-defined Excel export templates from Polarion also contain a hidden sheet named Polarion. This sheet specifies some settings for the exporter and enables column to field mapping when template column names do not correspond to Polarion Work Item field names. Your custom template should contain this sheet with the 2 tables from Polarion's default templates. You will need to manually "un-hide" this sheet in the workbook if you need to make changes to it, and then change it back to hidden before distributing your custom template.

The first table in the **Polarion** sheet is a table of property-value pairs that enable you to set default values for two properties:

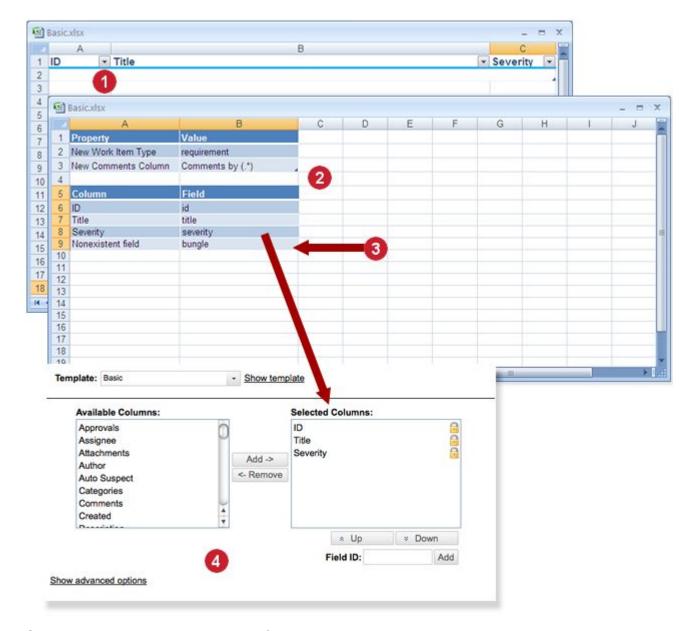
- New Work Item Type Specify the type for any new Work Items that may be specified by end
  users in exported round-trip workbooks based on the template, and created in Polarion during
  re-import.
- **New Comments Column** A regular expression value that matches headers of columns that should be imported as new Work Item comments.

The second table on the Polarion sheet is a column-to-field mapping table. Use this table to map any columns in the template table of Sheet 1 whose column names do not correspond to Polarion field names. Each row corresponds to a column in Sheet 1 that doesn't have a column name matching a field ID. (If there are rows in this table that duplicate the mapping on Sheet 1 you don't need to remove them. The mapping will not be duplicated.)

When adding a new column and field row to the mapping table, be sure to use the **Insert** context menu on a row in the mapping table. Do not add rows outside the table. Rows inside the table will be shaded.

If you should happen to specify a field that doesn't exist, it will not appear to users in the **Export Work Items** dialog when they select the Excel export option.

The end result of your customized template, when uploaded to the portal, should be that all fields you have mapped in the template appear to users as the default fields in the **Selected Columns** list of the Export Work Items dialog. It's recommended that you save your templates with different filenames from the default templates. You can upload your finished templates on the Export Templates page (described earlier) of Administration, then test them by invoking **Export** in the Work Items table and selecting **xlsx**: **Microsoft Excel** in the **Format** field. If you need to modify your template to correct errors (or for any reason), you can do so and re-upload it using the **Replace existing export template with this upload** option.



Sheet column to Polarion Work Item field mapping in an Excel export template.



# Make a column read-only

- 1. Open the sheet of the export template containing the WorkItems region (normally on Sheet1).
- 2. In the first row after the header row (the "template" row referred to earlier), select the cell you want as read only.

- 3. On the cell's context menu, choose **Format Cells**.
- 4. Select the **Protection** page of the dialog and check **Locked**.

#### Add a column

- Open the sheet of the export template containing the WorkItems region (normally on Sheet1).
- On first row after the header row (the "template" row referred to earlier), drag the lower right corner of the right-most cell until you see the outline for a new column. (Or use the content menu of the right-most cell to insert a column to the right).
- Set the value of the new column's header to the name of a Polarion field to which you want the column mapped.

Alternatively, set the header value to any unique value, and then on the **Polarion** sheet add a row to the mapping table.

#### Add a calculated column

#### Tip

Download and open the Time Report.xlsx export template to see an example of calculated columns.

- 1. Create a new column as described in the previous section, but do not map it to any field. Rather, specify a unique value in the column header.
- 2. Enter the desired formula in the template row cell of the new calculated column.

#### **Note**

You cannot use relative cell references. Rather you should use table column references. For example:

```
=[Time Spent]+[Remaining Est.]-[Initial Est.]
```

Again, the calculated columns in the pre-defined Time Report.xlsx export template can provide a useful example of such references.

### **Specify column for Test Step iteration numbers**

Excel export for offline Test Case execution supports export of Test Cases with multiple Iterations of Test Steps. By default, the export template is configured to append a label for each Iteration to the **Title** column in the exported workbook. It is possible to configure which field displays the Iteration number in the property Append Iteration Label to Field in the hidden Polarion sheet. The default is the **Title** field.

# **Configure OLE Object support**

It is possible to import Microsoft Word documents that contain OLE objects. Polarion can display OLE Object thumbnails during Word document import. However, some additional third-party image converter software must be installed and configured before you can import such Word documents. OLE Objects in documents must contain their thumbnails in .emf or .wmf file formats, and the image converter used must support their conversion into JPEG. OLE Objects themselves are not imported, only their thumbnails.

Installation of third-party image converter, and configuration of Polarion to support OLE objects are covered in the Polarion installation guide documents for Windows and Linux, bundled with the respective distributions and available on Siemens Doc Center.

# **Customize the Login screen**

You can customize the Polarion portal login screen to display a custom logo or other image in place of the Polarion logo. The image must be hosted online in a location accessible to your Polarion server via http. Images may be GIF, PNG, or JPEG format. Recommended width is 100 pixels. (Height is scaled automatically.)

In a clustered environment where Polarion is served over multiple servers (physical and/or virtual), if a custom image is specified on the server that is the entry point for the system, the image will be used as the icon for each server's entry point. In an environment utilizing a single server, the custom image appears beside the Polarion logo.

- 1. Using a text editor, open the system properties file polarion properties (follow link for location).
- 2. Add the following line to the file:

```
login.companylogo=[IMAGE URL]
```

..where [IMAGE URL] is the URL of the logo or other image you want to display in the login screen.

Save the file.

The Polarion server must be restarted before the image will appear in the login page.

# Repository browser mime types

The Repository Browser uses the following information to detect if a file is binary or text: the svn:mime-type property, the container (Tomcat) mime-type settings, and configuration parameters BinaryMimeTypes and BinaryMimeTypes set in the web.xml (located in the web client plugin folder of the Polarion installation: polarion/plugins/org.polarion.svnwebclient n.n.n/webapp/WEB-INF/ - where "n.n.n" are numbers specifying a Polarion release number: 3.5.0, for example).

If mime type is detected in SVN properties or container settings, it is then looked up in the BinaryMimeTypes and BinaryMimeTypes lists. If the mime type isn't set or isn't present in those lists, then an automatic algorithm, which analyzes the first 1000 bytes of the file and returns the best guess at the file type, is used.

This approach still cannot guarantee 100% accuracy of file type reporting and file handing in the Repository Browser. For example, one organization might want to have a file type .xyz treated as binary, while another wants the same file type interpreted as text. If you find that some file types are not reported and handled correctly in the Repository Browser, or if you have some file types that you want to make sure are always interpreted correctly, you can edit the above-mentioned web.xml, setting mime-type mappings and interpretations in the BinaryMimeTypes and BinaryMimeTypes lists.

# Take a thread dump

A thread dump captures the state of all Java threads in the system and provides crucial information for the troubleshooting of performance problems. Polarion administrators can manually take a thread dump directly from the Maintenance administration UI. The thread dump is then written to the Polarion log that can be sent to Polarion technical support.

# Advanced system tuning

# **System properties**

There is a set of system level configuration properties available to administrators and developers that affect how various Polarion system processes and functions work or behave. All properties are embedded in the Polarion system. A default subset is exposed in the polarion.properties file, to which more properties can be added if it is desirable to modify their default values. This topic documents a small subset that Polarion administrators tend to modify most often. The full set of properties is documented in the Polarion System Configuration Properties Reference, a PDF document available online in the Polarion support area.

# attachments.indexingOfAttachmentContent.enabled

When not present in polarion.properties, or if added to the properties file and set to true, indexing of the content of attachments is enabled. When enabled, attachments to Documents, Wiki pages, Work Items and Test Runs are indexed. Attachment indexing is performed by a background job when the server is started. Users may not be able to search attachments for some time until the indexing job is completed.

Regardless of the setting, other attachment fields (author, title, fileName, updated and length) are not affected by this property and are always indexed (and therefore searchable by users).

**Example**: attachments.indexingOfAttachmentContent.enabled=true

The attachments index can be explicitly refreshed via the Maintenance page in Administration.

For a list of supported attachment file types, see Attachment File Types.

# attachments.maxSizeOfIndexedAttachmentInMB

If property attachments.indexingOfAttachmentContent.enabled is enabled, this property sets the maximum file size, in megabytes, that will be indexed for content, and therefore searchable for content by users. If an attachment is larger than the value specified in this property, only the attachment fields (author, title, fileName, updated and length) are indexed and searchable.

The value can be either an integer value (file sizes of 1 MB or more), or a float value. If no value is provided, or the property is left out of the properties file, a default value of 50 MB will be used if attachments.indexingOfAttachmentContent.enabled is enabled. The default limit of 50 MB has been found generally suitable for Polarion installations meeting the recommended system requirements

The setting applies to all attachments to any content that supports them: Wiki pages, Documents, Work Items, Test Runs, etc. The limit also applies to imported documents. If users encounter "Out of Memory" errors when uploading attachments or importing documents, check the limit in this

configuration setting and either adjust the setting, or add more system memory and storage, as needed to meet the needs of your users. **Example**:

- attachments.maxSizeOfIndexedAttachmentInMB=2
- attachments.maxSizeOfIndexedAttachmentInMB=0.5 (i.e. 512 KB)
- attachments.maxSizeOfIndexedAttachmentInMB=5.5 (i.e. 5.5 MB)

#### maxAttachmentSize

Controls the size of any attachment in Polarion. If the attached file is bigger than the configurable limit, the upload is stopped.

It applies to Work Item, Wiki Document, Live Document and imported document attachments.

The default setting is 200 megabytes written in bytes (20000000).

# com.polarion.tracker.defaultNumberOfHeadings

Administrators can configure the maximum number of heading levels available by adding this property plus a value (=x) to the polarion.properties file. There is no upper limit to the number of heading levels.

# com.polarion.durationHoursPerDay

Specifies the number of hours per work day. Requires an integer value (default value is "8"). Other type values are silently ignored and the default substituted. The setting applies globally and cannot be overridden in the project or project group scopes. The specified value appears read-only in the Work Items: Planning page of Administration.

#### com.polarion.enumerations.objectEnumerationsLimit

Sets a hard limit on the number of items that are loaded into enumerated lists of system objects such as Documents, Pages, Test Runs, users, etc. Excess items cannot be selected in the UI. Default value is 10 000. You can set a whole number (for example, 200), or -1 for unlimited.

Take for example a custom field in which users select a Document from a drop-down list of Documents (provided by an object enumeration). If the system contains hundreds or thousands of Documents, populating the select list can take too long. Setting a lower limit in this property can significantly improve performance.

See also: com.polarion.ui.maxRenderedItemsInCombo.

## polarion.reindex.from.revision

- Which revision is taken as the first one to process when Polarion is started for the first time, or if its runtime data were cleared (e.g. due to reindex).
- Possible values are FIRST and HEAD.
- Default value: FIRST.
- setting this to HEAD will speed up fresh start significantly, but the history of Work Items will not be available (only changes done after the start).

#### polarion.startup.disable.artifacts.auto.recognition

- Controls whether or not build artifacts auto-recognition during startup should be disabled (affects only projects where there are no configured build artifacts and auto-recognition is not disabled on the project level).
- The auto-recognition can be initiated manually from Administration.
- Default value: false.
- Setting the value to true will result in significant speed-up of first start or reindex.

# polarion.startup.disable.artifacts.change.detection

- Controls whether or not detection of build artifacts-related changes should be disabled during startup (does not affect first startup or reindex or projects created while the server was not running).
- Default value: false.
- Setting the value to true will result in significant speed-up of restart.

# com.polarion.ui.maxRenderedItemsInCombo

Universally controls the number of items loaded into combo box lists. Default value is 100. Decreasing the value can improve loading of lists, if performance is noticed to be slower than optimal, which may occur if there are many projects with many concurrent users. Users can filter for excess items by typing a string of characters in the combo box. For example, assuming an item "Zebra Stripe Pattern", user can filter by typing <code>zebra</code>.

See also: system property com.polarion.enumerations.objectEnumerationsLimit.

# useDecimalHoursDurationFormat

- Controls whether or not the value for hours in time reporting fields (Initial Estimate, Time Spent, etc.) must be entered as a decimal value rather than the default fractional.
- Default value: false.
- When set to true, hour durations in all time reporting fields are displayed and entered as decimal hours only 22.33 for example. The examples displayed by clicking on the "?" next to time input fields are changed when this feature is enabled, and the export option Convert Durations to decimal hours is not displayed in the export dialog.

Storage in XML persistence and pre-2011 LiveDocs is still in the fractional format, only hours are not converted to days. So the value in the example above is actually stored as 22 1/3h. It is stored that way rather than 22 33/100h because of smart parsing of user-entered values to the fractional format, which recognizes the decimal representation of fractions (y/3, y/6 and y/12). So for example entering 0.08 or 0.083 is parsed as 1/12h. The smart parsing can be disabled by setting the system property dontUseSmartDecimalHoursDurationFormatParsing to true.

# wordImport.ignore.[CLASS]

Where [CLASS] the short name of an object class from the docx4j library

- Default value: false
- Controls whether or not unsupported content, of the type represented by the specified class, is reported with the Unsupported Content message (see screenshot) in the result Document after importing a document from Microsoft Word.

By default, unsupported content contained in the import source is not imported, and is replaced by the graphical message shown below in the result Document after the import process finishes. If you would rather that such content be silently ignored, you can add this property for each type of unsupported content you do not want reported in the import result Document.

# The original document containted an embedded object that has been removed. Embedded objects are not accessible via all browsers.



Polarion recommends to:

- 1. Open the original document in Word
- 2. Review the object and copy visual contents to the clipboard
- 3. Detach this embedded object as a file in its native format
- 4. Attach the file to the appropriate Polarion Work Item
- 5. Paste the clipboard contents into the Work Item decription

Graphical message when an object in a source document cannot be imported to Polarion. Set this property to suppress it.

**Example:** wordImport.ignore.R.DayLong=true Where R.DayLong is the short name of the class of the object from docx4j library.

In this case, content objects of the R. DayLong class will be ignored during import and no message will appear in the import result. To find the class name, look at the tool-tip on the Unsupported Content message in an import result where some unsupported content was present in the source document.

#### Note

OLE thumbnails are supported for imported Word documents when third-party image converter software is installed and configured. See Documents with OLE Objects for more information.

# com.polarion.logoURL

Optionally specifies the URL of a remotely hosted logo or icon image to display in the Navigation panel in lieu of the default logo. The hosted location must be accessible to the Polarion server. If this property is not specified, the default logo is displayed.

If specified, the remote image should ideally be 60 W x 64 H (pixels) in GIF, PNG, or JPG format. Larger images should ideally have the same aspect ratio, and will automatically be resized and cropped if necessary to fit the display space allocated by Polarion in the Navigation panel.

This can especially useful for organizations running multiple Polarion servers. A different logo image can be configured for each server, enabling end users to tell at a glance which one they are connected to.

# Example:

com.polarion.logoURL=http://myhost.myorg.com/images/server1 logo.png

Property: com.polarion.ui.lock.timeout

#### com.polarion.ui.lock.timeout

Optionally specifies a lock timeout in minutes for the Diagram Editor. When a user begins editing a diagram, a lock is set and no other user can edit it until the diagram changes are saved, the user closes the Diagram Editor, or the lock timeout set in this property occurs.

If this property is not specified, the lock timeout defaults to 8 hours. The user who obtained the lock must save the diagram or close the Diagram Editor in order to remove the lock.

Example: com.polarion.ui.lock.timeout=30 (Sets lock timeout to 30 minutes.)

# com.siemens.polarion.repository.git.timeout

Administrators can set a Repository Polling timeout value for when changes are pulled from an external Git repository.

If this property is not specified, the default timeout for Repository Polling job is set to 30000ms (30s)

Example: com.siemens.polarion.repository.git.timeout=3000 (Sets Repository Polling job timeout to 3 seconds.)

#### com.polarion.ui.login.resetPasswordLinkURL

Holds the URL of a page on your system that you have provided for users to reset their password. You must also set the **resetPasswordLinkLabel** property (below).

#### Example:

com.polarion.ui.login.resetPasswordLinkURL=http://www.mydomain.com/user/recoverpw.php

See also: Enabling Users to Reset Password.

#### com.polarion.ui.login.resetPasswordLinkLabel

Link text to display on the Polarion portal login page, leading to your password reset page. You must also set the *resetPasswordLinkURL* property (above).

#### Example:

com.polarion.ui.login.resetPasswordLinkLabel=Reset Password

See also: Enabling Users to Reset Password.

#### **Configure Tomcat to use an HTTP connector**

By default the Tomcat application server is connected via an AJP connector to a local Apache that gets the clients' traffic from the load balancer. Starting with Polarion 3.18.2 it's possible to configure HTTP protocol instead, bypass the local Apache instance for the browser clients' traffic and directly

connect the load balancer with the Tomcat application server. This is useful mainly when a different Load balancer is used. See Apache Tomcat 9 for details.

The AJP connector is deployed automatically, but you can specify the ajp-port in the polarion.properties file by adding the following line:

```
TomcatService.ajp13-port=8889
```

# Configuration steps

1. Find the following property in the polarion.properties file.

```
TomcatService.protocol=AJP/1.3
```

2. Replace it with:

```
TomcatService.protocol=HTTP/1.1
TomcatService.port=80
```

This will deploy Polarion directly on the HTTP protocol on port 80.

# Log system load averages

A log for system load averages has been introduced and can help administrators analyze performance issues that are hard to pinpoint.

(Where it's unclear if a slowdown, was due to a suspected operation or because the system happened to be overloaded when the operation was run.)

#### Note

Only available for Linux. Here's why. (The settings will not harm Windows environments, but no load data will be collected.)

#### It can be configured and customized via the following system properties:

```
com.siemens.polarion.systemload.loggingTimeInterval
```

- The system load average time interval is expressed in milliseconds.
- The default is 10 seconds. (10000 milliseconds.)

```
com.siemens.polarion.systemload.minimalValueToLog
```

- Enter a double value that defines the point you want to start logging the system load average.
- The default value is 1.

#### **Note**

The value provided from the system load average is not a percentage of the CPU usage.

For example, with a CPU usage of 88.3, the system load average could be 43.

You can get some good information on the Linux stress tool at the following websites:

- https://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/
- https://linux.die.net/man/1/stress

#### Tip

If you'd like to learn more about Linux load averages, check out this informative blog post.

# **Activity Stream properties**

Administrators can configure history limits (number of days and maximum number of items) for each Activity source, plus a default registered activity source, via the system properties listed in this section.

# All Source Types:

- com.polarion.activity.maxActivitiesAge Default is 90 (days)
- com.polarion.activity.maxActivitiesLimit Default is 5000 (activities)

#### **Specific Source:**

- com.polarion.activity.[sourceID].maxActivitiesAge Specify an integer value. Value represents days.
- com.polarion.activity.[sourceID].maxActivitiesLimit -Specify an integer value. Value represents number of activities.

#### **Specific Source Type:**

- com.polarion.activity.[sourceID].[type].maxActivitiesAge Specify an integer value. Value represents days.
- com.polarion.activity.[sourceID].[type].maxActivitiesLimit Specify an integer value. Value represents number of activities.

# Work Item tree rendering

In the **Tree** view of **Work Items**, some user queries can return a tree so large that the performance of the entire system is adversely affected. Before constructing a tree of Work Items, Polarion checks two system properties with default values designed to prevent this. However, administrators can review and optionally adjust these depending on their system's resources.

The system property <code>com.polarion.ui.treeTable.softLimit</code> sets a first check limit on the number of items returned by a query run by a user in the Tree view. The default value is 1000. If you want to change that limit, you need to add this line to the system properties file: <code>com.polarion.ui.treeTable.softLimit =500</code>. (Replace "500" with an integer specifying the limit... it can be smaller or larger than the default 1000 items, depending on your system's resources.)

The check is performed on the number of items returned by the query. If the count exceeds the limit, the user is presented a dialog informing that the tree is large and can take a long time to load, and asking permission to display only the top **N** nodes (where N is the limit set in the above property). These top items can be on any level of the tree. This is neither the final number of nodes, nor the final

number of roots. Rather, it is the initial number of roots. When the tree is created some of these items might be removed and others might be added.

If the user rejects loading of the top N items, a second condition is checked in the system property com.polarion.ui.treeTable.hardLimit (default value 10000). If the total number of nodes returned does not exceed this limit, the tree is constructed from the full query result. Otherwise, the user is shown a message that the tree is too large to show and has been limited to N nodes, where N is the value set in this property. Again, the administrator can set a higher or lower limit by adding com.polarion.ui.treeTable.hardLimit to the polarion.properties file.

# **Matrix rendering**

There are two properties in the polarion properties file that define the maximum number of cells in a Traceability matrix or a Variants comparison matrix.

```
com.siemens.polarion.ui.matrix.hardLimit = 50 000
```

(The maximum allowed number of cells.)

```
com.siemens.polarion.ui.matrix.softLimit = 10 000
```

(Triggers a warning before displaying the matrix if the number of cells exceed this number.)

#### Fonts and font substitution

The fonts available in LiveDoc Documents and rich text edit fields, and font-substitution for import and round-trip operations is configurable in the system propertycom.polarion.ui.richText.fonts.

#### Font substitution works for:

- Import from Microsoft Word and Microsoft Excel.
- Round-trip for the above.
- Text copied from sources outside Polarion AML (e.g. from a Word document) and pasted into a LiveDoc or rich text field.

Font substitution replaces certain fonts with fonts defined in Polarion. Any font not available in Polarion and for which there is no substitution set is replaced by Polarion's default font.

Available fonts are separated by semicolon, and font substitution is separated by a comma. Consider the following example:

```
com.polarion.ui.richText.fonts=Arial, Helvetica; "Courier New", Courier, monospace,
HanWangKanTan; Georgia, Garamond; "Times New Roman", Times
```

- Arial, Courier New, Georgia and Times New Roman are set as available fonts.
- Helvetica is substituted by Arial
- Courier, monospace, and HanWangKanTan are substituted by Courier New, etc.

The defined fonts must be installed on client computers. Any missing fonts will be substituted by the default font, Arial This is hard-coded into Polarion and cannot be changed by administrators.

#### Lucene word splitting

In Version 2014-SR1, Polarion updated the embedded Apache Lucene query engines. The way indexed text is split into words changed from the previous Lucene version. Polarion now provides a system properties to enable administrators to control how words are split during query operations.

Property: search.wordBoundaries

Defines how word boundaries are defined for the index. Two values are supported:

- splitByPattern The input is split to words using a regex pattern. This is the default.
- standard The word boundaries are defined according to Unicode standard. This value should always be used for Asian languages

**Property:** search.wordBoundaries.splitByPattern

Defines the pattern to be used when using "search.wordBoundaries=splitByPattern". It is used with: java.lang.String.split(String).

#### Default value:

The default value ensures that the text is split by white-space, and the leading and trailing sequences of non-alphanumeric characters are removed from the remaining character sequences. The rest are "words" for the index. For example: <code>MELIO! WI-1234 \_help\_</code>. is split to words <code>Hello, WI-1234, and help</code>.

### **JVM** properties for Maven calculation

When updating to a new Polarion version, your current settings in JVM properties for Maven calculations may result in an Out of Memory (OOM) error **PermgenSpace**. In this case, you will need to adjust JVM memory values in the system file calculation.properties

(Repository path: .polarion/reports/calculation.properties).

# Split DAO cache by prototype

Polarion provides the option to customize the DAO cache configuration via the file \$POLARION HOME\$/polarion/configuration/ehcache.xml. The default configuration should be optimal for 8 GB Xmx settings on the server and should work well for larger installations as well. However, administrators have the option to adjust the cache size to store more objects in the in-memory cache.

The size of some objects may vary considerably. For example, you may have small manual Test Runs that are significantly smaller than the expected 2 MB object size. It might be useful to resize the cache for such objects.

To do that, copy the ehcache.xml file from:

\$POLARION\_HOME\$/polarion/plugins/com.polarion.platform\_x.y.z/platform.jar/META-INF/ehcache.xml(where
"x.y.z" denotes your current Polarion version), to

\$POLARION\_HOME\$/polarion/configuration/ehcache.xml and adjust particular settings in the copy.

(On Windows, replace forward slash characters in these paths with backslashes.)

# **Chapter 5: Reference**

# **Administration reference**

#### Polarion folder structure

This section contains reference information about the default Polarion folder structure after installation. Where there are differences between Windows and Unix, these are noted. The Polarion folder structure is quite flexible - it depends directly on the installation procedure and Polarion configuration. The structure described here is the structure created by the default installation procedure.

#### Windows structure

In documentation, <code>\$POLARION\_HOME\$</code> means the root folder of a Windows installation. The default path used by the Windows installer is <code>C:\Polarion</code>.

After installation on a Windows system, the Polarion folder (\$POLARION\_HOME\$) contains the following subfolders which are unique to the Windows installation (i.e. are not created during the Linux installation process):

#### **Bundled**

This folder contains several third-party applications that Polarion requires in order to work, and which are installed automatically by the Windows installer. If you chose not to have the installer install some application and manually installed it somewhere else, then it doesn't appear in this folder. For example, if you took the Custom installation option and unchecked the option to install Subversion, but chose to install Apache and PostgreSQL, then only the folders with the Apache and PostgreSQL distributions will appear here. By default, the <code>bundled</code> folder contains the following folders with the appropriate distributions:

- apache
- postgres
- svn

#### Note

Java is also required in order to run Polarion ALM. However, it is no longer bundled due to licensing issues. It is necessary to obtain and install the Java version recommended in the Polarion Installation Guide for Windows, a PDF document included in the Windows distribution.

### polarion shortcuts

\*.lnk-files (desktop shortcuts) for starting and stopping the Polarion server and Apache services.

#### **Linux structure**

The installation procedure places binary, executable, and configuration files under (/opt/polarion). Data is stored at /opt/polarion/data on Linux, and the location is user-configurable. The Linux installation contains no folders that are unique to it (i.e., not present in a Windows installation).

#### Common folders

This section covers folders that are common to both Windows and Linux installations.

#### data

Parent folder: /var/lib/polarion (Linux) \$POLARION HOME\$ (Windows)

This folder contains all data that Polarion operates with. By default the following folders are present:

bir (Build Information Repository)

The projects folder contains builds of all projects.

logs

Log files

portal-data

Internal Polarion portal data

rr (Reports repository)

This folder contains project reports and reports configuration files (as global as for projects).

shared-maven-repo

Maven 2 repository accessible through http://.../maven2 by default

This is the integrated Subversion repository. The folder contains both required files (access and passwd) and the repository itself (in the folder repo by default).

workspace

Contains: logs, manifest-files for plugins and some configuration files. You can view and remove Polarion log files here.

#### polarion

Contains: Polarion system files including configuration and properties files, plugins, license-related files, and Help files. The following subfolders are present:

configuration

Polarion configuration files including the polarion properties system configuration file.

features

Properties files needed by various Polarion features.

license

Optional storage for your Polarion license file. Contains definition file specifying which users can use which product edition (in cases where you have a license for multiple product editions).

• plugins

Polarion system plugins that enable Polarion to function. There are many subfolders. Documentation does not cover each one individually, as there is really nothing there that you as a user or administrator need to deal with. It may be worth noting that Help files are stored in the folder <code>com.polarion.xray.doc.user\_<n></code> (where <n> is the current version number). This can be useful should any Help updates be issued between releases and you need to install them manually.

# **Passwords**

In general, local users (i.e. users mentioned in passwd file) cannot have non-ASCII passwords (i.e. passwords containing characters not available in the US-ASCII character set. However:

 If Polarion server is running on Linux, LDAP users can have non-ACSII passwords provided the following line exists uncommented in the polarion.properties system properties file (follow link for location):

```
svnkit.http.encoding=UTF-8
```

2. If Polarion server is running on Windows, LDAP users can have passwords containing characters from a subset of non-ASCII characters if one of the following lines exists uncommented in the polarion.properties file:

```
svnkit.http.encoding=iso-8859-1
svnkit.http.encoding=<LOCAL CHARSET>
```

The first allows passwords containing characters from the ISO-8859-1 encoding. <LOCAL\_CHARSET> in second should be replaced with a local Windows encoding (e.g. windows-1250). When one of the encodings is specified, then passwords containing characters from the specified encoding will work.

#### **Note**

The ASCII double-quote character (") may not be used in local passwords.

# Polarion system processes

The system processes used by Polarion are **polarion.exe** and one or more instances of **java.exe** or **javaw.exe** (on Linux, java). The exact number of processes depends on the number of currently running jobs. In addition to these processes, the Apache service, httpd.exe (on Linux, httpd) also runs, and usually runs two processes with the same program name (i.e., **httpd.exe** on Windows, or **httpd** on Linux).

# System properties file location

The system properties file polarion.properties contains configuration settings affecting various system processes and functions.

This file resides on the server's file system. The path varies according to the server operating system hosting the installation:

- **Windows**: %POLARION HOME%\polarion\configuration\polarion.properties
- **Linux**: %POLARION HOME%/etc/polarion.properties

You can edit the system properties file using a text editor application such as Notepad or Vi. There is no user interface for it in the Polarion portal.

#### **Note**

- It is highly recommended to make a backup copy of the current production version of the system properties file before modifying it. Then if the modified file causes problems, you can easily restore the last working version.
- Changes to system properties do not take effect until the Polarion server is restarted.

For more information, see System properties.

# Scheduler cron information and examples

Cron-like specification of moments in time when a job should be started.

Field Name	Allowed Values	Allowed Special Characters
Seconds	0 - 59	, - * / *
Minutes	0 - 59	, - * / *
Hours	0 - 23	, - * / *
Day-of-month	1 - 31	, - * ? / L W C *
Month	1 - 12 or JAN - DEC	, - * / *
Day-of-Week	1 - 7 or SUN - SAT	,-*?/LC#*
Year (Optional)	empty, 1970 - 2099	-*/

# Legend:

- \* = All values,
- ? = No specific value,
- , = Additional value specification,
- / = Increments specification,
- = Range specifier.

The following table contains examples of cron expressions used in job scheduling.

Expression	Meaning
0 0 12 * * ?	Fire at 12pm (noon) every day.
0 15 10 ? * *	Fire at 10:15 am every day.
0 15 10 * * ?	Fire at 10:15am every day.
0 15 10 * * ? *	Fire at 10:15am every day.
0 15 10 * * ? 2006	Fire at 10:15am every day during the year 2006.
0 * 14 * * ?	Fire every minute starting at 2 pm (14:00) and ending at 2:59 pm (14:49), every day.
0 0/5 14 * * ?	Fire every 5 minutes starting at 2 pm (14:00) and ending at 2:55 pm (14:55), every day.
0 0/5 14,18 * * ?	Fire every 5 minutes starting at 2 pm (14:00) and ending at 2:55 pm (14:55), AND fire every 5 minutes starting at 6 pm (18:00) and ending at 6:55 pm (18:55), every day.
0 0-5 14 * * ?	Fire every minute starting at 2 pm (14:00) and ending at 2:05 pm (14:05), every day.
0 10,44 14 ? 3 WED	Fire at 2:10 pm (14:10) and at 2:44 pm (14:44) every Wednesday in the month of March.
0 15 10 ? * MON-FRI	Fire at 10:15 am every Monday, Tuesday, Wednesday, Thursday and Friday.
0 15 10 15 * ?	Fire at 10:15 am on the 15th day of every month.
0 15 10 L * ?	Fire at 10:15am on the last day of every month.
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month.
0 15 10 ? * 6L 2003-2006	Fire at 10:15 am on every last Friday of every month during the years 2003, 2004, 2005 and 2006.
0 15 10 ? * 6#3	Fire at 10:1 5am on the third Friday of every month.

# **History Indexing job**

he  $\tt DB\ History\ Creator\ job\ indexes\ the\ embedded\ SQL\ database\ layer,\ updating\ it\ with\ the\ most\ current\ history\ data\ from\ the\ repository.$ 

History Indexing's performance has been greatly enhanced and now runs automatically.

(It can still be suspended, or resumed from the **Monitor**.)

Administrators that would like to have the job run periodically, at a scheduled time, can remove the disabled="false" property from the Cron Scripts via the

# OScheduler.

# Set the History Indexing job to run periodically:

# To start the job at 7pm (Monday to Friday):

```
<job name="Resume DB History Creator" id="polarion.jobs.db.history" disabled="false"</pre>
cronExpression="0 0 19 ? * MON-FRI" scope="system"/>
```

# To stop the job at 8am (Monday to Friday):

```
<job name="Suspend DB History Creator" id="polarion.jobs.db.history.killer"</pre>
cronExpression="0 0 8 ? * MON-FRI" scope="system"/>
```

# **Permissions management**

This section provides reference information that can be useful when configuring user permissions and user roles.

# **Permissions Configuration**

#### **Note**

The information below is provided for programmers who may need to check permissions in some custom extension or script. Administrators should use the "User Management" Permissions Management topic of Administration to configure user permissions.

The global security configuration is stored in the /.polarion/security/permissions.xml file.

Permissions configuration can be done for each project, so there can also be a project-scope premissions.xml. The project-level permissions.xml file is stored in PROJECT FOLDER/.polarion/security/.

When parsing permissions, Polarion checks the project scope first and then, if needed, the global scope.

### Default user roles and permissions

User roles are roles that can be defined and assigned by administrators. You can access the list of user roles in Administration: User Management: Roles. In Repository administration, only the repository-scope roles are shown. In project administration, only project-scope roles are shown. In addition to user roles, there are "dynamic roles". These are built-in roles used for controlling permissions for various artifacts such as Work Items and Documents.

The following tables provide a high-level description of the default user permissions. Note that permissions can be customized in the User Management: Permissions Management topic of Repository or project administration.

Roles assigned in the Repository (global) scope apply to all projects. For example if a user is given the global user role, that user will have the role and all globally configured permissions of the role in every project.

Role Name	Description of permissions
admin	Repository administrator. All permissions are granted.
user	Repository user. Users with this role can log in and manage their own account properties. A user with only this role, without any project role(s), cannot access projects. Note that login permission is tied to the user role and will not work if assigned to other roles watcher, for example.

Role Name	Description of Permissions	
project admin	User granted access to Administration for the project in which the role is assigned. User has full administrator permissions within the scope of the project. User cannot access Administration in other projects where this role is not assigned him/her. User cannot access Repository Administration unless granted the role <b>admin</b> in that scope.	
project_uu	Note  A user assigned that permission does not need to have the project_admin role assigned in any project.	
project_approver	Project-scope permission to review project content, including approving/disapproving Work Items, commenting Work Items and Documents, and resolving own comments. Users assigned this role for a project appear in the pick list of users who can be added as an Approving User in the Approvals section of the project's Work Items (available in the Table view of Work Items).	
project_assignab	project_assignable Project-scope permission to have Work Items assigned. Users with this role appear in the pick list of users in the Assignee field of Work Items. By default this role has broad permissions allowing users to create, read, and modify project content, including browsing and running reports and builds.	
project_user	User is granted limited access to the project. User can read all objects but not modify or create them. User can download builds and view reports, but cannot initiate builds or refresh reports. Note that login permission for projects is tied to the user role and will not work if assigned to other roles project_watcher, for example.	

For information on configuring roles, see: Configure User Roles.

# Dynamic roles and permissions

Dynamic roles are built into Polarion for the purpose of controlling permissions for various operations on Work Items and Documents. For example, there is a dynamic role **author** for both of these artifact types. It has a default set of permissions, and an administrator can grant or revoke various permissions to/from the role if the defaults are not what is needed. The Document or Work Item's

author will generally have broader permissions, DELETE permission for example, than users who have other roles.

Dynamic roles are dynamic in the sense that they are dynamically assigned to users by the system. For example, the creator of a Document is automatically assigned the **document\_author** role, and the author of a comment automatically gets the **comment\_author** role.

#### **Note**

Dynamic roles only work with permissions, not with workflor required roles.

Administrators can modify permissions for dynamic roles in Global or project scope, in

Administration User Management Permissions Management, in the By

Permission tab. Dynamic roles appear in the Applicable Roles section of each Permission. For example, the author role appears in the MODIFY permission of Work Items (i.e. the author can modify the Work Item), but not in the CREATE NEW permission (there is no author until a Work Item is created).

The following sections outline the dynamic roles and their default permissions.

#### **Dynamic roles for Documents:**

Dynamic Role	Permissions	
	Documents - Permission to READ	
	Documents - Permission to MODIFY FIELDS	
	Documents - Permission to MODIFY CONTENT	
document_author	Documents - Permission to MANAGE	
	Documents - Permission to DELETE	
	Documents - Permission to COMMENT	
	Documents - Permission RESOLVE COMMENT	
comment_author	Documents - Permission to RESOLVE COMMENT	

## **Dynamic roles for Pages:**

Dynamic Role	Permissions
	Pages - Permission to READ
Page-Author	Pages - Permission to DELETE
	Pages - Permission to MODIFY

# **Dynamic roles for Work Items:**

Dynamic Role	Permissions	
	Work Items - Permission to READ	
	Work Items - Permission to DELETE	
author	Work Items - Permission to MODIFY	
	Work Items - Permission to COMMENT	
	Work Items - Permission to RESOLVE COMMENT	
	Work Items - Permission to READ	
assignee	Work Items - Permission to DELETE	
	Work Items - Permission to MODIFY	
comment_author	Work Items - Permission to RESOLVE COMMENT	

# **Dynamic roles for Work Item fields:**

Dynamic Role	Permissions
author	Field [FIELD NAME] - Permission to READ
autiloi	Field [FIELD NAME] - Permission to MODIFY
	Field [FIELD NAME] - Permission to READ
assignee	Field [FIELD NAME] - Permission to MODIFY

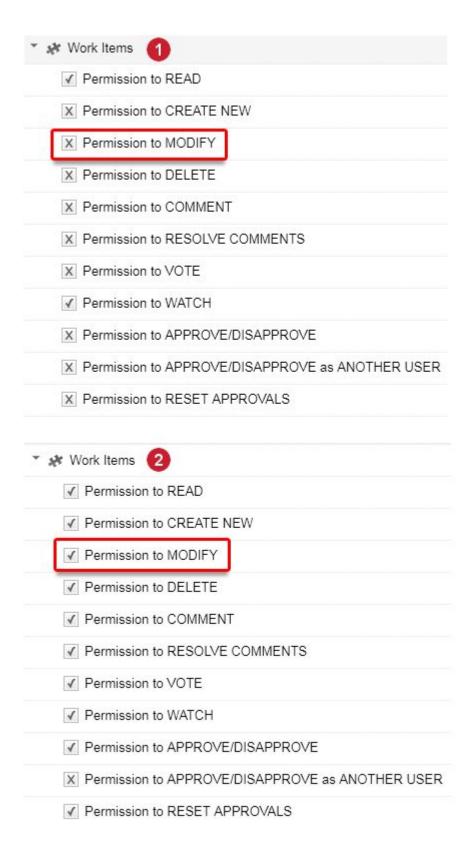
# Other dynamic roles:

Dynamic Role	Permissions	
lead	Projects - Permission to VIEW	
self	Accounts - Permission to MODIFY OWN ACCOUNT (Global scope only)	
33.1	Accounts - Permission to MODIFY OWN TIME-SPLIT ASSIGNMENTS (Global scope only)	

# Permission processing scheme

When Polarion processes user permissions, it observes the following rules:

- 1. A user may have multiple roles assigned that can be applicable in a given context. Roles can be global, project, or dynamic (e.g. author of a Work Item). Permissions for all roles are evaluated together; no role has precedence over others.
- 2. Project permission overrides global. For example, if the global configuration GRANTS the permission, but project configuration DENIES the same permission, the permission is denied in the project scope.
- 3. Custom Set permission overrides the generic permission. For example, if a Custom Set for Documents configuration GRANTS the permission to MANAGE for the "inReview" status, but the generic configuration DENIES the permission to MANAGE, a user is only able to manage Documents that have the "inReview" status.
- 4. If GRANT / DENY permissions are both assigned to a user on the same level, then GRANT takes precedence.
  - a. A user is assigned both a project\_user role and a project\_assignable role on the Froject
  - b. The project assignable role gives them permission to edit (MODIFY) Work Items, but the project user role does not. (See screeenshot below.)
  - c. Then the GRANT permission of the project assignable role would override the DENY permission of the project user role, and they would be able to edit Work Items.



- project\_user Role
- project\_assignable Role

See Configure User Permissions for more details.

## Tip

To avoid any role conflicts and confusion, it's a good idea to create a wide variety of roles to minimize users with more than one.

## Note

The Global admin role always has all permissions granted in all projects, and this cannot be overridden. Consequently, you should be careful about how widely you assign that role.

# Fields with READ permission

Administrators can configure the READ permission for Work Item fields in Administration: User Management > Permissions Management: Work Items > Fields. The following Work Item fields have the READ permission, enabling administrators to grant or deny read access to different user roles at the field level:

- Assignee
- Attachments
- Author
- · Boolean custom fields
- Categories
- Currency custom fields
- · Date custom fields
- Date Time custom fields
- Description
- Due Date
- Duration custom fields
- Enum custom fields
- Enum Multi-value custom fields

- Float custom fields
- Hyperlinks
- Initial Estimate
- Integer custom fields
- Linked Revisions
- Planned End
- Planned In
- Planned Start
- Planning Constraints
- Priority
- Remaining Estimate
- Resolution
- Resolved On
- · Rich Text custom fields
- Severity
- Status
- String custom fields
- Test Steps custom fields
- Text custom fields
- Time custom fields
- Time Point
- Time Spent
- Work Records

The following fields do not have configurable READ permission, and are therefore always readable for all users:

- 1. Created
- 2. Linked Work Items
- 3. Project

- 4. Title
- 5. Type
- 6. Updated

# Fields with MODIFY permission

Administrators can configure the MODIFY permission for Work Item fields in **Administration** → User Management → Permissions Management → Work Items → Fields. The following Work Item fields have the MODIFY permission, enabling administrators to grant or deny write access to different user roles at the field level:

- Assignee
- Attachments
- Boolean custom fields
- Categories
- Currency custom fields
- Date custom fields
- Date Time custom fields
- Description
- Due Date
- **Duration custom fields**
- Enum custom fields
- Enum Multi-value custom fields
- Float custom fields
- Hyperlinks
- **Initial Estimate**
- Integer custom fields
- Linked Revisions
- Linked Work Items
- **Planning Constraints**
- Priority
- Remaining Estimate

- Resolution
- Resolved On
- · Rich Text custom fields
- Severity
- Status
- String custom fields
- Test Steps custom fields
- Text custom fields
- Time custom fields
- Time Point
- Time Spent
- Title
- Type
- Work Records

The following fields do not have configurable MODIFY permission, and are therefore not modifiable by users:

- 1. Author
- 2. Created
- 3. Planned End
- 4. Planned In
- 5. Planned Start
- 6. Updated

## Impact on Round-trip operations

When a user is denied permission to MODIFY any Work Item fields, there can be several impacts on Work Item round-trip operations.

## In round-trip import:

• If the user re-importing a round-trip document is denied MODIFY permission for fields that have been changed externally in the round-trip document, those changed fields are ignored by the importer and a message is displayed.

- If the re-imported file contains new Work Items which have fields for which the re-importing user
  is denied MODIFY permission, the file is imported but the MODIFY-denied fields are ignored
  and a warning message is displayed.
- If the re-imported file contains new Work Items which have required fields for which the
  re-importing user is denied MODIFY permission, the import operation fails and a message is
  displayed.
- If a user exported items and had MODIFY field permissions at the time of export, and that permission is subsequently denied before re-import, the re-import is allowed, but the MODIFY-denied fields are not updated in the portal, and a message is displayed.
- In all cases, the logs of the import job specify any fields that were not updated by the import operation.

## In round-trip export:

• If the exporting user is denied MODIFY permission for some fields, those fields are read-only in the exported file.

# In round-trip import:

• **Important**: If import is performed as another user that has more MODIFY permissions than the user who performed the export (admin, for example), then all data are replaced.

# **Default Enumeration configuration files**

This section lists the default enumeration configuration files and the Work Item fields they provide list item values for. Keep in mind that project templates may have different default configuration files for template-specific Work Item types, fields, etc.

## Repository scope

Work Item Field	Configuration Filename	Work Item Type(s)	
Hyperlinks: Role	hyperlink-role-enum.xml	All	
Priority	priority-enum.xml	All	
Resolution	resolution-enum.xml	All	
Resolution	requirement-resolution-enum.xml	Requirement	
Risk	risk-enum.xml	All	
Severity	severity-enum.xml	All	
Severity	requirement-severity-enum.xml	Requirement	
Status	status-enum.xml	All	
Status	requirement-status-enum.xml	Requirement	

Work Item Field	Configuration Filename	Work Item Type(s)	
Link Role	workitem-link-role-enum.xml	All	
Type (of Work Item)	workitem-type-enum.xml	All	
Work Records: Type	work-record-type-enum.xml	All	

# **Notification events and targets**

The section provides reference information about events and targets for notifications which is useful when customizing notifications.

#### **Notification events**

Events occur when things change in the system... a user updates a Work Item, or a build finishes, for example. Notifications are sent to users in response to events. The tables in this section list and describe key notification events. In the configuration files, notification events are defined between <notification-config></notification-config> tags, and notification targets are inside events. The following events (and appropriate values of the <event\_name> tag) are defined:

Event	Description	Parameters	
workitem-created	Occurs when a new Work Item is created.	filter	
workitem-deleted	Occurs when a Work Item is deleted.	filter	
workitem-commented	Occurs when a comment is added to a Work Item.	filter	
workitem-comment-removedrs when a comment is removed from Work Item.		filter	
workitem-assigned	Occurs when the Work Item Assignee field changes from <i>None</i> to some other value.	filter	
workitem-updated	Occurs when some other (unrecognized) change is made to a Work Item. Also triggered when another Work Item gets a link to the one affected by this event.	fields, exceptFields, filter	
workitem-status-chan	gedccurs when a workflow action is executed on a Work Item.	statusFrom, statusTo, filter	
workitem-attachment-	addedurs when an attachment is added to a Work Item.	filter;	
workitem-attachment-	updated when an existing Work Item attachment is updated (i.e. new version of same attachment file is uploaded).	filter	

Event	Description	Parameters
workitem-attachment-	rechrowes when an existing attachment is removed from a Work Item.	filter
workitem-linked	Occurs when a specified Work Item is added to, or removed from another Work Item's set of linked Work Items. Can be configured per link role.	roles, filter
linked-workitem-upda	ted ccurs when another Work Item, linked from the specified one, is updated. For example, WI-1 links WI-2. When WI-2 gets updated, this event is triggered, allowing the WI-1 assignee to be notified. Can be configured per link role.	fields, exceptFields, roles, filter
backlinked-workitem-	to the specified Work Item, which links to the specified Work Item, is updated. For example, WI-1 links WI-2. If WI-1 gets updated, then this event is triggered, allowing the W-2 assignee to be notified. Can be configured per link role.	fields, exceptFields, roles, filter
workitem-watcher-ado	eoccurs when a watcher is added to a Work Item.	
workitem-watcher-rem	<b>o⊘ed</b> urs when a watcher is removed from a Work Item.	
workitem-vote-added	Occurs when a user adds a vote for a Work Item.	
workitem-vote-remove	dOccurs when a user removes a vote from a Work Item.	
workitem-approval-ad	d <b>e</b> ccurs when a user is added to the Approvals section of a Work Item.	status, filter
workitem-approval-ch	ar@ædurs when a user is changed in the Approvals section of a Work Item.	statusFrom, statusTo, filter
workitem-approval-rei	n <b>Occed</b> irs when a user is removed from the Approvals section of a Work Item.	status, filter
workitem-workrecord-	a@bedrs when a work record is added to a Work Item.	filter
workitem-workrecord-	reproved when a work record is removed from the Work Item.	filter
revision-linked	Occurs when a repository revision is linked to the Work Item.	filter

The filter parameter is a sub-string of a query, the first part of which is supplied by the system when the filter is applied during a relevant event. The system supplies id: "x" AND, where "x" is the value of a Work Item ID. Therefore, the expression you supply in the filter parameter can start with the NOT operator. If the filter expression uses the OR operator, then it must be surrounded by parentheses.

# **Examples:**

- filter="(type:task OR type:defect)" in effect, amounts to: filter="id:[workitemid] AND (type:task OR type:defect)"
- filter="NOT (type:task OR type:defect)" in effect, amounts to: filter="id:[workitemid] AND NOT (type:task OR type:defect)"

Event	Description	
document-commented	Occurs when a comment or comment reply is added in a Document.	
document-comment-remov	eveoccurs when an existing comment is removed from a Document.	
document-created	Occurs when a Document is created.	
document-deleted	Occurs when a Document is deleted.	
document-moved	Occurs when a Document is moved or renamed.	
document-status-changed	Occurs when the Status field of a Document changes (assuming Document workflow is configured).	
document-updated	Occurs when a Document is updated.	

Event	Description	
plan-created	Occurs when a Plan is created.	
plan-updated	Occurs when a Plan is updated.	
plan-deleted	Occurs when a Plan is deleted.	
plan-status=changed	Occurs when the value of a Plan's <b>Status</b> field value changes.	
plan-workitem-added	Occurs when a Work Item is added to a Plan, after Plan has started.	
plan-workitem-removed	Occurs when a Work Item is removed from a Plan, after Plan has started.	

Event	Description	
testrun-status-changed	Occurs when the status of a Test Run changes, or when a new Test Run is created.	
testrun-deleted	Occurs when a Test Run is deleted.	

Event	Description	
testrun-updated	Occurs when a Test Run is updated.	
testrun-finished	Occurs when the a value (date) is set for the <b>finishedOn</b> field of a Test Run.	

Event	Description
user-created	Occurs when a new user is created.
user-updated Occurs when an existing user is updated.	
user-deleted	Occurs when an existing user is deleted.

Event	Description	
build-finished	Occurs when a project build has finished running.	

# **Notification targets**

Target	Description	Event Type(s)	Parameters
affected-approved	When approval field is changed then only affected approvee(s) (added, removed, changed state) will be notified.	Work Item	
affected-user	Account owner will be notified about changes. This target works only for events user-created, user-updated, and user-deleted.	User	
all-commenters	All users who have commented a Work Item or a Document.	Work Item, Document	
all-voters	All users who have voted for the Work Item.	Work Item	
all-watchers	All users who have been set as watchers of the Work Item.	Work Item	
awaited-approve	<b>s</b> All users whose approval is awaited for the Work Item.	Work Item	
comment-thread-	palfsieripathto commented in a threaded comments discussion will be notified if someone replies to the comment in the appropriate thread.	Work Item	
current-assignee	The assignee of the item. If assignee is changed during the action, both the original and the new assignee will receive notification.	Work Item	

Target	Description	Event Type(s)	Parameters
current-author	The author of the Work Item.	Work Item	
current-user	The currently logged-in user (the one who has made the change).	Work Item, Module, User	
global-role	Users having the global role specified in the required role attribute will be notified. (Example: global-role role="administrator")	Work Item, Module, User, Build	role
new-assignee	When assignee field is changed then only new assignee(s) will be notified.	Work Item	
project-lead	The user defined in the project's properties as the leader of the project to which the event belongs.	Work Item, Module, Build	
project-members	All users who have any role in the project to which the event belongs.	Work Item, Module, Build	
project-role	Users having the project-scope role specified in the required role attribute will be notified (Example: project-role role="project_assignable")	Work Item, Module, User, Build	role
single-email	Notification will be sent to the specified e-mail address in the <b>email</b> attribute (Example: email="someone@somewhere.org")	Work Item, Module, User, Build	email
single-user	Notification will be sent to the user specified in the <i>user-id</i> attribute (Example: user-id="polarion_user_id")	Work Item, Module, User, Build	user-id

Notifications are prioritized so that only one notification is sent to each user about a change. For example, adding an attachment and a comment will result in just one attachment notification if email is the target of both. However the one notification will cover all changes.

Users falling into 2 or more categories will receive only one notification. For example, both assignee and current user. Please note that not all targets may be used for every event. For example, it makes no sense to use the **assignee** target for a **build-finished** event.

# System variables

The following system variable are available and can be referenced in configuration of custom notification email subject lines.

Variable	Description	
\$EVENT_NAME\$	General event name.	
\$EVENT_USER_ID\$	The ID of the user whose action triggered the event.	
\$PROJECT\$	The project name that appears in the Subject line of email notifications.	
\$SERVER_PREFIX\$	The prefix set in global notification settings.	

Variable Description		
\$BUILD_STATUS\$ Status of a build (e.g. failed, aborted, ok)		
\$BUILD_PROJECT_NAMESne of the project for which build was started.		
\$BUILD_ARTIFACT_LABEDSan-readable build artifact label (short description).		
\$BUILD_STAMP\$ Build's time stamp.		
\$BUILD_BY_NAME\$ Name of the user who started a build.		

Variable	Description	
\$USER_EVENT_NAME\$Name of an event, e.g. "User Updated".		
<b>\$USER_NAME\$</b> Name of the user who has been changed.		
\$USER_BY_NAME\$ Name of the user who evoked the notification.		

Variable	Description		
\$WI_EVENT_NAME\$	The name of the event triggering the notification, or new status if status has been changed.		
\$WI_ID\$	The ID of the changed Work Item.		
\$WI_TITLE\$	The new title of the changed Work Item.		
\$WI_ASSIGNEES\$	The new assignees of the changed Work Item.		
\$PROJECT\$	Holds the project name that appears in the Subject line of email notifications.		

Variable	Description	
\$MODULE_EVENT_NAMES ne of the module event triggering the notification.		
\$MODULE_NAME\$	Name of the module that has changed.	

Variable	Description
\$MODULE_BY_NAME	\$ Name of the user who evoked the notification.

# **Default notification priorities**

The following table lists the default priorities for notifications.

Event name	Priority
workitem-commented	0
workitem-comment-removed	0
workitem-workrecord-added	0
workitem-workrecord-removed	0
workitem-deleted	100
workitem-attachment-added	100
workitem-attachment-updated	100
workitem-attachment-removed	100
workitem-updated	200
workitem-approval-added	300
workitem-approval-removed	300
workitem-approval-changed	300
workitem-assigned	400
workitem-status-changed	500
workitem-created	600
workitem-linked	100
linked-workitem-updated	0
backlinked-workitem-updated	0
workitem-vote-added	100
workitem-vote-removed	100
workitem-watcher-added	0
workitem-watcher-removed	0

Event name	Priority
revision-linked	0
build-finished	100
module-created	700
module-deleted	700
module-updated	700
user-created	0
user-updated	0
user-deleted	0

# **Project template properties**

Project template properties can be specified for custom project templates you create.

#### **Note**

## If custom templates are used:

To have Polarion replace the Work Item Prefix from template-provided Work Items, add the following lines to your template.properties file:trackerPrefixToReplace=PREFIX, where PREFIX is the Work Item prefix specified in the project template's development project. For example, if your development project's Work Item prefix is "MYTEMPL", then you should specify trackerPrefixToReplace=MYTEMPL in the properties file.

process=.polarion/tracker/fields/workitem-type-enum.xml

These lines are added automatically in templates that ship with Polarion.

For information on project templates and creating custom project templates, see Customizing Project Templates

Each project template may contain a file named template.properties in its root. Use of this file is not required, but is recommended. The file has a standard Java properties file format. The following properties are recognized:

## name

Short template name

## description

Longer template description

#### parameters

Names of parameters required for project creation. Separated by ; (semicolon). The parameter names might contain letters, - (dash), and \_ (underscore). Names are case sensitive. The following parameters are predefined:

- project-id
- project-name
- tracker-prefix
- trackerPrefixToReplace

#### skip

List of files/directories to exclude from copying to a new project's directory tree. (template.properties is excluded automatically). Names cannot start with a / (forward slash).

#### process

List of text files where the parameters will be substituted during project creation. Files are separated by a ; (semicolon). Names cannot start with a / (forward slash)

## param.PARAM\_ID.name

Human readable names of parameters defined in the parameters property

## template.properties file example

```
name=Abc Software Project
description=Standard software development project for ABC-Corp, Inc. Development Teams
process=.polarion/polarion-project.xml
process=.polarion/tracker/fields/workitem-type-enum.xml
parameters=project-name
param.project-name.name=Project Name
```

Other files/directories in the template directory are copied to the new project directory during the creation process (except those listed in skip property). Files listed in the process property are assumed to be text files. During the project creation process, all strings having the form <code>%param-id%</code> are replaced by the actual parameter value. For parameters defined in the parameters property, there is also a predefined parameter <code>project-id</code>, which contains the ID of newly created project.

# **PDF Export Header/Footer Properties**

This section lists the properties which can be specified in the PDF Export configuration to embed data values in the header/footer of PDF documents created by exporting a Page (including Classic Wiki) or a Document. Properties may render some stored data value such as revision number or project name, or a calculated value such as total number of pages.

The following table lists the available properties. Note that the **Code** column contains the string that should be specified in the configuration file. For example, you should specify the property code <code>\$[projectName]</code> in the configuration, as opposed to the property name "ProjectName".

Name	Code	Description	
Revision	\$[revision]	The revision number of the exported Page or Document.	

Name	Code	Description
revision2	\$[revision2]	For use when exporting revision comparison of a Document to PDF. This renders the revision number of the newer of 2 compared revisions.

It is also possible to reference Velocity variables and page parameters in the header/footer configuration. See Configuring PDF Export.

# **Custom field types**

This topic provides a reference for the field types supported in the Polarion custom fields configuration. You can assign any of these types to a custom field.

For information on defining custom fields, see: Configure Custom Fields.

# **List of basic Types**

- String (single line)
- Text (multiple lines)
- Rich Text (multi-line rich text)
- Integer
- Float
- Currency
- Time
- Date
- Date time
- Duration
- Boolean
- **Enum**: (An existing Enumeration must be selected. See: Configure Enumerations.)
- **Test Steps** (Renders a table of Test Steps in Work Items of the selected type.)

### Tip

Check out the custom field and enumeration configurations in the demo projects included in product distributions.

# **Enumeration types**

The following enumeration types encapsulate data from the Polarion portal into lists, which can be specified as the value for Enum type custom fields.

- ID The type ID which can be referenced in scripts and API calls.
- **UI Name** The type Name that appears in the pick list for Enum type custom fields in the Custom Fields editor in Administration.
- **Data** Data encapsulated by the type. Parameters are referenced in parentheses in scripts or API calls, as shown, and in labeled text box widgets in the Custom Fields configuration user interface.

ID	UI Name	Data
@build	Build	When no query is specified, all builds visible in the current project.
@build[QUERY]	Build	All builds found by the query (using Lucene syntax) specified in the <b>Query</b> text box (or the QUERY parameter).
@category	Category	All Categories configured for the current project.
@document	Document	When no query is specified, all LiveDoc Documents visible in the current project.
@document[QUERY]	Document	All LiveDoc Documents found by the query (using Lucene syntax) specified in the Query text box (or the QUERY parameter).
@plan	Plan	All Plans in the project. Not user-sortable.
@plan[QUERY]	Plan	All Plans found by the query (using Lucene syntax) specified in the <b>Query</b> text box (or the QUERY parameter)
@project	Project	When Project Group is not specified, all active projects in the system.
TotalPages	\$[total]	Total number of pages in the PDF document.
PolarionProductName	\$[productName	Name of the Polarion product used to export the PDF document.
PolarionVersion	\$[productVers	⊌i Wersion of the Polarion product used to export the PDF document.
ProjectId	\$[projectId]	ID of the project from which the content was exported to PDF.
ProjectName	\$[projectName	Name of the project from which the content was exported to PDF.

ID	UI Name	Data
Space	\$[space]	System ID of the space from which the content was exported to PDF.
SpaceTitle	\$[spaceTitle]	Title of the space from which the content was exported to PDF.
Created	\$[created]	Date when content was created in Polarion.
CreatedBy	\$[createdBy]	User name of the user who created the content in Polarion.
Updated	\$[updated]	Date when the content was past updated in Polarion .
UpdatedBy	\$[updatedBy]	User name of the user who last updated the content in Polarion.
Generated	\$[generated]	Date when the PDF document was generated (i.e. exported).
GeneratedBy	\$[generatedBy	User name of the user who generated (i.e. exported) the content to PDF.
revision1	\$[revision1]	For use when exporting revision comparison of a Document to PDF. This renders the revision number of the older of 2 compared revisions.
revision2	\$[revision2]	For use when exporting revision comparison of a Document to PDF. This renders the revision number of the newer of 2 compared revisions.

# **Build configuration file format**

This topic gives the basic file format of the artifacts.xml file which is used to configure building

target="TARGET LOCATION IN LOCAL DEPLOYMENT SPACE"

```
recurse="WHETHER_TO_USE_FOLDER_CONTENTS_AS_A_RESOURCE"/>
... <resource> ...
  </resources>
... additional type-specific elements ...
  </artifact>
  <artifact>
    ...
  </artifact>
  </artifact>
</artifact>
</artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifact></artifacts></artifacts>
```

# Work report Job configuration

You can automate Work Reports by configuring a scheduled job to run them. This section documents the report parameters and provides an example configuration.

The job is configured in .polarion/jobs/schedule.xml (Access this using the Repository Browser).

#### **Job Parameters**

- Scale: day, week, or month
- Grouping: comma-separated list of available groupings:
  - o project
  - o projectGroup
  - o category
  - o user
  - o workItem
  - o type
- periodStart, periodFinish date specification in one of the following formats:
  - o date:yyyy-MM-dd
  - o timePoint:TIMEPOINT\_ID
  - o firstTimePoint
  - o lastTimePoint
  - o projectStart
  - o projectFinish

# Configuration example

```
<title>Work Report</title>
  <scale>week</scale>
  <grouping>project,user</grouping>
  <query></query>
  <periodStart></periodStart>
  <periodFinish></periodFinish>
</job>
```

- fileName can be specified with date-time format. For example: work\_report\_{yyyy-MM-dd}.xls. The format can be any Java SimpleDateFormat. The actual file name will be created replacing the {FORMAT} part with the formatted current date.
- If overwrite attribute is False, and a work report with the same file name already exists, the job will fail.
- Only work records from Work Items matching the query and dated within the specified period are exported
- If query is not specified, both periodFinish and periodStart must be specified

# **Workflow Conditions and Functions for Work Items**

#### **Workflow Conditions for Work Items**

This section documents the conditions available for configuring the workflow for Work Items. See functions for a list of available functions available for configuring the workflow for Work Items. For procedural information, see Configuring Work Item Workflow. It also describes system handling of missing or invalid conditions/functions in the configuration file. See also workflow Conditions and Functions for Documents. Workflow Conditions and Functions for Documents.

### **ApprovalState**

- Description: Tests whether all Work Item approvals have a specific workflow status.
- Parameters:
  - o CommonApprovalState the status value to be tested for... 'approved' or 'waiting', for example.

    If omitted, the condition testes for the 'approved' status. That is, whether all approvals have the 'approved' workflow status.

#### **AtLeastOneApprovedAndNooneDisapproved**

- **Description:** Tests that at least one user has logged approval of the Work Item AND no user has logged disapproval.
- Parameters: NONE

### **FieldNonEmpty**

• **Description:** Tests whether a Work Item field is empty or non-empty. Field is empty if it is missing, or if its value is null, or it is an empty collection, or an empty text (including Rich Text), or false Boolean value.

#### Parameters:

- field.name the field ID of the Work Item field to check.
- negate Possible values: "true" or "false". If true, the condition will be passed if the field is empty. If false (the default value), the condition will be passed if the field is non-empty

#### LinkedWorkItem

**Description:** Checks, that there is at least one linked (or backlinked) item (optionally with given link role and type).

#### Parameters:

- link.role (optional): Required link role.
- backlink (optional): If true, then backlinks are checked.
- target.work.item.type (optional): Required linked Work Item type.

### LinkedWorkItemsStatus

Description: Traverses incoming and outgoing links of defined roles and passes if all target Work Items are in one of valid states.

### Parameters:

- o back.link.roles: comma separated list of incoming link roles to check.
- link.roles: comma separated list of outgoing link roles to check.
- valid.states: comma separated list of valid states.

0

## LinkedWorkItemsType

Description: Checks that there is a linked or backlinked Work Item. Optionally, required link role and target Work Item type may be specified.

#### Parameters:

- backlink: true or false (optional, default is false)
- link.role: required role of the link (optional, e.g. "implements")
- target.work.item.types: required target Work Item type (optional, e.g. "defect")

#### RoleCondition

**Description:** Allows the action only if the current user has at least one of the roles specified in the roles parameter.

## Parameters:

o roles - Comma separated list of role names.

# ScriptCondition

• **Description:** Evaluate whether or not a specific workflow action is available according to the return value of a custom script, which is the result of the last line of the script. If the script returns Boolean value false, then the workflow action will not be available. If it returns some text, then the workflow action will not be available and the returned text will be shown as the reason.

#### Parameters:

- o engine: The name of the scripting engine to use. Groovy (groovy) and JavaScript/ECMAScript (js) are supported.
- o script: The name of the script to run as condition. (Script file must be stored in the scripts folder of the installation's file system.)

**Example**: Given the existence of a Javascript script with line "Test".equals(workflowContext.workItem.title); ...

The workflow condition will be fulfilled if the title of the Work Item is "Test".

## For Custom Scripts:

Custom scripts must be placed in a "scripts" sub directory in the \$POLARION\_HOME directory.

(C:/Polarion on Windows systems and /opt/polarion/ on Linux based systems by default, but may vary.)

If no scripts directory exists, create one.

(Administrators of UNIX/Linux based systems will require **Owner**, **User** and **Group** privileges and be set up as a Polarion **System User**.)

#### Note

For Developers: An instance of <code>com.polarion.alm.tracker.workflow.ICallContext</code> is available as variable <code>workflowContext</code>, and <code>com.polarion.alm.tracker.workflow.IArguments</code> is available as variable arguments in the script. Also, it is possible to configure more parameters of the condition than <code>engine</code> and <code>script</code>, which are then accessible through the arguments variable in the script.

#### **Workflow Functions for Work Items**

This section documents the available workflow **Functions**. See the **conditions** for configuring the workflow for Work Items.

## AddDefaultApprovals

 Description: Adds default approvals for the Work Item (i.e., sets approval status to 'waiting' for a specified list of users)

#### Parameters:

o approvals.users: comma separated list of user IDs comprising the default list of approvals that are added to the Work Item when executing a workflow action.

- approvals.roles: comma separated list of role IDs comprising the default role(s) for approvals. All users with a specified role are added to the Work Item when executing a workflow action.
- Required Permission: Work Items: Permission to MODIFY (See Configuring User Permissions for more information).

# **Changed Field**

**Description:** Changes value of specified field. Only setting of a date type field to current date, or a date-time type field to the current date-time is supported. The value type set depends on the type of the target field.

## **Parameters:**

- field.name: the field ID of the field to be changed
- field.value.type: only "current.date" is supported. It sets either date or date-time value depending of the type of the target field in specified in field.name.

#### ClearField

- **Description:** Clears existing value in the specified Work Item field.
- **Parameters:** 
  - field.name field ID of the field to clear. Field should not be a required field or an error can result.

#### CreateLinkedWorkItem

**Description:** Creates a linked or backlinked Work Item using a specified type, link role, and a set of categories. The new Work Item is auto-assigned.

## Parameters:

- type (required): Type of the new Work Item to be created.
- link.role (required): Role of the link between original and new Work Item.
- backlink (optional): if specified, the new Work Item is backlinked, otherwise it is linked.
- categories (See note below): comma separated list IDs of the Categories to be applied to the new Work Item.

#### Note

Whether or not this parameter is required depends on the general workflow configuration, which may or may not be set to require that Categories be specified when a new Work Item is created. The parameter is optional in the default global configuration.

## LinkedWorkItemsStatusChange

- **Description:** Traverses incoming and outgoing links of defined roles, finds all target Work Items and checks if they are in one of valid states. Those Work Items, that are not in one of the valid states, are put to the target state and optionally some of their fields are cleared.
- Parameters:
  - o link.roles: comma separated list of outgoing link roles to check
  - o back.link.roles: comma separated list of incoming link roles to check
  - o valid.states: comma separated list of valid states
  - o target.state: status to which found Work Items will be set
  - o clear.attribs: attributes that should be cleared for found Work Items

## MarkWorkflowSignaturesAsObsolete

• **Description:** Removes all of the Work Item signatures signed on the statuses that precede the one you've defined it for.

**Example:** If defined for say **Reopen**, then all workflow signatures that **Reviewed** and **Approved** the item, are removed if the Work item is **reopened**.

(The removed signatures are still tracked in the Work Item's History.)

- Parameters: None
- **Usage:** A workflow status was signed off on, but needs amendments that invalidate the signature(s).

## ResetApprovalsState

- **Description:** Resets all approvals for the current Work Item to the 'waiting' state.
- Parameters: NONE
- Required Permissions: 'APPROVE/DISAPPROVE as ANOTHER USER' or 'RESET APPROVALS'.

(Configured in Administration: See Configuring User Permissions for more information).

#### ScriptFunction

• **Description:** Invoke a custom script to perform some action(s) in conjunction with a workflow transition.

## Parameters:

- o engine: The name of the scripting engine to use. Groovy (groovy) and JavaScript/ECMAScript (js) are supported.
- o script: Name of the script run when the Work Item action is invoked. (Script file must be stored in the *scripts* folder of the installation's file system.)

reason: A message string to return if the condition is not passed. If not specified, a default string will be returned.

## For custom scripts:

Custom scripts must be placed in a scripts sub directory in the \$POLARION HOME directory.

(C:/Polarion on Windows systems and /opt/polarion/ on Linux based systems by default, but may vary.)

If no **scripts** directory exists, create one.

(Administrators of UNIX/Linux based systems will require **Owner**, **User** and **Group** privileges and be set up as a Polarion **System User**.)

#### Note

For Developers: An instance of com.polarion.alm.tracker.workflow.ICallContext is available as variable workflowContext, and com.polarion.alm.tracker.workflow.IArguments is available as variable arguments in the script. Also, it is possible to configure more parameters of the condition than engine and script, which are then accessible through the arguments variable in the script.

#### **SetDate**

**Description:** Sets the specified field to the current date OR date-time, depending on the type of the target field.

## Parameters:

o field.name - field ID of the field to write to. The field specified must be of a type which is compatible with a date type OR a date-time type value.

#### Handle workflow errors

Sometimes a workflow extension may be removed resulting in a missing or invalid condition and/or function which in turn impede the execution of some workflow action or actions. (Such situation can occur when the XML configuration file is edited directly and uploaded to the repository, as some advanced administrators prefer over using the Workflow Designer in Administration.) If such problems are introduced into the configuration file, Polarion handles it in the following ways:

- Errors resulting from missing workflow conditions or functions are written to the main Polarion log file.
- A user working in Polarion administration who attempts to edit a workflow configuration containing missing conditions/functions receives a warning dialog containing information about the missing conditions and/or functions.
- If a workflow function is missing, then the action for which it is configured will always be disabled and the transition will not be possible.

If a workflow condition is missing, then the action for which it is configured will not be disabled, but any attempt to save a Work Item after invoking such action will fail and an error will be logged.

# **Workflow Conditions and Functions for Documents**

#### **Workflow Conditions for Documents**

This section lists the workflow conditions available for configuring Document workflow. Also see the available functions for configuring the document workflow.

(For conditions and functions for Work Item workflow, see Conditions and Functions for Work Items.

This section documents the available workflow conditions for Documents.

#### RoleCondition

• **Description:** Allows the action only if the current user has at least one of the roles specified in the roles parameter.

#### Parameters:

o roles - Comma separated list of role names.

## ScriptCondition

Exactly the same as **ScriptCondition** for Work Items, except that it is configurable in Document workflow actions. See ScriptCondition for Work Items.

## FieldNonEmpty:

• **Description:** Tests whether a Document field is empty or non-empty. Field is empty if it is missing, or if its value is null, or it is an empty collection, or an empty text (including Rich Text), or false Boolean value.

#### Parameters:

- o field.name the field ID of the Document field to check.
- o negate Possible values: true or false. If true, the condition will be passed if the field is empty. If false (the default value), the condition will be passed if the field is non-empty

#### ContainsNoMatchingWorkItems

 Description: - if the Document contains any Work Items matching the query, then the workflow action cannot be executed

#### Parameters:

o query - Lucene query for Work Items that, if matched, will prevent the workflow action from executing

#### **Workflow Functions for Documents**

This section documents the available workflow *functions* for Documents. Also see the available conditions for configuring the document workflow.

## AddDefaultSigners

**Description:** Adds users to the list of invited signers for the transition to the selected Document status. Sets the signature status to "invited" for these users.

#### Parameters:

- signedBy.users: comma-separated list of one ore more user IDs. These concrete users are added as signers. Mandatory if signedBy.roles is not specified.
- signedBy.roles: comma-separated list of one ore more user roles (e.g. "project approver"). Users assigned any of the specified roles are added as signers. Mandatory if signedBy.users is not specified.
- signer.role: string, optional. Any meaningful string describing the organizational role of users signing a Document. E.g. "Signed for Legal Department".
- target.status.id: string, mandatory. ID of the target workflow status to which Document can transition after signature policy is satisfied. E.g. approved for the "Approved" status.

# ChangeField

**Description:** Changes value of specified Document field. Only setting of a date type field to current date, or a date-time type field to the current date-time is supported. The value type set depends on the type of the target field.

# Parameters:

- o field.name: field ID of the Document field to be changed
- field.value.type: only "current.date" is supported. It sets either date or date-time value depending of the type of the target field in specified in field.name.

### ClearField

**Description:**Clears existing value in the specified Document field.

## Parameters:

field.name - field ID of the field to clear. Field should not be a required field or an error can result.

#### **InvokeAction**

**Description:** Workflow transition is executed on all Work Items contained in the (filtered) Document. If any Work Item workflow action cannot be executed for some reason, then Document workflow action fails and user is informed.

#### Parameters:

- o statusId: terminal status of the workflow transition to be executed on the Work Items.
- actionId: action of the workflow transition to be executed on the Work Items.
- guery: query that additionally filters items contained in the Document.

## **MarkWorkflowSignaturesAsObsolete**

- **Description:** moves all workflow signatures in the specified states to state *Obsolete*, marks all verdict comments linked to the signatures as *Resolved*.
- **Parameter:** *signatureStates* default value is *"Done"*, but can be a comma-separated list if you do not want to mark some contexts (such as "Open") as obsolete.
- Usage: On a workflow action to reset/restart the review/signature process. For example:
  - o Document must be reworked, current signatures should not apply to new version.

#### Note

Before invoking this function, users should have manually processed all signature comments for already *Done* states, so there should be no unresolved comments for a transition for which signatures are being marked as obsolete.

## ResetSignaturesVerdict

 Description: Resets all signatures to "invited" status. An example usage would be when invited signers have declined to sign, requiring modification of a Document and start of a new round of reviews and signatures.

#### Parameters:

- o signatureStates default value is "Done", but can be a comma-separated list if you do not want to mark some workflow signature states (such as "Open") as obsolete.
- o target.status.id: string, optional. The ID of the target Status for the workflow transition. If not defined, then all signatures for unfinished workflow signatures will be reset, otherwise only signatures for specified status will be reset. The function accepts only single value as a parameter (a list of multiple parameters is not accepted).
- **Usage:** On a workflow action starts another round of the review/signature process after signature was declined and Document had to be reworked. For example:
  - o Document must be reworked because a signature for *Mark as Approved* action was declined, so signatures for that action need to be reset.

#### ScriptFunction

Exactly the same as *ScriptFunction* for Work Items, except that it is configurable in Document workflow actions. See *ScriptFunction* for Work Items.

### **SetDate**

• **Description:** Sets the specified Document field to the current date OR date-time, depending on the type of the target field.

## Parameters:

o field.name - field ID of the field to write to. The field specified must be of a type which is compatible with a date type OR a date-time type value.

# Workflow conditions and functions for Test Runs

### **Workflow conditions (Test Runs)**

This section documents the conditions for configuring the workflow for Test Runs. For procedural information, see: Configure Test run Workflow, Configure Document Workflow and Configure Work Item Workflow.

#### RoleCondition

- Description: Allows the action only if the current user has at least one of the roles specified in the roles parameter.
- **Parameter:** roles Comma separated list of role names.

## ScriptCondition

Exactly the same as ScriptCondition for Work Items, except that it is configurable in Test Run workflow actions. See ScriptCondition for Work Items.

## **FieldNonEmpty**

- **Description:** Tests whether a Test Run field is empty or non-empty. Field is empty if it is missing, or if its value is null, or it is an empty collection, or an empty text (including Rich Text), or false Boolean value.
- Parameter:
  - field name the field ID of the Test Run field to check.
  - negate Possible values: "true" or "false". If true, the condition will be passed if the field is empty. If false (the default value), the condition will be passed if the field is non-empty

#### **Workflow functions (Test Runs)**

This section documents the functions for configuring the workflow for Test Runs. For procedural information, see: Configure Test run Workflow, Configure Document Workflow and Configure Work Item Workflow.

This section documents the available workflow *functions* for Test Runs.

## MarkWorkflowSignaturesAsObsolete

- **Description:** Moves all existing workflow signatures related to the Test Run to Obsolete state.
- Parameter: None
- Usage: Some workflow action that was signed needs to be done again. For example, a Test Run was executed and signed, but something has occurred resulting in a decision to execute it again.

#### SetDate

**Description:** Sets the specified field to the current date OR date-time, depending on the type of the target field.

• **Parameter:** field.name - Field ID of the field to write to. The field specified must be of a type that is compatible with a date type OR a date-time type value.

## ScriptFunction

Exactly the same as ScriptFunction for Work Items, except that it is configurable in Test Run workflow actions. See ScriptCondition for Work Items.

# **Polarion log files**

This section documents the various log files, where to find them, and their purpose. It also provides information about cleaning up of old log files to conserve disk storage.

## **Location of Log Files**

The default location for log files is as follows:

- Linux: /opt/polarion/data/logs/main
- Windows: C:/Polarion/data/logs/main

### Warning

The default location is configurable. If you don't find log files at the location above, search your system for one of the log files named in the following section.

### Log files and usage

- log4j-TIMESTAMP.log generic log file, contains everything. Important for technical support purposes.
- log4j-errors-TIMESTAMP.log errors only
- log4j-jobs-TIMESTAMP.log messages related to scheduled jobs
- log4j-licensing-TIMESTAMP.log messages related to license(s) and license usage
- log4j-monitoring-TIMESTAMP.log messages related to system monitoring, audit and repair
- log4j-startup-TIMESTAMP.log messages related to startup and reindex process
- log4j-rpc-TIMESTAMP.csv Information about RPC calls (GWT)
- log4j-tx-TIMESTAMP.log statistics data

## Polarion (Log4j) log files

Apache log4j is used for logging. Its configuration file is polarion/plugins/org.apache.log4j\_\*/log4j.properties. It can be customized if copied to polarion/configuration/log4j.properties. A server restart is required for the configuration modifications to take effect.

## Additional log files

The following additional log files are configurable on both Windows and Linux platforms:

- logs/apache Apache logs
- logs/derby Derby logs

Windows location: C:/Polarion/data,

**Linux location:** (The location varies depending on the Linux distribution installed.)

## PostgreSQL log files

Log files for the PostgreSQL database are located on path c:/Polarion/data/logs/postgresql. These logs are rotated after 7 days.

## Installation log files

Installation log files store information about the installation process. If you should experience problems with installation you can provide these files to Polarion Technical Support.

Windows location: C:/Polarion/data/logs/install/\*.log,

**Linux location**: /var/log/polarion/\*.log - The location is configurable on both platforms.

## Log file rotation

Polarion log files are rotated on weekly basis.

Apache log files are rotated as follows:

- Windows log files are rotated on a weekly basis.
- For Linux, a distribution-specific log rotation utility can be used, most probably logrotate (see e.g. http://www.debian-administration.org/articles/117).

### Remove old log files

You can periodically remove old log files to optimize disk storage space on the server. For information, please see: Cleanup of Old Log Files.

# Xml export schema

The following is the XML schema for Polarion's XML Export feature. This is the schema of files that all the \*.xsl templates transform (and also the schema of exported files that are created by the XML export option xml: XML Export if the Copy-xml.xsl template is used). You will need refer to this schema for any custom XSL templates you may create to use with XML Export.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
targetNamespace="http://polarion.com/xml-export"
  xmlns="http://polarion.com/xml-export" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/1999/xhtml"</pre>
    schemaLocation="http://www.w3.org/2002/08/xhtml/xhtml1-transitional.xsd" />
  <xs:complexType name="Option">
```

```
<xs:sequence>
    <xs:element name="property" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:string" />
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" />
  <xs:attribute name="name" type="xs:string" use="optional" />
  <xs:attribute name="isDefault" type="xs:boolean" use="optional" />
  <xs:attribute name="sequenceNumber" type="xs:integer"</pre>
    use="optional" />
</xs:complexType>
<xs:complexType name="User">
  <xs:sequence>
    <xs:element name="description" type="Text"</pre>
     minOccurs="0" />
    <xs:element name="email" type="xs:string" minOccurs="0" />
    <xs:element name="id" type="xs:string" />
    <xs:element name="name" type="xs:string" minOccurs="0" />
    <xs:element name="votes" type="WorkItemRefList" />
    <xs:element name="watches" type="WorkItemRefList" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Approval">
 <xs:sequence>
    <xs:element name="status" type="Option" />
    <xs:element name="user" type="User" minOccurs="0" />
 </xs:sequence>
</xs:complexType>
<xs:complexType name="ApprovalList">
 <xs:sequence>
    <xs:element name="approval" type="Approval" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="UserList">
 <xs:sequence>
    <xs:element name="user" type="User" minOccurs="0"</pre>
     maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WorkItemRef">
  <xs:attribute name="workItemId" type="xs:string" />
  <xs:attribute name="projectId" type="xs:string" />
</xs:complexType>
```

```
<xs:complexType name="WorkItemRefList">
  <xs:sequence>
    <xs:element name="workItem" type="WorkItemRef"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Attachment">
  <xs:sequence>
    <xs:element name="author" type="User" minOccurs="0" />
    <xs:element name="fileName" type="xs:string" />
    <xs:element name="length" type="xs:positiveInteger" />
    <xs:element name="title" type="xs:string" minOccurs="0" />
    <xs:element name="updated" type="xs:dateTime" />
    <xs:element name="url" type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AttachmentList">
  <xs:sequence>
    <xs:element name="attachment" type="Attachment"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="OptionList">
  <xs:sequence>
    <xs:element name="option" type="Option" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StringList">
  <xs:sequence>
    <xs:element name="item" type="xs:string" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Category">
  <xs:sequence>
    <xs:element name="description" type="Text"</pre>
      minOccurs="0" />
    <xs:element name="id" type="xs:string" />
    <xs:element name="name" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CategoryList">
  <xs:sequence>
    <xs:element name="category" type="Category" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="Comment">
  <xs:sequence>
    <xs:element name="author" type="User" minOccurs="0" />
    <xs:element name="created" type="xs:dateTime" />
    <xs:element name="parentComment" type="xs:positiveInteger"</pre>
      minOccurs="0" />
    <xs:element name="text" type="Text" />
    <xs:element name="title" type="xs:string" minOccurs="0" />
    <xs:element name="visibleTo" type="StringList" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CommentList">
  <xs:sequence>
    <xs:element name="comment" type="Comment" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Text">
  <xs:choice>
    <xs:element name="html" type="xhtml:Flow"/>
    <xs:element name="plain" type="xs:string"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="Hyperlink">
  <xs:sequence>
    <xs:element name="role" type="Option" minOccurs="0"</pre>
      maxOccurs="unbounded" />
    <xs:element name="uri" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="HyperlinkList">
  <xs:sequence>
    <xs:element name="hyperlink" type="Hyperlink"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PlanningConstraint">
  <xs:sequence>
    <xs:element name="constraint" type="Option" />
    <xs:element name="date" type="xs:dateTime" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PlanningConstraintList">
  <xs:sequence>
    <xs:element name="planningConstraint" type="PlanningConstraint"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="Project">
  <xs:sequence>
    <xs:element name="active" type="xs:boolean" />
    <xs:element name="description" type="Text"</pre>
      minOccurs="0" />
    <xs:element name="finish" type="xs:date" minOccurs="0" />
    <xs:element name="id" type="xs:string" />
    <xs:element name="lead" type="User" minOccurs="0" />
    <xs:element name="location" type="xs:string" />
    <xs:element name="name" type="xs:string" />
    <xs:element name="start" type="xs:date" minOccurs="0" />
    <xs:element name="trackerPrefix" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="TimePoint">
  <xs:sequence>
    <xs:element name="closed" type="xs:boolean" minOccurs="0" />
    <xs:element name="description" type="Text"</pre>
     minOccurs="0" />
    <xs:element name="earliestPlannedStart" type="xs:date"</pre>
     minOccurs="0" />
    <xs:element name="id" type="xs:string" />
    <xs:element name="name" type="xs:string" minOccurs="0" />
    <xs:element name="time" type="xs:date" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WorkRecordList">
  <xs:sequence>
    <xs:element name="workRecord" type="WorkRecord"</pre>
     minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WorkRecord">
  <xs:sequence>
    <xs:element name="comment" type="xs:string" minOccurs="0" />
    <xs:element name="date" type="xs:date" />
    <xs:element name="timeSpent" type="xs:string" />
    <xs:element name="user" type="User" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LinkedRevisionList">
 <xs:sequence>
    <xs:element name="linkedRevision" type="LinkedRevision"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LinkedRevision">
  <xs:sequence>
    <xs:element name="message" type="xs:string" minOccurs="0" />
    <xs:element name="revision" type="xs:string" />
```

```
</xs:sequence>
</xs:complexType>
<xs:complexType name="RevisionList">
  <xs:sequence>
    <xs:element name="revision" type="Revision" minOccurs="0"</pre>
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Revision">
  <xs:sequence>
    <xs:element name="author" type="xs:string" />
    <xs:element name="created" type="xs:dateTime" />
    <xs:element name="internalCommit" type="xs:boolean" />
    <xs:element name="message" type="xs:string" />
    <xs:element name="name" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LinkedWorkItemList">
  <xs:sequence>
    <xs:element name="linkedWorkItem" type="LinkedWorkItem"</pre>
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LinkedWorkItem">
  <xs:sequence>
    <xs:element name="revision" type="xs:string" minOccurs="0" />
    <xs:element name="role" type="Option" />
    <xs:element name="suspect" type="xs:boolean" />
    <xs:element name="workItem" type="WorkItemRef" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WorkItem">
  <xs:sequence>
    <xs:element name="fields">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="approvals" type="ApprovalList"</pre>
            minOccurs="0" />
          <xs:element name="assignee" type="UserList"</pre>
            minOccurs="0" />
          <xs:element name="attachments" type="AttachmentList"</pre>
            minOccurs="0" />
          <xs:element name="author" type="User" minOccurs="0" />
          <xs:element name="categories" type="CategoryList"</pre>
            minOccurs="0" />
          <xs:element name="comments" type="CommentList"</pre>
            minOccurs="0" />
          <xs:element name="created" type="xs:dateTime"</pre>
            minOccurs="0" />
          <xs:element name="description" type="Text"</pre>
            minOccurs="0" />
```

```
<xs:element name="dueDate" type="xs:date"</pre>
        minOccurs="0" />
      <xs:element name="hyperlinks" type="HyperlinkList"</pre>
        minOccurs="0" />
      <xs:element name="id" type="xs:string" minOccurs="0" />
      <xs:element name="outlineNumber" type="xs:string" minOccurs="0" />
      <xs:element name="initialEstimate" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="linkedRevisions" type="LinkedRevisionList"</pre>
        minOccurs="0" />
      <xs:element name="linkedRevisionsDerived" type="RevisionList"</pre>
        minOccurs="0" />
      <xs:element name="linkedWorkItems" type="LinkedWorkItemList"</pre>
        minOccurs="0" />
      <xs:element name="location" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="module" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="plannedEnd" type="xs:dateTime"</pre>
        minOccurs="0" />
      <xs:element name="plannedStart" type="xs:dateTime"</pre>
        minOccurs="0" />
      <xs:element name="planningConstraints" type="PlanningConstraintList"</pre>
        minOccurs="0" />
      <xs:element name="previousStatus" type="Option"</pre>
        minOccurs="0" />
      <xs:element name="priority" type="Option"</pre>
        minOccurs="0" />
      <xs:element name="project" type="Project"</pre>
        minOccurs="0" />
      <xs:element name="remainingEstimate" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="resolution" type="Option"</pre>
        minOccurs="0" />
      <xs:element name="resolvedOn" type="xs:dateTime"</pre>
        minOccurs="0" />
      <xs:element name="severity" type="Option"</pre>
        minOccurs="0" />
      <xs:element name="status" type="Option"</pre>
        minOccurs="0" />
      <xs:element name="timePoint" type="TimePoint"</pre>
        minOccurs="0" />
      <xs:element name="timeSpent" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="title" type="xs:string"</pre>
        minOccurs="0" />
      <xs:element name="type" type="Option" minOccurs="0" />
      <xs:element name="updated" type="xs:dateTime"</pre>
        minOccurs="0" />
      <xs:element name="workRecords" type="WorkRecordList"</pre>
        minOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="customFields" minOccurs="0"</pre>
  maxOccurs="unbounded">
```

```
<xs:complexType>
          <xs:sequence>
            <xs:element name="field" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CustomFieldValue">
                    <xs:attribute name="id" type="xs:string" />
                    <xs:attribute name="name" type="xs:string" />
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 <xs:complexType name="CustomFieldValue">
    <xs:choice>
      <xs:element name="string" type="xs:string" />
      <xs:element name="boolean" type="xs:boolean" />
      <xs:element name="currency" type="xs:decimal" />
      <xs:element name="date-time" type="xs:dateTime" />
      <xs:element name="duration" type="xs:string" />
      <xs:element name="enum" type="Option" />
      <xs:element name="integer" type="xs:integer" />
      <xs:element name="multi-enum" type="OptionList" />
      <xs:element name="text" type="Text" />
      <xs:element name="time" type="xs:time" />
      <xs:element name="date" type="xs:date" />
      <xs:element name="float" type="xs:float" />
    </xs:choice>
  </xs:complexType>
  <xs:element name="workItems">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="workItem" minOccurs="0" maxOccurs="unbounded"</pre>
          type="WorkItem" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# Internally defined metrics

This topic lists the default set of internally defined Work Item metrics for projects contained in the workitem-metrics.xml file.

Any of these may be removed or commented out in the configuration file if you don't want the metric to be calculated. For information about configuring Work Item metrics, see: Work Item Metrics.

Name	Key	Туре	Description	
Resolved vs. Unresolv	d RESOLVED	-wistributi	Counts resolved (subkey RESOLVED) and unresolved (subkey UNRESOLVED) Work Items in the project.	
Unresolv By Assigned	UNRESOLV	e <b>rdistrib</b> uti	Counts unresolved Work Items for every assignee in the project. Subkeys are User IDs.	
Unresolv by Category			communes unresolved Work Items for every Category in the project, on and also Work Items with no assigned Category. Subkeys are Category IDs, and _uncategorized.	
Unresolv by Time Point	<b>(ed</b> nresolv	ED-BY-TIM <b>distributi</b>	Counts unresolved Work Items for every Time Point in the project, on and also Work Items with no Time Point set. Subkeys are Time Point IDs, and _unscheduled.	
Most Assigned to an Individua		integer	Counts the highest number of Work Items assigned to a single user. The rendered value includes the name of that user).	
Most Entered by an Individua	MEI	integer	Counts the highest number of Work Items created by a single user. The rendered value includes the name of that User.	
Most Solved by an Individua	MSI	integer	Counts the highest number of resolved Work Items assigned to a single user. The rendered value includes the name of that User.	
Estimate Accuracy	ESTIMATE	-ACCURACY	Calculates accuracy of Initial Estimate values of individual Work Items in this way:	
			<ul> <li>Finds all Work Items that have a value set for Initial Estimate, plus a value set for at least one of Time Spent and Remaining Estimate.</li> </ul>	
		percenta	ge Sums all the Initial Estimate values (TOTAL_INITIAL).	
			Sums absolute values of differences initialEstimate - (timeSpent + remainingEstimate) (TOTAL_DIFFERENCE).	
			Finally, accuracy is calculated as TOTAL_INITIAL / (TOTAL_INITIAL + TOTAL_DIFFERENCE). The higher the accuracy value, the better.	

Name	Key	Туре	Description	
Project Plan Accuracy		percenta	<ul> <li>Calculates overall accuracy of time estimates in the project in this way:</li> <li>Finds all Work Items that have a value set for Initial Estimate plus a value set for at least one of Time Spent and Remaining Estimate.</li> <li>Sums all the Initial Estimate values (TOTAL_INITIAL).</li> <li>Sums all values for Time Spent and Remaining Estimate (TOTAL_ACTUAL).</li> <li>Finally, accuracy is calculated as TOTAL_INITIAL / TOTAL_ACTUAL, if TOTAL_INITIAL &lt; TOTAL_ACTUAL, or as TOTAL_ACTUAL / TOTAL_INITIAL, if TOTAL_INITIAL &gt; TOTAL_ACTUAL. The higher accuracy value, the better.</li> </ul>	
Planning	1	-accuracy percenta	Calculates percentage of unresolved Work Items having Time Point set, that are assigned to a Time Point that is not yet past, counting all unresolved Work Items with Time Point set (PLANNED), and counting how many of them are assigned to a Time Point ending before the current day (DELAYED). Calculates the value as PLANNED - DELAYED / PLANNED.	
Estimate vs. Not Estimate Work Items	d PLANNING	-coverage distributi	Calculates number of estimated (subkey planned) and not estimated	
In Time vs. Underes Work Items			Rate of In-time (subkey correct) vs. underestimated (subkey incorrect) Work Items. (On time means that Initial Estimate >= ionTime Spent + Remaining Estimate). Only Work Items having a value set for Initial Estimate and a value set for at least one of Time Spent and Remaining Estimate are taken into account in the metric.	
Requirer with vs. without linked documer	n <b>ents</b> :eabi		Rate of requirements (Work Items of type Requirement) having at ioneast one attachment or hyperlink (subkey traceableRequests) and without any attachment and hyperlink (subkey untraceableRequests).	

# In-place defined metrics

See: In-place Defined Metrics

#### **Project group metrics**

Name	Key	Туре	Description
All Work Items	ALL	integer	Counts all Work Items in the project group.
Resolved vs. Unresolv	RESOLVED-UNR	s <b>distributi</b>	Counts resolved (subkey RESOLVED) and unresolved (subkey UNRESOLVED) work items in the project group.
Unresolv by Project	<b>/ed</b> nresolved-by		Counts unresolved Work Items in all projects in the project orgroup (and in child project groups, if any). Subkeys are project IDs.
Unresolv by Child Entity	<b>'ed</b> nresolved-by		Counts unresolved work Items in all projects and project orgroups directly in a project group (but not in child project groups). Subkeys are project IDs or project group names.

To display a specific element in a chart, it is necessary to prefix the element key with the pound (#) character. For example, following will display everything from the resolved-unresolved fact base:

```
{line-chart:report-path=charts/workitems-trend.xml|items=workitems-trend-data:
RESOLVED-UNRESOLVED/*|tags=14|width=100%|height=100%}
```

However, if you want to reference a specific element in this fact base, only the unresolved items, for example, then the key must have the # sign prefix:

```
{line-chart:report-path=charts/workitems-trend.xml|items=workitems-trend-data:
#RESOLVED-UNRESOLVED.UNRESOLVED|tags=14|width=100%|height=100%}
```

#### Calculations and facts

This section documents facts that can be displayed in dashboards and the calculations that can render them.

## **Fact bases**

This section contains reference information about Polarion fact bases.

## processaudit/project-level/source-check-aggregate

This fact base is calculated by **processaudit** calculation.

Values available in the project scope:

overall-test-coverage: unit test coverage

overall-test-results: unit test success ratio

#### repo-analysis:

This fact base is calculated by the **repoanalysis** calculation.

Values available in project and project group scope:

- **NOU**: Number of Users the number of users having a role in the project or in some project in the project group.
- NOF: Number of Files the number of files in the repository for the scope being reported.

Values available only in the project scope:

- **NOD**: Number of Developers where "developer" means the author of some commit to the repository.
- NODD: Number of Developers for the past day
- NODW: Number of Developers for past week
- NODM: Number of Developers for past month
- NOC: Number of Changes where "change" means a commit in the repository.
- NOCD: Number of Changes during the past day
- NOCW: Number of Changes during the past week
- NOCM: Number of Changes during the past month
- NOC-BY-DEVELOPERS: Number of Commits by Developers where "Developer" means the author of some commit to the repository.
- **MAD**: Most Active Developer meaning the Developer making the most commits to the repository.
- LAT: Last Activity Time meaning the time of the latest commit to the repository.

# workitems-data:

This fact base is calculated by the **trackeranalysis** and **trackeranalysis-projectgroup** calculations. The available values are those for Work Item Metrics

#### Trend fact bases

These parallel fact bases with trend data exist for each regular fact base listed in the previous section.

#### workitems-trend-data

Calculated by the trackeranalysis and trackeranalysis-projectgroup calculations.

Available values: Trends of all values from workitems-data fact base.

#### repo-trend

Calculated by the **repoanalysis** calculation.

Available values: Trends of particular values from the **repo-analysis** fact base.

Values available in the project scope:

NOC: Number of Changes ("change" means any commit in the repository)

- NOCD: Number of Changes for Last Day
- NOF: Number of Files (i.e. number of files in the repository)
- NOD: Number of Developers ("developer" means author of some commit in the repository)

#### processaudit/project-level/source-check-aggregate-trend

Calculated by the **processaudit** calculation.

Available values: Trends of all values (and their properties) from the **processaudit/project-level/source-check-aggregate** fact base.

Values available in both project and project group scope:

 test-coverage.ratio: test coverage (i.e. ratio of total lines of code to line of code covered by unit tests)

Values available in the project scope only:

- test-sucess-ratio.ratio: test success ratio
- test-sucess-ratio.testsCount: number of tests
- test-sucess-ratio.errorsCount: number of errors in tests
- test-sucess-ratio.failuresCount: number of failures in tests

# **Velocity macros location**

Polarion provides a few default macros implemented in Velocity that address, for example, the display of signatures in LiveDocs.

These built-in Velocity macros are defined in:

```
[Polarion HOME]\polarion\plugins\com.polarion.alm.wiki_N.NN.N\src\main\webapp\templates\macros.vm (Where wiki N.NN.N contains a version specification sich as 3.18.0).
```

Each of the sections is implemented as a specific sub-macro. For example, if a developer wishes to customize a macro such as **documentPanel**, they can refer to the following definition:

```
#macro(documentPanel $showPending $approvedState)
<div>
    #renderProperties()
    \\
    #renderLastApproved($approvedState)

    #renderSignatures($showPending)
</div>
#end
```

# **Velocity variables for Active Wiki Pages**

The following table lists the Apache Velocity variables available during Wiki page rendering. Developers may use these for extending Polarion Wiki pages via the respective Polarion API classes and interfaces. Descriptions of classes and interfaces are available in the Javadoc provided with the API.

Variable	API Class	Notes
\$calendarToo	com.polarion.wiki.util.Calendar	Tool
\$document	com.polarion.alm.tracker.model.	Only available on Wiki Content Blocks. (Not available for Wiki Pages.)
\$me		Available in Velocity context only. References the ID of the current user when used in a wiki page.
\$page	com.polarion.alm.wiki.model.IWi	kiPage
\$plan	com.polarion.alm.tracker.model.	ı∌A⊌ailable on old wiki pages that were used as Plan pages.
\$platformSer\	(ICOm.polarion.platform.IPlatform	Service
\$projectServi	Cecom.polarion.alm.projects.IProj	ectService
\$securityServ	ice om.polarion.platform.security.	   ISecurityService
\$testManager	<b>nentService</b> on.alm.tracker.ITestM	anagementService
\$testRun	com.polarion.alm.tracker.model.	ਾA∀ailable on old wiki pages that were used as TestRun pages.
\$trackerServi	Cecom.polarion.alm.tracker.ITrack	erService
\$transactionS	ervice polarion.platform.ITransact	ionService
\$wikiService	com.polarion.alm.wiki.IWikiServ	ice

# **Apache Lucene options**

Polarion uses an implementation of the Apache Lucene query engine to enable portal searches and Work Item queries. This section documents some properties which administrators can set in the polarion properties system configuration file to control some behaviors of the embedded Lucene.

Property	Description	Default Value
luceneMaxClauseCo	Defines how many OR/AND parts a query can have. This includes simple range queries which are expanded by Lucene winternally to this form. Polarion itself needs such queries with many ORs when there are many links to a single Work Item (e.g. thousands).	20000
luceneMaxFieldLen	Defines the maximum number of terms that will be indexed for a single field in a Document. This means that object (e.g. Work Item) will not be searchable on words appearing too far from the beginning of the value of the field (e.g. description). Special example is artificial field in which words are searched during full text search. Value should be set to higher value in case there are errors logged: "Lucene: maxFieldLength 50000 reached, ignoring following tokens". It can be set to negative value to mean no limit. However, Lucene documentation says that OutOfMemory error can occur. (This has not been tested with Polarion).  When the value of this property is increased because of the occurrence of errors in the log, the system index must be refreshed in order to fix the index entries of truncated fields.	50000

# License usage reference

The License page of Repository Administration shows current license usage information in the following tables:

# **Concurrent License:**

Column	Description	
Total	Maximum number of users than can access Polarion at the same time with concurrent license assigned.	
Configured	Number of users configured in the users file to share the licenses from the concurrent license pool.	

Column	Description	
Free	Number of licenses from the pool that are not assigned to any user at the moment.	
Peak Greatest number of concurrent licenses that were used at the same tir		

#### Named License:

Column	Description	
Total Number of named licenses that can be assigned to specific users.		
Configured Number of licenses that are currently assigned to a user.		
Free	Number of licenses from that are not assigned to any user.	

# **Usage Details:**

Column	Description		
User	The user ID of a user who is currently using a license.		
Status	<ul> <li>Status of the license used by Status</li> <li>Active: User is logged in to Polarion and using a license</li> <li>Waiting for expiration: User was using a concurrent license and has logged out, but the license is not yet returned to the license pool</li> </ul>		
License	Name and type of the license in use by <i>User</i> . (Type includes add-on licenses.)		

# Default assignment of concurrent licenses

License assignment of concurrent licenses occurs in the following way:

- Default license specified in the users file.
- If not specified, then the license with highest number of concurrent slots.
- If some licenses have the same highest number, then the lowest license that shares this highest number is assigned.
- · If all are zero, then no license is assigned.

# **Self-signed SSL certificates**

To use Polarion with SSL, you may opt to create and use a self-signed SSL certificate rather than using one from an authority trusted by Java. Using a trusted certificate is preferred and recommended. An untrusted certificate requires the additional configuration steps to import it to

the Java keystore. See: Configuring Polarion to use SSL. This topic describes how to create a self-signed certificate on Linux and Windows.

#### Linux

The following example creates a certificate named serverCert.pem You can name the certificate file as desired.

```
apt-get install openssl
openssl req -x509 -nodes -days 3650 -newkey rsa:1024 -keyout
serverKey.key -out serverCert.pem
```

#### **Note**

Depending on your operating system version, the above command may require some additional parameters.

#### **Windows**

The following example creates a certificate named servercert.pem. You can name the certificate file as desired.

Run the following command from the command prompt, as a single line:

```
openssl req -x509 -nodes -days 3650 -ewkey rsa:1024 -keyout privateKey.key -out
serverCert.pem
```

You may need add -config [Polarion]\bundled\apache\conf\openssl.cnf in order to add the appropriate openssl.cnf - which is in Apache's conf directory (hidden .cnf suffix). Example:

```
openssl req -config D:\Polarion\bundled\apache\conf\openssl.cnf -x509 -nodes -days 3650
-newkey rsa:1024 -keyout privateKey.key -out serverCert.pem
```

# **User Reference**

## **User Reference overview**

The User Reference provides information you may need to refer to when using various Polarion features - Documents, dialogs, Work Items, etc.

# The user reference has the following main sections:

- Dialogs, User Interface, Features which includes.
  - o Activity Stream Describes sources and types that can be referenced when searching the Activity Stream.
  - Dialogs Describes fields and other information you may need to reference in several commonly-used dialogs.
- Diagram Editor Information about using the Diagram Editor tool to create diagrams in Documents and Work Items.

- Documents Information you may need to refer to when using various Document features.
- Non-Latin Alphabets Information you may need to know when using Polarion with non-Latin characters and alphabets.
- Templates Reference information about different kinds of templates.
- Testing Management Reference info for test management and testing features.
- Work Items Various types of reference information related to Work Items.

# **Activity Stream**

This section documents the sources and types of Activities in the Activity Stream. You can filter the Activity Stream for activity source, or activity source and type by specifying a query with the following structure:

- Activity source only: sourceId: [ID]
- Activity source and type: sourceId:[ID] AND type:[TYPE-ID]
- Activity source, multiple types: sourceId:[ID] AND (type:[TYPE-ID]) OR type:[TYPE-ID])

Source ID	Type ID(s)	Activity Source and Type
administration	changed	<pre>sourceId:administration, sourceId:administration AND type:changed</pre>
builds	started, finished, finished_with_failures, failed, canceled	sourceId:builds, sourceId:builds AND type:finished, sourceId:builds AND type:failed
commits	changed	sourceId:commits, sourceId:commits AND type:changed
jobs	started, finished, failed, canceled, aborted	<pre>sourceId:jobs, sourceId:jobs AND type:started, sourceId:jobs AND type:failed</pre>
testruns	created, updated, status-changed, deleted	sourceId:testruns, sourceId:testruns AND type: created, sourceId:testruns AND type:updated AND type:status-changed
workitems	created, updated, deleted	sourceId:workitems, sourceId:workitems AND type:created, sourceId:workitems AND type:updated

# **Attachment File Types**

## File types

This topic lists the supported attachment types searchable by their content and/or metadata. Generally every attachment is indexed by its file name, file extension, author ID, author name,

update date and file size. The ones listed here are searchable by their content also unless otherwise noted. Attachments are only indexed (and therefor searchable) if the system property to enable indexing of attachments is enabled in the system configuration. (See Advanced system tuning: attachments.indexingOfAttachmentContent.enabled.)

- Archive/Package formats
- Audio formats
- eBook formats
- FLV Video (Metadata is indexed)
- HTML/XHTML files
- Image formats
- Java class files (.class)
- Mailbox files (.mbox)
- Microsoft formats
- OpenDocument formats
- **PDF**
- Rich Text
- TXT and derived formats
- XML and derived formats

# **Archive/Package Formats**

The contents of archive files is also parsed.

- ar
- bzip2
- cpio
- gzip
- jar
- pack200
- tar
- UNIX dump
- XZ

zip

#### **Audio formats**

Only metadata is indexed.

- Audio Interchange File (aiff, aif, aifc)
- Basic Audio (au, snd)
- MIDI (mid, midi)
- Wave audio (wav)

## eBook formats

- IBook (ibook)
- Epub (epub)

# **Image formats**

Metadata is indexed.

- Bitmap (bmp, dib)
- Portable Network Graphics (png)
- Graphics Interchange Format (gif)
- JPEG (jpg, jpeg)
- Windows Icon (ico)
- Wireless Application Protocol Bitmap (wbmp)

#### **Microsoft formats**

- Microsoft Office:
  - o Excel (xls, xlsx, xlsm, xltx, xltm)
  - o Outlook (msg)
  - o Powerpoint (ppt, pptx, pptm, ppsm, ppsx, ppam, potx)
  - o Project (mpp)
  - o Publisher (pub)
  - o Visio (vsd)
  - o Word (doc, docx, docm, dotx, dotm)
- OLE Objects (only if Polarion is hosted on Windows Server platform)

# **OpenDocument file formats**

#### Documents:

- o Open-Document Chart (odc)
- Open-Document Formula (odf)
- o Open-Document Graphic (odg)
- o Open-Document Image (odi)
- o Open-Document Master (odm)
- o Open-Document Presentation (odp)
- Open-Document Spreadsheet (ods)
- o Open-Document Text (odt)

# Templates:

- Open-Document Chart Template (otc)
- Open-Document Formula Template (otf)
- o Open-Document Graphic Template (otg)
- Open-Document Web page Template(oth)
- o Open-Document Image Template (oti)
- o Open-Document Presentation Template (otp)
- Open-Document Spreadsheet Template (ots)
- Open-Document Text Template (ott)

# **Captions**

# **Captions overview**

You can add a caption to the following types of content in Polarion:

- Images (in Documents, Work Items, table cells)
- Tables (in Documents, Work Items)
- Diagrams (in Documents, Work Items, table cells)

To add a caption, click on the item and click **Add Caption** on the pop-up menu. In the **Add Caption** dialog, you can:

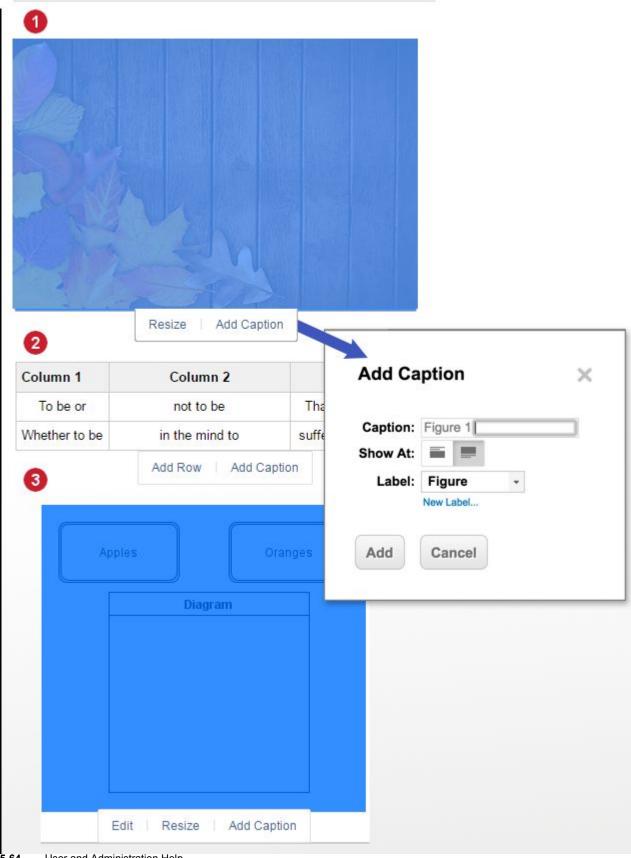
Enter your caption text in the Caption field.

- Select whether the caption should appear above or below the item by clicking the appropriate icon in **Show At**.
- Select a label that describes the item type, or add a new label. The value set in this field is prefixed to the value in the **Caption** field.

# Captions and "On-demand Work Item Loading"

- With **On-demand Work Item Loading turned ON**, if the document "Work Item Presentation configuration" is set for **Test Steps** or *Rich Text* **Custom Fields** to be visible, *all caption numbers appear as "#"*. They will be replaced with the correct numbers when exported.
- If the document Work Item Presentation configuration does not contain Test Steps or Custom Fields as outlined above, then caption numbers will display, even with On-demand Work Item Loading turned ON.
  - (Visible caption numbers will display correctly even if previous captions from the same sequence have not finished loading.)
- With **On-demand Work Item Loading turned OFF**, all caption numbers will appear in the document but may take a while to load in very large documents.

# **Captions**



Captions are available on these elements in Documents and Work Item rich-text fields



There are two default caption label values: **Table** and **Figure**. These values determine if a caption is included in a Table of Figures or a Table of Tables, if these are inserted into a Document.

# **User-defined Caption Labels**

You can optionally add a new label and apply it when captioning images, diagrams or tables. For example, you might create a label "Citation" and specify it when adding a caption to some items. You could then insert a "Table of Citations" in your Document. Note that tables of figures, tables, and custom-labeled items appear only in documents generated by export to PDF and Round-trip for Microsoft Word. A placeholder appears in your Polarion Document.

A custom caption label appears in the **Label** list when the Document contains a caption that uses it. To remove a custom label, remove all captions that use it and refresh your browser.

# **Dialog boxes**

## **Create New Project dialog box**

The Create New Project dialog box presents several pages where you specify properties, template, and other details about a new project. These pages also appear when the dialog box is invoked in Administration to mark an existing repository folder as a project.

## Basics page

- **ID:** Specify a unique identifier for your new project. Characters are restricted to: a-z, A-Z, 0-8, underscore, dash, and period. No spaces.
- **Location:** Specify the Repository folder to be the root of the new project. If this will be child of an existing folder, specify that folder. For example, to specify a new project folder under the existing Sandbox folder, you would enter **Sandbox/MyNewProject/**.
- **Icon: SVN Picker:** Click to launch the SVN Picker dialog box to pick an existing repository folder as parent of your new project's folder. Folder name selected will be appended in the Location field.
- **URL:** Read-only field showing the full repository path to your new project's root folder. Field changes as you type in the Location field.
- Tracker Prefix: Specify a short string to be used as a prefix to the IDs of new Work Items created in the project.

## **Note**

This prefix should be *unique* to your Polarion system. If it is duplicated, then citations of Work Item IDs in commit comments may cause the revision to be linked to the wrong item in the wrong project. For example, if two projects use the default prefix WI, there could be two Work Items in the system with WI-3 as the ID. Then if a developer references WI-3 in a commit message in an external SVN client, the revision will be linked to the first Work Item found by the system having that ID, which may or may not be the right one.

# Template page

This page enables you to choose a project template on which to base the new project. Project templates may provide any or all of various system configurations, workflow, repository folder structure, wiki content, Documents, Work Items, shortcuts, etc. all customized to support a particular type of project.

Only templates enabled by your system configuration and supported by your license appear in the Create New Project dialog box. For a listing of the default project templates, see: Default Project Templates.

#### Summary page

This page displays a read-only summary the details you have specified for the new project up to this point. You have the opportunity to make changes before the project is created in the system. Click **Previous** to go back and make any changes. When you are satisfied with the details, click **Next** to begin creating the new project.

#### Creation page

This page displays a progress indicator while the project is being created, and an alert which the operation is finished. If any error occurred, it is reported here. When the page reports Project created, click **Next**.

#### Details page

- Name: The name of the project that appears to all users in the Open Project or Project Group dialog box.
- ID: The project's ID. Read-only.
- Lead: Name of the project leader. Default value for a new project is the current user. You can select from a list of current users. If the user is not listed, you can safely skip this and specify it later in the project administration interface after the project leader's user account has been created in the system.
- Active: Flags that this project is actively under development. Default for a new project is checked state. If the new project will be active later on, you can clear this option and check it later in the project's administration interface.
- **Start:** The date on which work will begin (or began, if you have marked an existing code base as a Polarion project). Default value for a new project is the current date. Click the calendar

icon to pick a date. If not known, it is safe to skip and specify a date later on in the project administration interface.

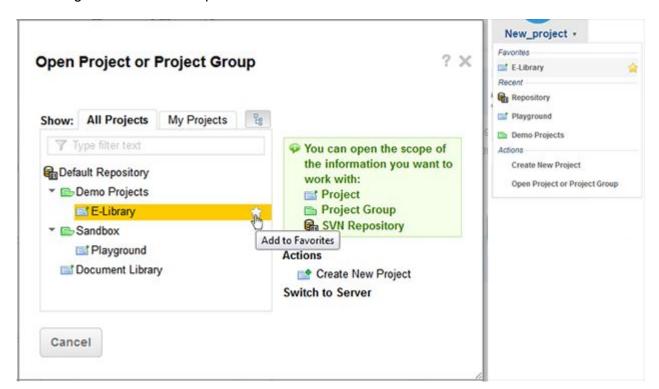
- **Finish:** The date on which the project is scheduled to finish. Click the calendar icon to pick a date. If not known, it is safe to skip and specify a date later on in the project administration interface.
- URL: The URL of the project's root in the repository. Read-only.
- **Description:** Enter free-form text describing the project. This can be helpful to future new team members, managers, and other stakeholders.

## **Open Project or Project Group dialog box**

Invoked from: **Open Project or Project Group** item on the project menu (click the name of the current project in Navigation).

This dialog box enables you to access an information scope in which you want to do some work. This dialog box enables you to:

- Access the current Repository to view dashboards, wiki and other information, or perform configurations in this scope.
- Access any project group (in the current repository) to see dashboards, wiki and other information, or perform configurations in this scope.
- Access a project to process Work Items, see dashboards and other information, or perform configurations in this scope.



The left side of the dialog box is a tabbed content browser comprised of the following items:

- **All Projects** tab: Displays all the content in the current repository allowing selection of any project group or project to open.
  - When a project is selected the **Add to Favorites** icon appears in the highlighted group or project name. Click it to add the selected project to the Favorites tab.
- My Projects tab: Displays only those project groups and projects for which you have some role. Select any available content to open it.
  - When a project is selected the Add to Favorites icon appears in the highlighted group or project name. Click it to add the selected project to the Favorites tab.
- **Switch Flat/Tree** button: Appears to the right of the tab set and switches between hierarchical and flat presentation in the currently selected tab.

The right side of the dialog contains the following:

- Information box: green box explains the icons used in the dialog
- Actions: displays a list of available actions you can invoke. For example, if you have permissions
  to create a new project, the Create New Project link appears. If there are no actions available
  given your user permissions, the section does not appear.
- **Switch Polarion Server**: Appears only if your Polarion system has been configured to work with multiple repositories hosted on different servers. Displays a list of links for the currently configured repositories. Click the name of the repository you want to access.

Related information is available in the Access Projects topic.

# Linked Work Items dialog box

Invoked from: Navigation → ■ Documents and Pages → , selected Document, sidebar, Work Item Properties, Edit Links.



Manage the links of Work Items defined in online Documents

#### Columns:

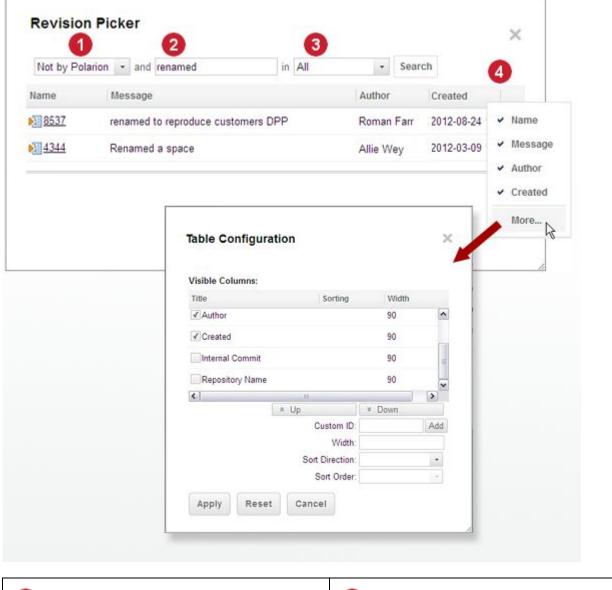
Suspect: Toggles the Suspect property of the link. For information, see: Suspect Links.

- **Role**: The link role defining the relationship between the linked Work Items. (For information, see: Link Roles and Rules.) List contains the link roles currently configured for the project and which are applicable to the Work Item type as defined in link role rules in the project configuration.
- Title: Displays the ID and Title of the linked Work Item. The string is a link that leads to the linked Work Item in the Table view of the Work Items topic. When adding a new link the edit field accepts the ID or outline number of the target Work Item. Use ID to link to Work Items that are not defined in the current Document. Outline number works only with Work Items defined in the current Document. The icon in this column leads to a picker dialog in which you can browse or use querying to locate the target Work Item for the link.
- **Project**: Read-only. Displays the name of the project to which a linked Work Item belongs.
- Revision: The repository revision of the target Work Item to which the source Work Item links.
   Normally, links will point to the Head revision of the item, but you have the option to link them to a specific revision number, either the current revision, or a specific revision which you explicitly select in the picker dialog box.
- Assignee: Read-only. Displays the name of the person(s) to which the linked Work Item is assigned. If no name appears, the linked item is currently unassigned.
- Actions: Provides icons enabling you to remove existing links or add new links to the Work Item currently selected in a Document. The Remove icon is disabled when a link is structural in nature, derived from the element structure of the Document. For example, if a Work Item is a Heading 3 which is under a Heading 2 which is also marked as a Work Item, the two sections have a parent-child relationship which is tracked with a link using the appropriate link role. As this is inherent structure, the system maintains these structural links and does not allow users to explicitly remove them via link editing. If the two sections were merged into a single section, the link would be automatically removed.

# **Revision Picker dialog box**

Invoked from the Table view of Work Items (Navigation  $\Rightarrow$  Work Items), Linked Revisions section,  $\[ \]$  (Select Revision) icon .

Use this dialog box to find and select a repository revision to link to the current Work Item, and enter the revision number in the current line of the item's **Linked Revisions** section. Please refer to the following figure:





The dialog box presents a query bar and a table of revisions found by the current query. Click on the revision number in the table to select that revision as the link target and close the dialog box.

# **Query segments**

The query bar has 3 controls that enable you to formulate 2 query segments:

Control	Туре	Values	Note
Query scope	Pick List	All - All revisions, regardless of how created	Revisions may be created externally
		Not by Polarion - Revisions not created by users using Polarion	using a Subversion client.
Search target	Text box	Enter any text, or a field ID:value	You can use AND, OR and NOT operators to query for multiple things. For example, query for revisions containing some text in the commit message, and committed on a specified date.
User filter	Pick List	<ul> <li>All - Revisions by all users</li> <li>Created by me - Limit results to revisions created by you.</li> <li>List of all user names - Select a user name to limit results to revisions created by that user</li> </ul>	You can also query on the <b>author</b> field of the revision

# **Revision fields**

Polarion indexes several fields for repository revisions, enabling you to query for values in those fields. The following table lists these fields.

Field ID	Description	Notes
Name	This is actually the revision identifier, which is a number	Example: name:5874 or NOT name:87512
message	Text of the revision commit message	If all you want to find is some word or phrase in the commit message, you can skip entering this field name and just enter the text you want to find. You can combine search on this field with search on some other field.  Example: Read only AND created: 20090628
	J	If you want to eliminate some word or words, then you need to reference the field. Example: message:Read AND NOT message:only

Field ID	Description	Notes
author	User ID (not name) of the Polarion user who created a revision	In many cases you would not need query on this field. Rather, you could select the user in the user filter segment.
		In cases where you want to eliminate some user, or you want results from several users, then you would need to query on this field, selecting <b>All</b> in the user filter segment.
		<pre>Examples: NOT author:jonesL, author:jonesL: OR author:smithM</pre>
created	The date when a revision was created	Specify date as: yyyymmdd. For example, for July 22, 2009 enter created:20090722
internalComm	Flags whether or not a revision was committed internally it using Polarion, or externally using a Subversion client.	This is a Boolean field, values true, false. Example: internalCommit:false
repositoryName		Only viable if Polarion is configured to use external repositories. <i>Example</i> : repositoryName:myGitRepo

# **Table configuration**

You can configure which revision fields appear in the Revision Picker dialog box, in what order, and their sorting order and direction. To invoke the **Table Configuration** dialog box, right-click on the header bar of the revisions table (in the Revision Picker dialog box), and on the menu choose **More**.

# Import from Excel

This topic provides reference information that may be useful when importing Work Items defined in Microsoft Excel.

# **Supported Work Item fields**

Columns and data in Excel can be mapped to the following Work Item fields during import to Polarion. Please see following notes on mapping multivalued data to multivalued fields.

- Assignee \*
- Author
- Categories \*
- Comments
- Custom fields \*

- Description
- Due Date
- Hyperlinks \*
- Initial Estimate
- Linked Work Items \*
- Priority
- Remaining Estimate
- Resolution
- Resolved On
- Severity
- Status
- Time Spent
- Title

# Note

\* character in the above list means the field supports import of multivalued data. Note that import of multivalued custom fields is supported for enumeration types only, not for primitive types, which can still be imported as single values.

#### Multivalued data and fields

It is possible to define multivalued data in Excel so that it can be mapped to multivalued fields in Polarion Work Items, and multiple values written to such fields during import.

The following fields must be written in a special format using a specific pattern:

- Linked Work Items: pattern: "[role:][project/]id". External linked items pattern is: "[role:]URL".
- Hyperlinks: pattern: "[role:]URL"

# **Keyboard Shortcuts**

This topic documents the default keyboard shortcuts available in Polarion.

Shortcut	Function
ALT + Q	Focus Work Item query

Shortcut	Function	
ALT + E	Edit button on all forms	
CTRL + S	Save button in data entry forms	
ALT + UP, ALT + DOWN	Select previous or next Work Item in Multi Edit view of the Work Items topic.	

## Shortcuts that focus fields on the form in the Work Item editor

To focus the following fields, press CTRL+ ALT +

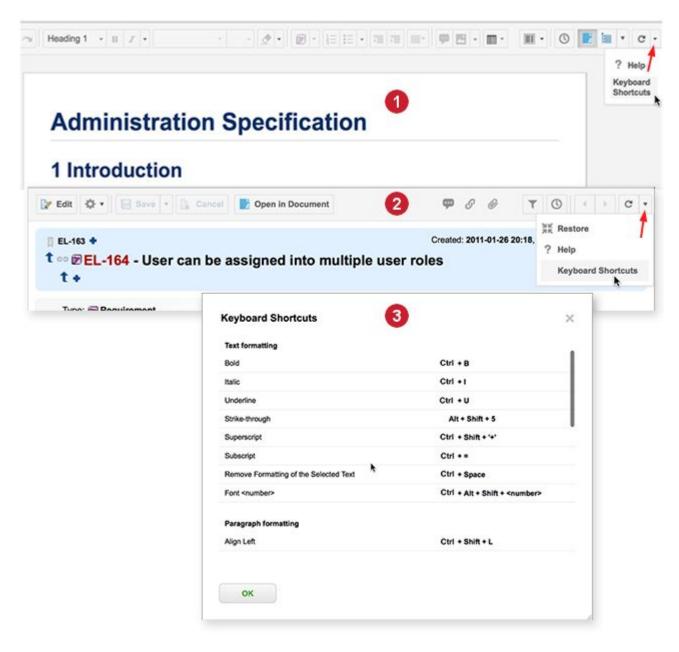
- A Assignee
- C Comments
- D Description
- E Initial estimate while creating a new item, Remaining estimate otherwise.
- · G Categories
- K,L Linked Work Items
- N Time Spent
- P Time Point
- · R Resolution
- T Title (edit mode only)
- V Severity
- **W Status** (workflow)
- Y Priority

# **Note**

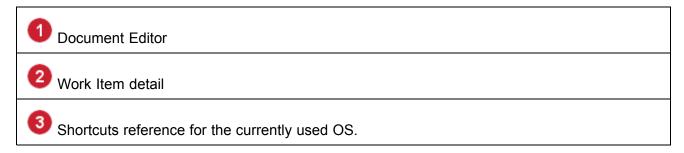
It is possible to disable all keyboard shortcuts (also in Multi Edit) by setting the property disableKeyboardShortcuts=true in the system configuration file polarion.properties (follow link for location).

#### **Document and Rich-text editor shortcuts**

A comprehensive reference dialog, context-sensitive for your operating system, is available via the **Keyboard Shortcuts** item on the Help menu of the respective editors.



How to access Keyboard Shortcuts reference dialog in Documents and Rich-text fields.



#### Shortcuts for tables

The following keyboard shortcuts are available in tables in the Work Item Editor (including a Test Steps table), and the Document Editor:

Shortcut	Function	
Tab	Move cursor to the next cell to the right, or if located in the last cell of a row, to the first cell of the next row. Content of the cell is selected.	
Tab	If cursor is in the last cell of the last row, insert a new row locating the cursor in the first cell on the left.	
Shift + TabTab	Move cursor to the previous cell to the right, of if located in the first cell of a row, to the last cell of the previous row. Content of the cell is selected.	

# Time values

Values for time are always rendered in server's time zone, regardless of the time zone where the user is located. For example, if the server is located in the Eastern time zone of the United States, and a user who is located in the Central Europe time zone updates a Work Item, the time stamp of the update will be according to the server's time zone, not the user's time zone.

# **Diagram editor**

#### **Diagram editor overview**

# ALM REQ

Polarion provides the Diagram Editor tool for creating and editing diagrams. You can use it to create and edit diagrams in LiveDoc Documents, including Work Items contained in Documents, and in the Description field of Work Items in the Table view of the Work Items Tracker. After inserting a new diagram, you can insert a caption above of below it.

The Diagram Editor provides an extensive library of shapes and symbols that can be using to create a wide variety of diagrams including UML diagrams, flowcharts, and circuit diagrams.

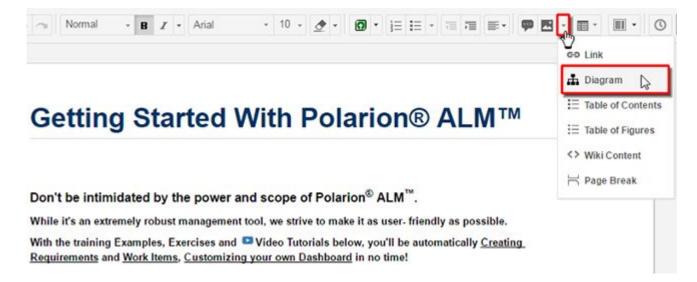
- Create a new Diagram,
- Edit a Diagram,
- Shape and symbol palettes,
- Copy a Diagram,
- Delete a Diagram,
- Print a Diagram,
- Miscellaneous notes.

# Create a new diagram

You can create new diagrams in LiveDoc Documents or the Description field of Work Items in the Work Item Tracker. In Documents, you can create diagrams in the Document's plain text, or in Document-based Work Items.

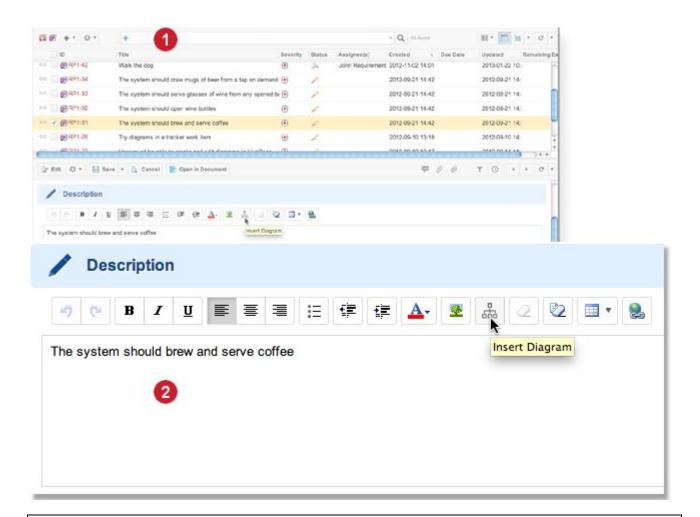
# Create a new diagram in LiveDoc Documents

- 1. Open the project containing the Document, and in **Navigation** browse to the space and Document, clicking on the Document to open it in the Document Editor.
  - Alternatively, you can use the Search box in **Navigation** to search for the desired Document.
- 2. In the Document, create a new line at the place you want to insert a new diagram. If you want to place the diagram inside a Work Item, place the caret inside the Work Item and use the Insert menu on the Work Item icon in the Document's left margin to add a new line above or below the current caret position.
- 3. On the Diagram Editor toolbar, click on the Insert drop-down menu and choose **Diagram**.



#### Insert a digram in Tracker-based Work Items

- 1. In the project containing the Work Item, select **Work Items** in **Navigation** to open the Work Item Tracker. If the Table view is not the current view when the page opens, select that view.
- 2. Browse or query Work Items, and in the upper pane of the Table view select the item in which you want to insert a diagram.
- 3. Scroll the lower pane of the Table view until you can see the **Description** field of the selected Work Item.
  - Click on the header to open the field for editing.
- 4. On the Description field toolbar, click the **Insert Diagram** button.







# Tip

You can also create and edit diagrams in custom fields of the Rich Text type.

# **Edit a Diagram**

When you edit a Document or the Description field of a Work Item containing a diagram, the diagram is not automatically editable. An additional action is required to open the diagram in the Diagram Editor. However, you can resize the diagram without opening it in the Diagram Editor.

# To edit an existing diagram in a Work Item's Description field:

- 1. Click the field header to open the field for editing.
- In field, click anywhere on the diagram.

3. Click the **Edit** button that appears on the diagram to launch the Diagram Editor.

# To edit an existing diagram in a Document:

- 1. Click anywhere on the diagram.
- 2. Click the **Edit** button that appears on the diagram to launch the Diagram Editor.

#### Resize a Diagram

To resize an existing diagram, follow the same steps as for editing, except click the **Resize** button.

Enter the new pixel dimensions in the **Resize Image** dialog. Note that the aspect ratio is automatically maintained unless you toggle off the **Keep Aspect Ratio** button.

#### **Shape and Symbol palettes**

The Diagram Editor provides an extensive library of shapes, symbols, and clip art images that can be placed into diagrams. These are grouped into categories, and each category appears as an expandable section in the symbol library panel on the left side of the Diagram Editor. Expand or contract any section by clicking on the name.

You can control which categories appear in the symbol library panel. Click **More Shapes** to display a menu listing all the categories. Categories showing check marks are visible in the panel, while all others are hidden. Check or uncheck any category to show or hide it in the symbol library panel.

When you finish editing a diagram, the visible symbol categories are saved and will appear in the symbol library panel next time the diagram is edited.

#### Insert Shapes, Symbols and Images

Depending on the type of shape or symbol, you may need to use different actions to insert it into the diagram. Try clicking on a shape, symbol, or clip art image. A click inserts many elements, especially block-like elements. If clicking does not insert the element, use drag-and-drop from the library panel to the diagram background. Connector type elements in particular may require this action to insert them.

# **Use Custom Images**

You can include an image attached to a Document or a contained Work Item into a diagram using the **Insert Image** action in Diagram Editor toolbar. Caveats:

- When attaching an image to an item in the Diagram Editor, use the default attachment link, not the direct link for image URLs.
- The image URL will typically be full URL, so there may be problems if the ID of the containing (Work Item or Document) is changed, or if it is moved to another project.

# Copy a Diagram

If you want to copy an existing diagram to another Document, to another place in the same Document for subsequent modification, or to a Work Item, you should copy the diagram source code to your clipboard and paste it into a new diagram. Do not copy the diagram image, as it will not be editable in the new location.

# Copy the Diagram's source code

- 1. In the Diagram Editor menu, choose **File Diagram** Editor menu, choose **File Edit Source** to launch the source code editor dialog.
- 2. Select the entire source code in the dialog, copy it to your clipboard, and click the **Cancel** button.

#### Create a copy of the Diagram

- 1. Open the Document or Work Item in which you want to create a copy of the first diagram, and create a new diagram.
- 2. In the new diagram, in the Diagram Editor menu, File Edit Source to launch the source code editor dialog.
- 3. In the source code editor, paste the previously copied source code from your clipboard making sure you replace all the code that appears there when you open the source code editor dialog.
- 4. Make sure the **Replace existing drawing** option is selected in the dialog and click **OK** button.

# **Delete a Diagram**

If a diagram is no longer needed, it is easily deleted. The action is the same regardless of whether the diagram is part of a Document or a tracker-based Work Item.

- 1. If viewing the diagram in the **Description** field of a Work Item in the Tracker, click the field header to place the field in Edit mode. (Ignore this step if working in a Document.)
- 2. Place the caret after the diagram image and press the Backspace key (Windows) or Delete key (Mac). The diagram image is removed.
- 3. **Save** the Document or Work Item to complete the delete operation.

## **Print a Diagram**

The File menu of the Diagram Editor includes the Page Setup command and the Print command. Use these to set up printing of the current diagram and print it to any accessible local or network printer. Only the currently open diagram is printed, not the Document that contains it.

#### Miscellaneous notes

Several notes that may be of interest:

When a user begins editing a diagram, a lock us set and no other user can edit it until the diagram changes are saved, the user closes the Diagram Editor, or the lock timeout occurs. The duration of the lock before it times out is set by the system administrator.

See: Advanced System Tuning (com.polarion.ui.lock.timeout).

- If the Duplicate Work Item function is invoked while a diagram is being edited, pending changes to the diagram are saved before the Work Item is duplicated. Subsequent changes to the diagram after the Work Item has been duplicated will not be reflected in the new duplicate Work Item.
- Changes to diagrams in Work Items are tracked in the Work Item history. That is, if the only change to the Work Item is a diagram change, the revision of the Work Item in which the change

was made contain the diagram in the state it was when the Work Item was saved. (Changes to a diagram are not saved to the history until the Work Item is saved.)

# **Documents**

# **Create LiveDoc Document Dialog**

Access point: Index page of any **Space**. Click the • icon on the page toolbar, select **New LiveDoc Document**.

Field	Description
Title	Human readable title of the Document. Supports non-ASCII characters. Contained Work Items can be searched using the Document Title visual query element, entering any part of the value. A value is required.
	Should not be confused with the Document heading style <b>Title</b> .
Name (ID)	The unique system identifier of the Document within the project and space. Only ASCII characters may be used. Contained Work Items can be searched using the <b>Document ID</b> visual query element, entering the exact value only. A value is required.
Туре	The Document Type appears only if Document Types are enabled in the system configuration. Field list contains all the Document Types currently configured in the current scope. If present, it is a required field.
Space (or Subspace)	Name of the Space or Subspace in which the new Document will be stored. Field list contains the names of all spaces and subspaces in the current scope. Value defaults to the space name from which the dialog was invoked.
Work Item Type	Specifies which type of Work Item the new Document will contain (Addition types can be added after the Document is created.) Field list contains the names of all Work Item types currently configured in the current scope. Value defaults to the type specified as the default type in the Work Item Types configuration in Administration.
Link Role	Defines the relationship that hierarchically structured Work Items in the Document will have to each other. Field list contains the link roles currently configured in the current scope.
Enable Outline Numbering	If checked, Work Items in the new Document will automatically be outline numbered based on the Work Item level and position in the Document.
Prefix	Optional. If specified, the value is prepended to each outline number.

# **Document Sidebar Selector**

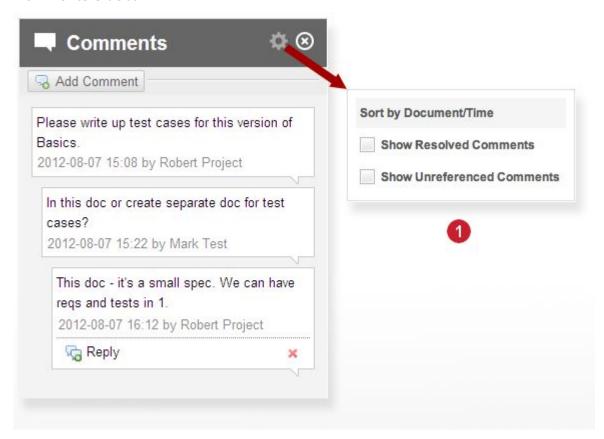
The document sidebar selector ( ) is available when editing a Document. It can display any of several sidebars showing comments, Document outline, and other information about the Document.

There is also access to tools such as Test Run planning. Select the desired sidebar via the Sidebar drop-down menu on the Document Editor toolbar.

Commands
P Comments
Work Item Properties
Recycle Bin
Signatures
Document Properties
E Document Outline
✓ Approvals
▲ Test Run Planning

Planning

#### Comments sidebar



View, compose, and manage comments in a Document



This panel shows the comments that have been inserted into the Document, if any. It also enables adding of new comments (including threaded replies to existing comments), and removal of existing comments.

#### Note

Document comments are not the same as the **Comments** field of a Work Item

#### **Component descriptions**

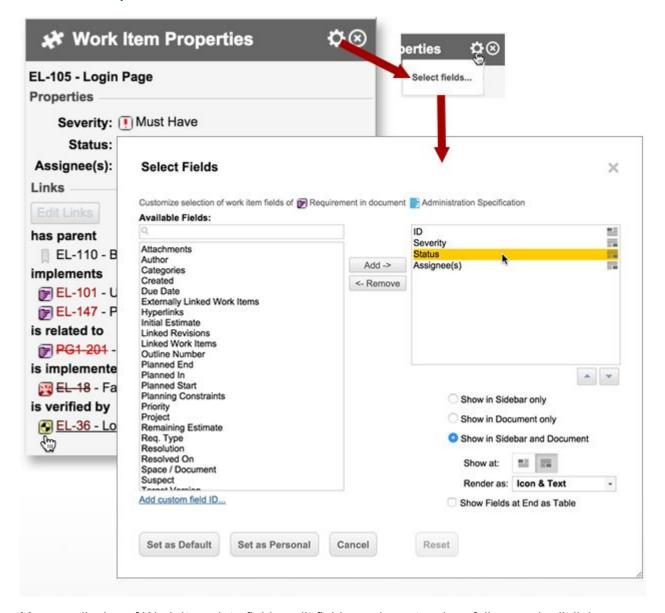
- Add Comment Adds a new comment thread to the Document.
- Reply Appears inside a comment balloon when the balloon is selected. When clicked, adds a new threaded comment one level down.
- Melete comment Appears inside a comment balloon when the balloon is selected.

Removes the selected comment from the Document.

The comment is not actually deleted in the repository until the Document is saved.

- Sidebar Settings When clicked, displays a drop down menu of settings applicable to the Comments sidebar. Settings:
  - o **Sort by Document/Time** Controls the sort order of the comments appearing in the Sidebar.
  - o **Show Resolved Comments** Toggles visibility of comments and comment threads that have been marked as resolved.
    - (Resolved comments and threads are hidden to keep them out of the way of subsequent comments and discussions.)
  - o **Show/hide Unreferenced Comments** Toggles appearance of unreferenced comments in the Comments page.
    - Unreferenced comments are comments that have been removed by deleting their icon in the Document text, but which still exist in the Document repository.
- Close In the sidebar's header. Closes the Comments panel.

## **Work Item Properties sidebar**



Manage display of Work Item data fields, edit fields, and create, view, follow, and edit links.

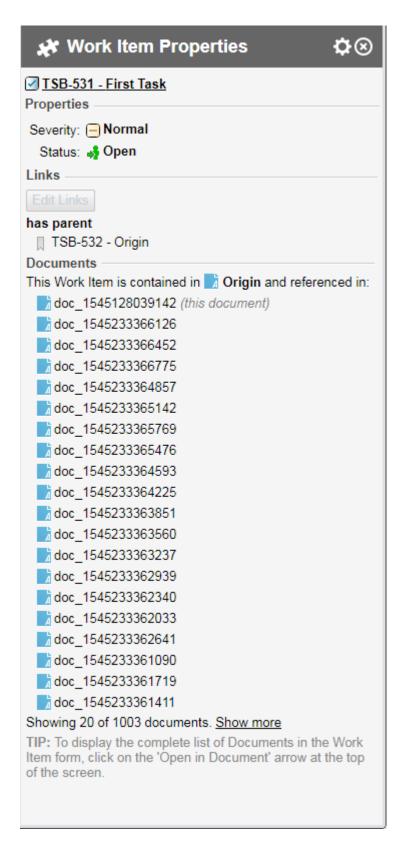
This panel shows Work Item data fields that can be edited from the Document. It enables a user to specify which data fields appear in the panel, and in what order.

The panel also shows outgoing links from the Work Item currently selected in the Document. Users can access the linked items, create new links via pasting, and to manage links in a pop-up dialog.

# **Component Descriptions:**

- Label: Shows the ID and Title of the Work Item currently selected in the Document body.
- Properties section Displays edit controls for the Work Item data fields currently enabled for editing from the Document.

- **Links** section Displays the **type** icon, **ID**, and **Title** of all Work Items that the currently selected Work Item has outbound links to. Each item is a hyperlink. If the target item is in the same Document, the link leads to the properties of the linked Work Item, which display in this pane and replace the previous information. If the link is to an item external to the Document, the item opens in s separate browser tab or window.
  - Multiple links are split into categories according to link role (e.g. "has parent", "implements", etc.).
- Edit Links Launches the Linked Work Items dialog that enables users to manage the linking of the Work Item currently selected in the Document. See Edit Work Item links.
- **Documents** section Displays the name of the Document that contains the selected Work Item, and the names of other Documents that the selected Work Item is referenced in (if any). When a reference is clicked it opens the document in a new browser tab.



The number of Document references displayed is set to 20 by default but can be changed via the following property in the polarion.properties file:

com.siemens.polarion.ui.referencingDocuments.softLimit=20

(The most recently updated documents appear at the top of the list.)

#### Note

Due to possible Document restrictions, the displayed result may not accurately represent the total number of times that the Work Item is referenced.

Sidebar settings - Displays a menu with options applicable to this sidebar. See next section.

# Sidebar settings

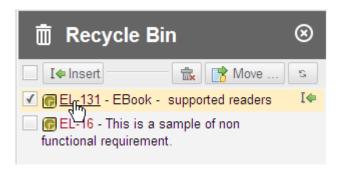
The Select Fields menu option leads to the Select Fields dialog. The dialog displays a list of all standard fields for the Work Item currently selected in the Document. Custom fields may be added to the list of Available Fields using the link **Add Custom Field ID**. Fields in the list on the right appear in the Fields section of the panel. User can remove fields by selecting fields in the right-hand list and clicking **Remove** button, or add fields by selecting them in the Available Fields list and clicking **Add** button. Fields can be moved up or down in the list by first selecting, then clicking either the **Up** or **Down** button in the dialog. The **Reset** button returns the Work Item selection to the default settings.

The **Set as Default** button applies the changes and makes them the default for the Document. Other users will see the selected fields in the Work Item Properties sidebar. The Set as Personal button applies the changes and makes them the default for the current users. Other users see the Document default fields in the sidebar.

When a field is selected in the right-hand list, several options appear in the dialog for how to display the field.

- **Show in Sidebar Only** The field only appears in the Document sidebar.
- **Show in Document Only** The field only appears in the Document.
- Show in Document and Sidebar The field appears in both Document and Document sidebar.
- **Show At** Appears only if option to show field in Document is selected. Controls whether to show the field at the start or end of Work Items.
- Render As Appears only for Polarion fields that have an associated icon, and only if option to show field in Document is selected.
  - Controls whether the Work Item in the Document displays as icon, text, or both.
- Show Fields at End as Table If checked, any fields having option set to show them at the end of Work Items will be formatted as a table.

## Recycle Bin sidebar



View Work Items in repository that are no longer used in the Document. Access options for handling them.

This panel displays Work Items removed from the Document but still stored in the repository, and provides tools for handling them. It can be accessed from the Document Editor toolbar, and via the **Open in Document's Recycle Bin** button on the toolbar of Work Items currently in the Recycle Bin, when such items are viewed in the Table view of the Work Items Tracker. Options are:

- Insert Work Item(s) into the Document
- Delete Work Item(s) from the repository.
- Move Work Item(s) to another Document in the project

## **Component Descriptions:**

- Check box: unlabeled Selects all Work Items in the Recycle Bin.
- Insert Inserts selected Work Item(s) into the current Document at the current insertion point, and re-associates the item(s) with the Document.
- Permanently deletes selected Work Item(s) from the Recycle Bin and the underlying repository.
- **Move** Invokes the **Move Work Item(s)** dialog which provides the following options:
  - o **Move selected Work Item(s) out of Document**: Disassociates the selected orphaned item(s) from the Document, but retains them as project artifacts in the repository where they are managed in the integrated Tracker and accessed via the Work Items topic.
  - o Move selected Work Item(s) to Document: Associates the selected orphaned item(s) with another Document, to which they are appended. User must identify the target Document by selecting the name of the space in which the target Document resides, and then selecting the name of the target Document.
- Refreshes the list of Work Items in the Recycle Bin. This can be useful if one user is reading the Document while another user is editing it, or if one author has removed or merged Work Items in the document since the Recycle Bin was displayed in the Sidebar.

# **Content handling**

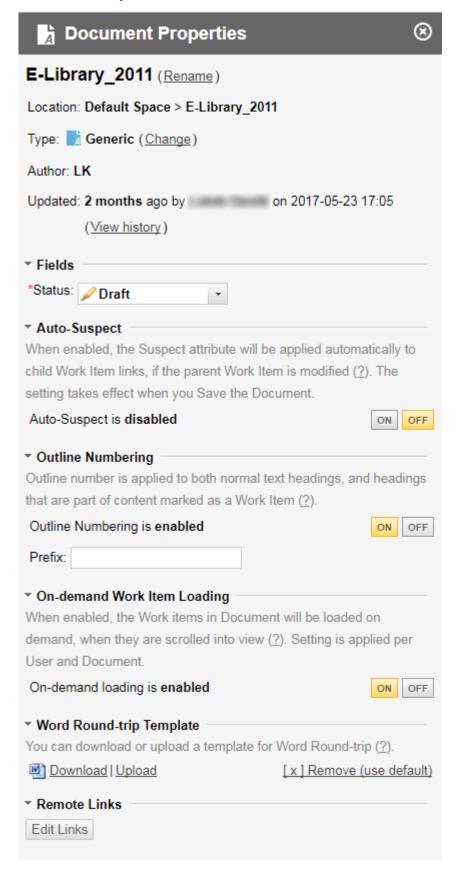
The following tables provide a reference to how various actions affect moving of Work Items to the Recycle Bin.

Action	Content to Bin?
Mark text as Work Item and press Delete/Backspace.	YES
Mark Work Item as part of selection and delete it.	YES
Mark text as Work Item and CUT the entire Work Item.	YES
Mark text as Work Item, CUT the entire Work Item, then PASTE it back.	NO
Merge Work Item.	YES

Action	Content to Bin?
Cut	YES
Cut, followed by Paste.	NO
Unmark	YES
Delete	NO

Action	Content to Bin?
Unmark Work Item	YES

# **Document Properties sidebar**



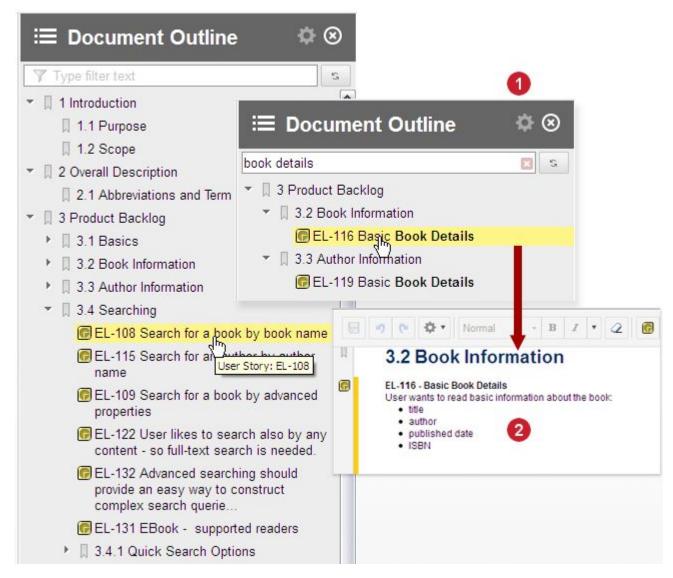
View properties and settings for the current Document

# This panel displays several properties and settings related to the current Document:

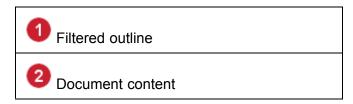
- Links enabling you to rename the current Document, or change its type.
  - Document types must be enabled in the project configuration or the *Change* link is disabled.
- Document custom fields and Document workflow status, if these have been configured for the project. (See: Configure Document Custom Fields and Configure Document Types and Workflow).
- Controls to turn Auto-suspect (for Work Items) on and off.
- Controls to turn outline numbering on and off, and to specify a prefix for outline numbers.
- Button to toggle on-demand loading of Work Items in the Document. On-demand loading can significantly improve usability of Documents containing a large number of Work Items. When enabled, only a few Work Items are initially loaded, enabling the Document to be viewed right away. Other items are loaded as the user scrolls. There are currently some limitations.
- Links to the Microsoft Office Word template used for Word Round-trip export.
- **Edit Links**: Adds remote linked data or edits existing links to the LiveDoc.

If the Document is a branched Document, the panel shows the name of the master Document and provides a link to it.

#### **Document Outline sidebar**



View the outline structure of the current Document and jump to content.



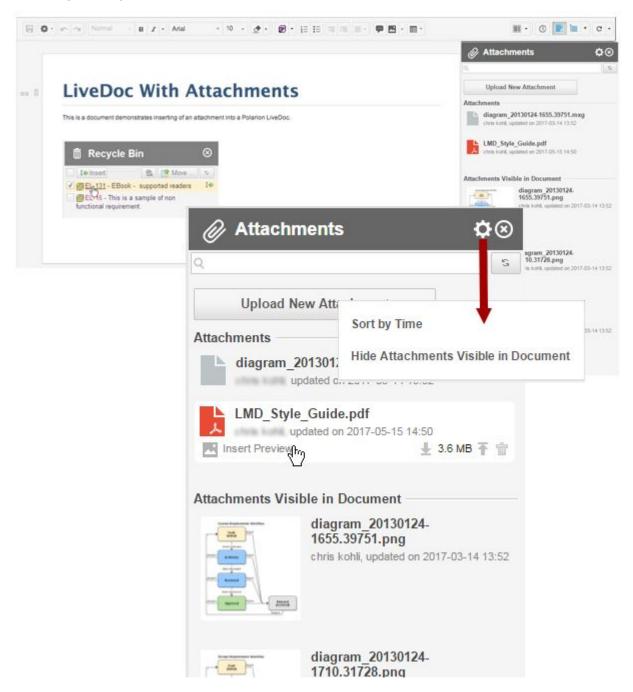
This panel displays the outline structure of the current Document.

- The tree contains nodes for headings and Work Items contained in the Document.
- Clicking any node in the tree structure scrolls the Document to show the selected content. Selecting content in the Document selects the equivalent node in the outline.

- When content is added to or removed from the Document, the Update button in the page may be used to refresh the outline.
- The panel provides a filter for the outline, which can help locate content quickly in a large Document. Simply type filter text in the edit control to automatically and incrementally search the outline structure. Clear the edit control to remove filtering and restore the full outline.

#### Attachments sidebar

# ALM REQ QA



This sidebar enables you to add new attachments to a Document, insert an image or attachment preview and to view and manage the list of existing attachments.

# **Component descriptions:**

- Sidebar header The icon displays a menu with the following items:
  - o **Sort Alphabetically** The default setting. Sorts the list of attachments alphabetically by file name.
  - Sort by Time Sorts attachments from newest to oldest.
  - Hide Attachments Visible in Document Hides the panel section Attachments Visible in Document and any attachments listed under it. This is useful if the there are many image or attachment previews visible in the Document body and you only want to focus on other attachments in the Attachments sidebar.
- **Search box** Filters the list of attachments according to the sting entered: "abc" matches "abc", "abcd" or "xabc", etc.
- Refreshes the list of attachments. (For example, another user might add an attachment while you are viewing the Document.)
- Upload New Attachment Lets users select a local file and upload it as an attachment to the Document.

(The document must then be saved to complete the attachment process.)

- Insert Preview: Inserts a Preview of the image or attachment into the document where the cursor is placed.
- **Download**: Downloads the object to the browser's download folder.
- **Update**: Upload an updated version of the previewed object. (Is disabled for read-only documents or for users with read-only permissions.)
- Delete: Deletes the selected attachment.
- Label Attachments labels the section of the sidebar that lists all attachments that are NOT displayed in the Document's body. It may contain images, Visio diagrams and other types of files. Files listed in this section can be inserted into the Document body by placing the cursor at the desired spot within the document and clicking Insert Preview button. Once inserted there, they are removed from this listing in the Attachments panel and added to the Attachments Visible in Document section.
- Label Attachments Visible in Document labels the section of the panel that lists all the attachments currently displayed in the Document body. This section may be hidden using the Hide Attachments Visible in Document item in the (sidebar settings) menu. When hidden, it can be restored using the **Show All** item in the same menu.

#### **Note**

# Pasting Images:

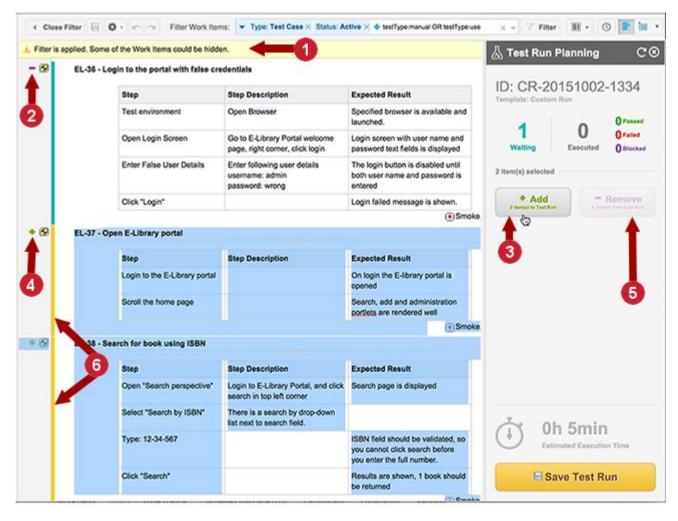
CTRL +V or Command +V can be used for images within the document, but if they are used for an image in a Work Item within the document, its thumbnail will NOT appear in the Attachments sidebar.

# Tip

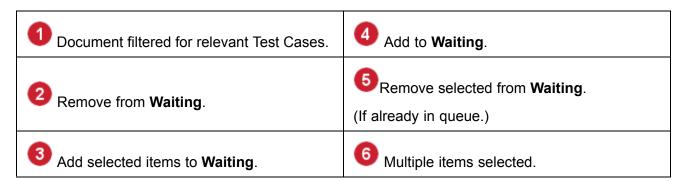
- No changes to a Document's attachments are permitted if the Document is derived from another Document. The original Document's attachments must be changed or updated.
- Clicking on an attachment representing an image displayed in the Document body scrolls the Document to its first appearance.
- Deleting an attachment representing an image displayed in the Document removes the image from all places where it was displayed in the Document.

Test Run Planning sidebar

ALM QA



Populate a Test Run with Test Cases from the current Document.



This panel enables you to select Test Cases to be executed when a Test Run is executed by testers. (For information see: Selecting (Planning) Test Cases.)

#### **Component descriptions:**

 Label - ID {name} shows the ID of the Test Run for which Test Cases are being selected or removed.

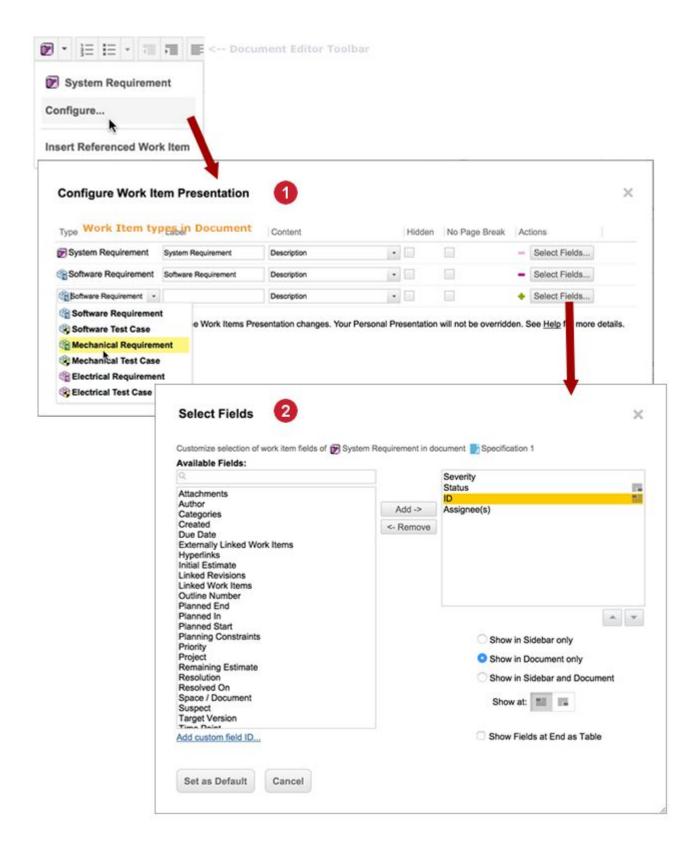
- Label Template {name} shows the name of the template on which the Test Run shown in ID is based.
- Panels Waiting shows a count of the Test Cases currently populating the Test Run. Executed shows a count of the Test Cases current populating the Test Run which have not currently been executed.
- + Add Adds one or more Test Cases, selected in the Document body, to the queue of waiting
  Test Cases for the Test Run. Enabled when Test Case(s) not already in the Waiting queue is/are
  selected in the Document.
- Remove Removes one or more selected Test Cases, selected in the Document body, from
  the queue of waiting Test Cases for the Test Run. Enabled when Test Case(s) already in the
  Waiting queue is/are selected in the Document.
- Sidebar Estimated Execution Time displays the total estimated time needed to execute all Test
  Cases currently populating the Test Run. Each Test Case has a field for storing the estimated
  time needed to execute it.
- Save Test Run saves the current Test Case selection to the Test Run.

# **Configure Work Item Presentation dialog**

Invoked from the Document Editor toolbar of an open Live Document (accessed from Navigation: Documents and Pages).

Content elements in Documents can be marked as Work Items - a paragraph can be marked as a Requirement, for example. Marking of elements may be done automatically via Work Item Rules when a Document is created via import from Word, or explicitly by a user working on the Document. The Configure Work Item Presentation dialog is where you can specify which type(s) of Work Items the Document can contain, and customize the presentation of elements marked as Work Items so that they display certain Work Item data field values in addition to the textual content. For example, you might display the Work Item ID and severity in paragraphs marked as Requirements, or the ID and status in paragraphs marked as Test Cases.

You can display the value of both standard Work Item fields and Custom Fields via the **Select Fields** button. You can specify whether field values appear at the start of the Work Item, or at the end. For example, you might display the value of the **outlineNumber** field at the start, and the **severity** field at the end of paragraphs that are marked Work Items. You can display multiple field values at both the beginning and end of marked elements. For example, you might show **outlineNumber** and **id** at the start, and **severity** and **timePoint** at the end of paragraphs marked as Work Items.



Adds support for multiple Work Item types in a Document and specifies fields to appear along with text.

## Legend



Add a Work Item type.



Configure fields to show in Work Item Properties.

Each row in the table represents a Work Item type. Document content can be marked as a Work Item of any type specified in this dialog.

#### Fields:

• **Type**: Specifies a Work Item type to be supported by the Document. The list contains the types currently configured for the project in the workitem type enum.xml enumeration.

(See: Configuring Enumerations.)

- **Label**: The name of the Work Item type specified in *Type*. This value appears in the drop-down menu of the Document editor toolbar. Value defaults to the configured name of the Work Item type selected in *Type*. User can change it, but there is usually no need to do so.
- Content: Specifies what Work Item content will be included when a new Work Item is inserted, or Document text is manually marked as a Work Item of the type specified in *Type*. Take note of the following:
  - o If you select **Description**, all selected content is placed in the Work Item's Description and no value is set for its Title. When viewed in the integrated tracker, the Work Item's Title field will be derived from the first 80 characters of the Description. No separate title appears in the Document.
  - o If you select **Title and Description**, all selected content is placed in the Description, and a line for Title is created which the user fills in. Title is maintained as a separate data field from Description.
  - o If the setting in **Content** includes **Title** and/or **Description**, the **ENTER** keyboard shortcut has different behavior in these fields.
- No Page Break: When checked, then Polarion will attempt, so far as possible, not to have page breaks within Work Items when exporting the Document to PDF, or round-trip for Microsoft Word.
- **Hidden**: If checked, the Work Item type is not displayed in the drop-down menu of the Document Editor toolbar. The presentation will be applied to existing Work Items of the type specified in *Type* (if any), but users will not be able to apply it manually to new or existing Document content.
- Actions: The → icon adds another row to the table in which support for an additional Work Item type can be configured for the Document. The → icon removes a row from the table, removing support in the Document for the specified Work Item type. The remove icon is disabled, and the row for the Work Item type cannot be removed if any Document content has been marked as a Work Item of the type defined by the row. The content must either be deleted from the Document, or unmarked as a Work Item in order to remove a row from the table, and support for the Work Item type it represents.

The **Select Fields** button leads to the Select Fields dialog in which you can specify which Work Item fields should appear in the Document, and where they should appear. You can opt to show any field (including custom fields) in the Document, the Document Sidebar (Work Item Properties page) or both. Some fields (usually of Enumeration type) allow you to specify whether they appear as text, as an icon, or both. (Note that this dialog can also be opened via **Select Fields** menu item of Work Item Properties page of the Document Sidebar.)

Fields configured as showing in the Document will be exported to PDF, and to Microsoft Word for Word round-trip.

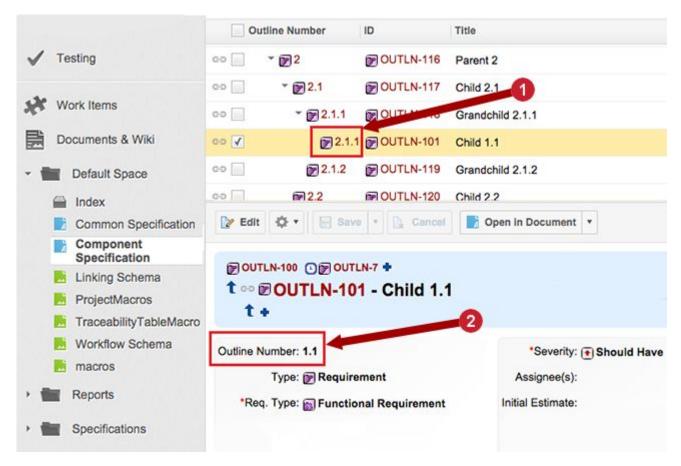
# Tip

The **Select Fields** dialog can also be invoked via the (Sidebar Settings) icon in the Work Item Properties sidebar (enabled when a Work Item is selected in a Document). When invoked from there, the fields configuration can be saved as a user-specific personal configuration only for the current user, or as the default for the Document.

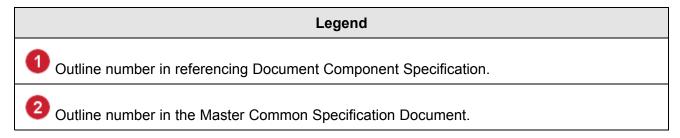
## Outline numbering display

When outline numbering is enabled in Documents, outline numbers appear for Document headings and Work Item titles in the Document Editor. (See: Use Outline Numbering.) This topic describes the display of outline numbers in the Table and Tree Table views of Documents, including Work Items referenced from a different Document.

- If an outline number is assigned to a Document-based Work Item, then the Work Item Table and Tree Table views show the outline number from the *master* Document (the Document that actually contains the original item).
- When a Document is selected in Navigation and viewed in the Table or Tree views, and a Work Item is selected the table or tree, the outline number you see in the Outline Number column of the table/tree is the outline number from the Document selected in Navigation. The value in the Outline Number field in the selected Work Item's detail form may be different from the value in the table column if the item is referenced from another Document. In that case, the value in the field is the outline number of the item in the Master Document the Document that actually contains the Work Item and from which it is referenced to other Documents.
- If a Work Item has no outline number assigned, then the Outline Number column in the table section of the page shows only icon representing the Work Item type.



Outline numbering display for referenced Work Item in Tree Table view of Document.



# PDF watermarks order of precedence

The order of precedence used by the PDF exporter for watermark images in PDF output is as follows:

- 1. Exact match (confidential\_portrait\_A4.png).
- 2. Default for orientation (confidential portrait.png).
- 3. Bigger size for the same orientation (confidential\_portrait\_A3.png).
- 4. Smaller size for the same orientation (confidential\_portrait\_letter.png).
- 5. **Default** (confidential.png).

- 6. The same size for the other orientation.
- 7. Default for the other orientation (confidential landscape.png).
- 8. Bigger size for the other orientation (confidential landscape A3.png).
- 9. Smaller size for the other orientation (confidential landscape letter.png).

Project repository is checked first, followed by global repository. If no suitable image is found in any scope, no watermark is included in the PDF output, and an error is written to the log files. For additional information, see Multiple Watermark Images.

#### Permissions and license

Your ability to access and use Document features depends on the permissions assigned to your user account, which may be tied to the user role(s) assigned to you by the system or project administrator. The license you are currently using may also affect your ability to use some Document-related features. This section provides some specifics that may help if you find you cannot access or use some feature or perform some operation.

## Word Import and Round-trip

# **Word Import:**

- The Import action is only available if you have permission to create Documents and Work Items and you are using either a Polarion ALM or Polarion Requirements License.
- Even if you have user permissions as above, you can only save import rules if you have write permission for the relevant repository location, i.e. .polarion/tracker/import configurations/word/.

#### Word Round-trip:

- You can invoke Word Round-trip > Export when you have read-only permission for a Document provided you are using a Polarion Requirements or Polarion ALM license.
- To import changes in a modified exported Word document, you need to be using a Polarion Requirements or Polarion ALM license, and you need to have write permission for the Subversion repository, and also Polarion permission to modify the Document.

# **Document features and operations**

Polarion provides project managers the ability to exert control over who can do what with a Document. This is accomplished by configuring Document permissions in global and/or project administration. This section describes the Document permissions that can be assigned and what they allow users who have the respective permissions.

#### **Note**

The MODIFY permission for Documents implemented in Polarion versions prior to version 2014 SR2 is replaced by the MANAGE, MODIFY FIELDS, MODIFY CONTENT permissions. If MODIFY was granted, the 3 new permissions will be granted after updating. Likewise, if MODIFY was denied, the new permissions will also be denied. The net outcome is that users should not experience any difference in what they are able to do with Documents.

See also: Dynamic Roles for Documents.

#### **Document access and content:**

#### **READ**

Allows user to read Document content and view Document information is Document sidebar panels.

#### **CREATE NEW**

Allows user to create new Documents, provided that user is also granted MANAGE, MODIFY FIELDS, and MODIFY CONTENT permissions.

#### **MODIFY FIELDS**

Allows user to modify Document fields (Status, for example) and custom fields shown in the Document Properties sidebar panel.

Granting of the MANAGE and/or MODIFY CONTENT permission for Documents does not grant the permission to MODIFY FIELDS.

#### **MODIFY CONTENT**

- Add/remove/move Work Items, including Headings and referenced Work Items, to/from/within the Document structure using the Document Editor.
- Move Work Items, including Headings and referenced Work Items, to/from the Document Structure via the Move Work Item dialog.
- Fields that appear in the Content column of the Configure Work Item Presentation dialog (i.e. Title, Description, and Test Steps).
- Add, modify, or remove Document attachments.
- Add, modify, or remove attachments to Work Items of all types contained in the Document (including Headings), provided user also has permission to modify Work Items in the project.
- Add, edit, or delete plain text in the Document. That is, text which is not contained in Work Items.

#### Additional notes on this permission:

 Users who are denied this permission, but who are granted the COMMENT and/or RESOLVE COMMENT permission, can insert comments into the Document and/or resolve comments in the Document (in text and Work Items).

#### Warning

If MODIFY CONTENT is granted, do not also explicitly grant COMMENT or RESOLVE COMMENT permissions. The granted user will automatically be granted them.

- This permission does not apply to referenced Work Items in the Document.
- Granting this of permission does not grant permission to MODIFY FIELDS.
- Granting of the MANAGE permission for Documents does not grant permission to MODIFY CONTENT.

# "Manage" allows user to:

- Configure default Work Item type(s) and Work Item presentation for the Document.
- Change the Document type (if the Document types feature is enabled).
- Rename or move the Document.
- Upload a round-trip template.
- Configure Document outline numbering.
- Enable/disable Auto-suspect for Document Work Items.
- Delete comments from a Document.

User must also have the MODIFY and MODIFY-FIELDS permissions for Documents.

#### **DELETE**

Allows the user to delete Documents.

## **Document Comments:**

#### COMMENT

#### Allows:

- Create new Document comments,
- Reply to Document comments.

#### RESOLVE COMMENTS

Allows user to resolve Document comments.

#### **Permissions for Document actions**

The following table lists the permissions required for different Document-related actions, and the capabilities/features available when a user has the listed permissions.

#### Action: View Document objects in user interface

Required Permissions:

- READ Document,
- READ Work Item.

#### Note

- Documents appear in Navigation Documents and Pages topic.
- Documents are listed in space Index pages and overviews.
- Documents can be opened (e.g. from links).

## **Action: Read a Document**

Required Permissions:

- **READ Document**
- **READ Work Item**

#### Note

- Documents are read-only.
- Round-trip and PDF exports will work.

#### **Action: Create a new Document**

Required Permissions:

- **READ Document**
- **MODIFY Document**
- **CREATE Document**
- **READ Work Item**
- MODIFY Work Item
- **CREATE Work Item**

#### **Note**

- Option to create new Document appears in space Index pages and space Overview.
- Import for Microsoft Word is available.

# **Action: Modify a Document**

Required Permissions:

- READ Document
- MODIFY Document
- CREATE Work Item
- READ Work Item
- MODIFY Work Item

#### Note

- Modify Document body text.
- Modify Work Item description and fields.
- Mark/paste OR unmark/cut/or recycle bin Work Items.
- Mark/paste OR unmark/cut/delete Headings.

If you want to control the deletion of headings via the DELETE permission, then set the headingsObeyDeletePermission property to true.

(Otherwise it's enough to have the "MODIFY CONTENT" permission.)

· Round-trip import is available.

## **Action: Delete a Document**

Required Permissions: Same as for Modify a Document, plus DELETE Document Notes:

#### Note

Option to delete Documents appears on space index pages.

## Action: Re-import changes in exported round-trip (Word) document

Required Permissions: Same as those for Modify a Document

Notes:

#### **Note**

Changes are imported if there are no errors or invalid data.

# Macros and Wiki mark-up support in Word Round-trip

This section the level of support in the export operation of Word Round-trip for text formatted with Wiki mark-up and inserted into a Document, and for Wiki macros which may be inserted into a Documents. Items with full and partial support are listed. You can assume that any mark-up and macro not

listed here is not supported when a Document is exported to Microsoft Office Word using the Word Round-trip feature. It may still be possible to export Documents containing unsupported mark-up and/or macros, but formatting and content in the exported document may be sub-optimal or missing.

#### The following macros and other elements are supported:

- Headings: converted to Word as native Word headings; style of Wiki/Document headings is not preserved
- Generic formatting, including: bold, underlined, italic, strike-through, monospace, subscript and superscript
- URL links (e.g. [https://www.polarion.com], [Polarion >https://www.polarion.com]
- Lists (unordered, ordered, hybrid multi-level)
- Escaped text, explicit line break (i.e. '\'), horizontal rule (i.e. '—')
- {code} macro: text is converted into Word in monospace font (Courier New)
- {quote} macro
- {workitems} macro (Note: some styles may not be converted, but the macro is generally supported)
- Document macros:
  - 1. {document}: link is converted to Word (with icon)
  - 2. {document-workitems}: Note that tables are not converted completely to Word.
  - 3. {document-property}
- Project macros:
  - o {project-property:active}
  - o {project-property:description}
  - o {project-property:finish}
  - o {project-property:lead}
  - o {project-property:location}
  - o {project-property:name}
  - o {project-property:[PROJECT NAME]|users}

Where [PROJECT NAME] is the name of a project. Example:

```
{project-property:drivepilot|users}
```

- o {project-property:start}
- o {project-property:users}

- {pages} macro
- {timepoint} macro
- {show:pdf} macro: the content is shown only when document is exported to Word

## Partially supported items

The following elements and macros are partially supported during export to Word. For example, content may export but formatting does not.

- Links: [Project/Space.Page], {portallink:/monitor}Monitor{portallink}
- {table} macro
- Boxes: #message(), #info(), #warning(), #error()
- User macros: {user}, {users}
- Dashboard macros: {dashboard-property}, {report}
- Factbase: e.g. {fact:base=repo-analysis|value=NOF}

# Import rules for tables

When importing a Microsoft Word document in which content that should be recognized as Work Items is contained in a table, you need to specify an import Rule that recognizes Work Items in tables. You may also want the Rule to run some action(s) on these table-contained Work Items when the document is imported to Polarion. This is done using regular expressions with the text matching regex rule type option. This topic provides some examples.

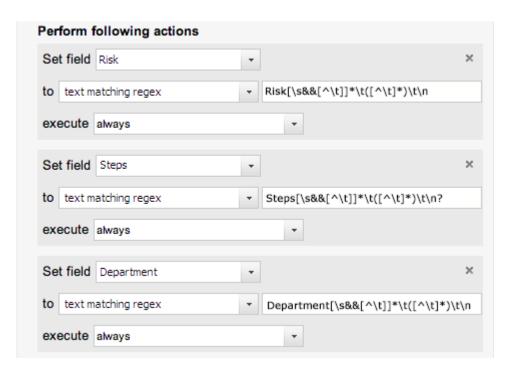
Consider the following table:

Risk	High
Steps	Hello World
Department	34

These regular expressions can be used to parse the values from the above table:

The selected fields must actually exist within the target document in order to work.

- Single line: Risk[\s&&[^\t]]\*\t([^\t]\*)\t\n
- Multiple lines: Steps[\s&&[^\t]]\*\t([^\t]\*)\t\n?



"text matching regex" rules specified in the Document import preview.

See the topic Configuring Automatic Work Item Recognition for information about recognition of other content elements.

#### Manage saved import configurations

If users have created and saved Document import configurations (i.e. rules for recognizing Work Items in imported Word documents), at some point in may become desirable to delete some configurations that are no longer necessary or used. There are several points to remember about deleting import configurations:

- There is no graphical user interface for deleting import configurations. They must be deleted from the repository.
- The repository folder storing import configurations is:

```
/.polarion/tracker/import_configurations/word Of
[PROJECT]/.polarion/tracker/import configurations/word
```

You can use Polarion's Repository Browser or an external Subversion client to browse to the import configuration folders.

- If an import configuration for the repository scope has been overridden in a project, the configuration cannot be deleted at the repository level.
- A user must have change permissions for the repository folder(s) containing import configurations in order to delete any import configuration.

See also: Saving Work Item Rules for Reuse.

# Import and export formatting

Polarion always attempts to preserve the formatting of imported Microsoft Word and Excel documents, and of exported Polarion LiveDocs. In general:

- Font size and family is imported from Word and Excel documents except the Normal style.
- Polarion imports only known fonts defined by the system configuration that are replaced on import by the font substitution settings of the Polarion system.
- Text color and text and table cell background color is imported from Word and Excel documents.
   Text background color set text strings in body text or table cells is *not*imported. In cells, only the cell background color is imported.
- Font size, font family and colors are exported during Word and Excel round-trip.
- Font size, font family and colors are exported by PDF Export

# Round-trip export template fields

The following IDs can be set in a project-scope Word Round-trip export template, and are visible in the header or footer of exported round-trip files:

ID	Description
created	Date/time when the Document was originally created
createdBy	User who originally created the source Document
date	Current date
documentName	System ID of the source Document
documentTitle	Title of the source Document
generated	Date the round-trip document was created
generatedBy	Name of the user who exported (generated) the round-trip document
productName	Name of the Polarion product used to create the round-trip document
productVersion	Version identifier of the Polarion product used to create the round-trip document
projectId	ID of the project containing the source Document in Polarion
projectName	Name of the project containing the source Document
revision	Revision number of the source Document
space	System ID of the space containing the source Document
spaceTitle	Title of the space containing the source Document

ID	Description
updated	Date/time when the source Document was last updated
updatedBy	Name of user who last updated the source Document

## Round-trip export template locations

A Round-trip Export template is a Word .docx document that is used by Polarion as a basis for creating a Word round-trip document from a Polarion Live Document. Templates can define e.g. header and footer, table of contents, heading styles, etc. This section specifies where such templates are stored in the repository in case you wish to customize them.

- The location of the global default template, used for export of Documents created in the portal is: /.polarion/tracker/templates/roundtrip.docx. This template is accessible via the Administration topic Work Items Round-trip Template in the Repository scope.
- You may create and upload a project-specific copy of the global default template to be used for all round-trip exports from a specific project. Project-specific templates are stored as: [PROJECT]/.polarion/tracker/templates/roundtrip.docx. You can create and access a project-scope round-trip template via the Work Items Round-trip Template in the project scope.
- If a Document is created by importing a Word Document, a document-specific round-trip export template file is created and stored in the Document folder: [DOCUMENT\_NAME]/templates/roundtrip.docx. This template is accessible in the Document Properties panel of the Document Sidebar. You can either upload a customized template (one that was downloaded before and modified), or it can be one you have created as a template out of any Polarion-created Document (these options are presented in the upload dialog).

For more information on using Word Round-trip and round-trip export templates, see: Sharing Documents Using Word Round-trip and Using Round-trip Export Templates.

# **Deleting Round-trip Templates**

If you have default round-trip templates in some project(s) that you no longer need, these can be deleted using the Repository Browser. Navigate to the project location as described above and delete the roundtrip.docx file.

#### Warning

Do not delete the global default round-trip template:

/.polarion/tracker/templates/roundtrip.docx

You can delete a Document-specific round-trip template using the Remove (use default) link in the Document Properties panel of the Document Sidebar in the Document Editor.

# **Document Editor toolbar**

This topic documents the actions available on the Document Editor toolbar. The toolbar is also available in the Test Execution View. Please refer to the following figure. Toolbar actions are listed in this topic as they appear left to right on the toolbar

Action	Description
<b>a</b> Save	Saves changes to the Document. Disabled when there are no unsaved changes.
Operations	Drops down a menu of available Document operations. See: Operations Menu.
<b>▶</b> Undo	Rolls back changes until the undo limit is reached.
<b>∼</b> Redo	Reapplies changes undone by Undo until the limit of undone actions is reached.
Normal Paragraph Type	Applies paragraph or heading styles to the current paragraph. The maximum number of available heading levels can be configured by an Administrator. See the com.polarion.tracker.defaultNumberOfHeadings property for details.
B Bold	Formats selected text, or text forward from insertion point, in a bold font.
I Italic	Formats selected text, or text forward from the insertion point, in an <i>Italic</i> font.
	Contains the following formatting options:
	<u>U</u> - Underline,
Arial - 10 -	ab- Strike-through,
<u>U</u> <u>ab</u> ײ ×₂ A ⋅ <u>M</u> ⋅	ײ- Subscript,
Farmatting	×₂- Superscript,
Formatting	A <sup>*</sup> - Text Color,
	A⁻- Text Background Color,
Arial *	
Font	Enables font selection.
10 🕶	Enables font size selection.
Font Size	

Action	Description
<b>夕</b> Remove Formatting	Removes formatting from selected text.
Convert to Text	Converts selected items to Plain Text.
	Clicking icon marks or unmarks selected text as a Work Item.
☑ Task Configure	Drop-down menu enables marking selected text as a work item.  Work Item types, if multiple types are configured in the Document Work Item presentation. The Configure item access the Configure Work Item Presentation dialog where type(s) and other propertie of Document Work Items are specified.
Insert Referenced Work Items	Enabled if current selection does not contain a Heading, Wiki content, TOC, or Work Item. Also means that selection does not contain two or more Work Items, or combination of more top level elements.
Mark/Unmark Work Item	Enabled if current selection contains only one Work Item and nothing else.
≟= Ordered List	Formats the current paragraph as a Numbered list (ordered list) item.
<b>:</b> ≡ Unordered List	Formats the current paragraph as an unordered (bullet) list item.
	Presents a toggle control enabling/disabling the continuation of a numbered list that is interrupted by text not part of a list item.
	Only top-level list items are affected. Numbering of further levels always begins with "1".
Continue Numbering	Continue Numbering is available for plain text, contained Work Items, and referenced Work Items.
	<ul> <li>Continue Numbering is available only in the Document Editor.     Work Items containing continued lists will show lists continuing     with "1" in other Views (e.g. Table), but the continuation is     preserved in the Document.</li> </ul>
	List continuation is preserved in Round-trip export for Word. Changes from Word should be re-imported correctly unless a list is interwoven with a different list type - bullet list, for example.

Description
Continuation is preserved in copy/paste operations within the same Document.
Outdents the current paragraph one level. Disabled if the current paragraph is already as far left as it can go.
Indents the current paragraph one level until the indent maximum is reached.
Sets the alignment for the current paragraph. The action will be denied if the cursor is located in a heading or Work Item text. In a Work Item description, only the alignment of images and tables is supported. (An entire table or just text in a single cell). An image is aligned only when selected.
■- Left aligns the current paragraph.
■- Center the current paragraph.
■ Right aligns the current paragraph.
Launches the <b>Insert Image or Attachment Preview</b> dialog with options to upload a local image, select an image attachment, or insert an image currently on your clipboard. The image is inserted at the current cursor position.
Enabled if current selection does not contain a Heading, Wiki content, TOC, or Work Item title.
ent Preview
erence
Insert several types of content into the Document. See: Insert Menu.

Action	Description
Insert Menu	
<b>■</b> Table	Inserts a table on the current line if it is empty, or on a new line if not. It is also used to modify an existing table (Add, Remove, Merge or Unmerge previously merged cells.)
Show Sidebar	Displays a menu of Document Sidebar panels and opens a selected panel.
O View History	Displays the Document's revisions history and revision comparison features.
Home	Returns the insertion point to the top of the Document.
Tree Table Live Plan Road Map Matrix Time Sheet	Displays a menu of Work Item Views and opens a selected View in the Work Item Tracker. The opened View contains all Work Items in the Document.
<sup>C</sup> Refresh	Reloads the current Document.
? Help Keyboard Shortcuts  Document Options	Presents a context-sensitive menu of options. The option to display Help is always present.

# **Insert Menu**

Action	Description
<b>₽</b> Comment	Inserts a comment marker at the current cursor position, opens the Comment sidebar, and inserts a new comment that's ready for editing.
	Enabled if the current selection does not contain a Heading, Wiki content, TOC, or Work Item title.
	Available on the toolbars found on Pages, LiveDocs and rich text Work Item fields. It launches the Insert Image or Attachment Preview dialog.
☐ Image or Attachment Preview	The dialog initially displays previews of images and attachments that are already attached.
	It is also where new images or attachments can be added.
ಆ Link or Cross-reference	Launches the <b>Insert Link</b> dialog where a <b>Live Link</b> URL or a <b>Cross-reference</b> can be specified. A hyperlink to the specified URL is inserted at the cursor position. If the Live Link or Cross-reference refers to an item that contains outline numbers, you'll also have the option to include them in the link's text. The cross reference's outline numbers and title update automatically whenever its source's numbers or title change. Enabled if the current selection does not contain a Heading, Wiki content, TOC, or Work Item title.
<b>.</b> Diagram	Launches the Diagram Editor and inserts a new diagram at the insertion point when diagram is saved in the Editor.
	Inserts a table of contents on the current line (if empty) or above it if not.
E Table of Contents	Enabled if current selection does not contain a Heading, Wiki content, TOC, or Work Item, <i>and</i> if there is no TOC already inserted in the Document. Only one TOC may be inserted in a Document.
≑ Table of Figures	Enables insertion of a table of figures, table of tables, or a table of other items with user-defined caption labels, on the current line, if empty, or above it if not.
	The type of table inserted depends on selection in the Caption Label field of the Insert Table of Figures dialog.
<> Wiki Content	Launches the <b>Insert or Modify Wiki Content</b> dialog in for selecting Wiki syntax and mark-up that inserts Wiki content (macros, for example) into the Document.
	Enabled if current selection does not contain a Heading, Wiki content, TOC, or Work Item.

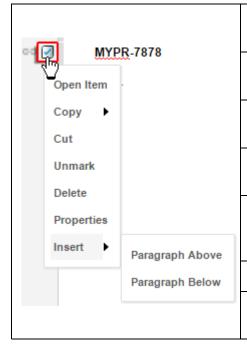
Action	Description	
☐ Page Break	Inserts a page break on a new line. Page breaks are applied when the Document is exported.	

# **Operations Menu**

Action	Description		
▼ Filter	Displays the filter bar where you can enter a query that filters the Work Items shown in the Document. For example, the query status:open would filter the Document to show only Work Items with an <b>Open</b> status.		
Q Find	Displays the Find dialog, enabling you to find text in the current Document. (Including text contained in Work Items.)		
Q Find and Replace	Displays the Find and Replace dialog, enabling you to find and optionally replace text in the current Document. (Including text contained in Work Items.)		
<b>₽</b> Print	Launches the Print dialog that displays a print preview of the Document and provides print options.		
<b>四 Export to PDF</b>	Launches the Export to PDF dialog with several export options and exports the Document content to a PDF file.		
E Rename	Launches the Move or Rename dialog with options to Rename the document or Move it to another Space.		
<b>♣</b> Reuse	Launches the Reuse Document dialog with options to reuse the current Document.		
<b>♪</b> Branch	Launches the Branch Document dialog for branching the current Document to create a variant.		
Compare With	Enables user to compare the current state of a Document with some other Document, or the last saved version of the current Document.		
₩ Word Round-trip	Via its sub-menu, let you Export the current document to Microsoft Word for review, comment, or editing by an external user, and Import Changes in a previously exported Word round-trip document back into Polarion.		
	For more information, see Sharing Documents Using Word Round-trip.		
ReqIF Round-trip	Lets you <b>Export</b> a document in the ReqIF (Requirements Interchange) format or <b>Import Changes</b> made to a ReqIF document outside Polarion, back into the portal.		

#### Work Item Actions menu

When Document text is marked as a Work Item, an icon representing the Work Item type appears in the left margin. Clicking the icon displays a menu with the following actions.



**Open Item** - Opens the Document-defined Work Item in the Table view of the tracker.

**Copy ID** - Copies the ID of the Work Item to your computer's clipboard.

**Copy Link** - Copies the complete URL of the Work Item to your computer's clipboard.

**Cut\*\*** - Copies the complete URL of the Work Item to your computer's clipboard.

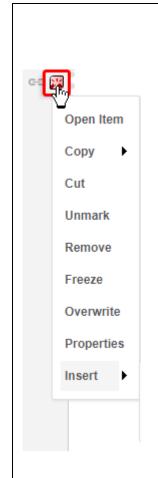
**Unmark** - Unmarks the current Work Item, reverting the content to regular Document text and sending the Work Item to the Document Recycle Bin (viewable in the Document Sidebar).

**Delete** - Deletes the current Work Item.

**Insert** - Inserts a paragraph above or below the current caret position via one of the sub-menu items.

#### Referenced Work Item Actions menu

A Referenced Work Item links to a Work Item that resides in another document. Its content can only be edited in the original Work Item, but if you use the **Overwrite** action, you can create a copy of the Work Item's content with a new ID and store/edit it within the current LiveDoc. (An is branched from link role to the original item is created automatically.)



**Open Item** - Opens the Document-defined Work Item in the Table view of the tracker.

Copy ID - Copies the ID of the Work Item to your computer's clipboard.

Copy Link - Copies the complete URL of the Work Item to your computer's clipboard.

Cut - Places the item in the Recycle Bin. (Viewable in the Document Sidebar.) You still have the option to Undo this operation and reinsert the item back into the LiveDoc.

**Unmark** - Unmarks the current Work Item, reverts the content to regular Document text and sends the Work Item to the Document Recycle Bin. (Viewable in the Document Sidebar.)

**Remove** - Removes the Referenced Work Item. (No option to undo once the LiveDoc is saved.)

**Freeze** - Freeze the Referenced Work Item at a specific revision.

Overwrite - Creates a copy of the Work Item's content with a new ID, stores the item within the current LiveDoc and creates an **is branched from** link role to the original item.

Properties - Launches the Work Item Properties sidebar.

**Insert** - Inserts a paragraph above or below the current caret position via one of the sub-menu items.

## **Document structure change via Excel**

The following items apply when Document Work Items are exported to Excel using Allow Document structure changes in Excel, and an external user modifies the Document structure in the exported Excel workbook.

- When a heading is removed in the exported workbook, lower level headings below the deleted heading are not promoted, but rather retain the same level they had upon export from Polarion. The user should change promote lower-level headings occurring after deleting a higher-level heading.
- An heading added in the exported workbook is always 1 level higher in the structure as compared to the previous heading.

- If it is desired to preserve non-sequential headings occurring in a Document... "Heading 1" followed by "Heading 3", for example... then modifying headings in an exported Excel workbook is not recommended. On import, Polarion will "fix" the structure, promoting the "Heading 3" in the example to "Heading 2".
- Related to the previous, any invalid hierarchy created in the exported workbook (missing heading level between Work Items) will be silently fixed when the modified workbook is re-imported to Polarion and the online source Document will reflect a valid hierarchy.
- When adding a row in the exported Workbook, the user must write some content before attempting to indent it.
- The following points apply when conflicts are encountered upon re-import of a modified exported Excel workbook:
  - o If a Work Item is unmarked or deleted on server after being included in an export to Excel, and the item is **not** modified in the exported workbook, the importer ignores the item in the worksheet and does not re-import it. ("Modified" means modified value, not level or position.)
  - o If a Work Item is unmarked on the server after being exported to Excel, and it is modified in the exported workbook, then item is updated during re-import to Polarion.
  - o If a Work Item is deleted on server after export to Excel, and then modified in exported workbook, the importer treats it as a conflict. The import fails, or changes to the deleted items are ignored, depending on the import option settings selected by the user importing the workbook.

#### Merge concurrent changes

This topic explains the actions that Polarion takes to merge changes to Documents when users edit the same Document concurrently.

# **Automatic merging**

The following concurrent Document editing actions are automatically merged without any message to users:

- Adding elements to different places
- Deleting elements on different places
- Changing of different elements
- Work Item indentation (treated as remove/add)
- Commenting of a paragraph or Work Item description by one user that was modified by another user since the last saved revision.
- Changing of a Work Item property after another user has reordered some other parts of the Document.

# Automatic safe merge with warning

The following concurrent Document editing actions are automatically merged with relative safety after a warning message to users:

- Change to different properties of the same Work Item. Warning appears to the latest user to invoke Save. Merge is performed if user confirms Save in the warning dialog.
- Merging of other changes if the same property(ies) of a Work Item are changed. Changes of latest user invoking Save overwrite the conflicting change in Work Item property and merges other changes if user confirms Save in the warning dialog.

# Automatic unsafe merge with warning

The merge algorithm tries to merge the changes made by the current user, but some of those changes may be lost or may not make sense after the changes made by another user meanwhile. The following actions will display a warning to the User that he/she should review all recent changes (in the Document history) and manually make corrections if necessary.

- Addition or removal of a Work Item after another user reordered some parts of the Document.
- Addition of elements (paragraphs, Work Item, etc.) to the same place in the Document.
- Removing elements that were modified or removed concurrently.
- Changing of the same Document element.

# **Merge actions for compared Documents**

This topic outlines the various merge actions available in the **Merge** sidebar of the Work Items page when two Documents are compared (Document Editor, Compare with menu item). Actions can be categorized as follows:

- Insert actions available if the target item does not exist
- Replace actions available if target is an overwritten Work Item
- Delete actions available if the source item does not exist in the target Document, or is in the Recycle Bin.
- Referencing actions available for Work Items.
  - Available when the target Work Item is referenced from the source Document or when the target Work item branched from source Work Item and has been overwritten."
- Field actions available for Work Item fields. Available when Work Items have differences at least in one field that is valid for merging.

For any selected Document or Work Item, the Merge sidebar displays only those merge actions that are valid for the selection in each merge direction. However, project mangers can opt to disallow some action(s), and an Administrator can change the Merge Actions configuration so that disallowed actions are never available in the sidebar. Therefore, you might not see some of the actions listed in this reference. It can be helpful to review the topic on Merge Terminology before proceeding.

#### Insert actions

Insert actions are available if an item in the source Document does not exist in the target Document.

- **Insert Live Reference**: Insert Work Item as a "live" referenced Work Item. Subsequent changes to the original will propagate automatically. Available if an item in the source Document does not exist in the target Document.
- Insert Frozen Reference: Insert Work Item as a referenced Work Item "frozen" to the current revision of the source item. Subsequent changes to the source item will NOT propagate automatically. It will be possible to manually update the reference to some future revision of the source item. Available if an item in the source Document does not exist in the target Document.
- Copy the Work Item: Copy the Work Item to the target Document creating a traceability link
  to the source Work Item. Item in the target Document will be a static copy, not a reference.
  Never updated unless manually replaced by another copy. Available when a Work Item in the
  source Document is selected for merging, and the target Document contains no copy of or
  reference to the source item.

# Replace actions

These actions are available if the target is an overwritten Work Item.

- Replace Overwritten Work Item with Live Reference: Send an overwritten Work Item in the target Document to the Recycle Bin, and replace it with a "live" referenced Work Item. Subsequent changes to the referenced item will propagate automatically.
- Replace Overwritten Work Item with Frozen Reference: Send an overwritten Work item in the target Document to the Recycle Bin, replace it with a referenced Work Item from the source Document, and freeze the reference to the current revision of the source item. Subsequent changes to the source item will *not* propagate. It will be possible to manually update the reference to some future revision of the source item.

# **Delete actions**

- **Delete Reference**: Delete the referenced Work Item from the target Document. Available if target item is a reference to a Work Item that does not now exist in the source Document.
- **Send Item to Recycle Bin**: Remove Work Item from the target Document but keep it in the Recycle Bin. Available when the target item is not a referenced Work Item.
- Delete Work Item: Delete Work Item from the Document and the repository (but not from History).
  - This action is disabled by default, and not visible to users. Allowing deletion of Work Items should be considered carefully. It may not be acceptable in regulated industries where process transparency and proof of compliance are required. If project management decides to allow it, the action can be enabled by an administrator in the Merge Actions configuration.

# Referencing actions

• Freeze Work Item Reference: Freeze the Work Item reference in the target Document at a specific revision of the source item. Available when the target item is referenced from the source Document.

- Unlink and Insert Live Reference: Insert Work Item as a "live" referenced Work Item. Subsequent changes to the original will propagate automatically. The existing traceability link between the source item and the target item will be removed. Both the existing target Work Item and the new referenced Work Item will appear in the target Document. Available when target is an overwritten item from source Work Item.
- Unlink and Insert Frozen Reference: Insert Work Item as a referenced Work Item "frozen" to the current revision of the source item. Subsequent changes to the source will NOT propagate automatically. The existing traceability link between the source item and the target item will be removed. Both the existing target Work Item and the new referenced Work Item will appear in the target Document. Available when target is an overwritten item from source Work Item.

# Merge fields actions

Merge fields actions are available if the merge target is a Work Item and...

- Target Work Item is one that was branched and then overwritten in the branch, or...
- Source Work Item is frozen to a revision, and target Work Item is contained in the target Document (i.e. not referenced) and source and target Work Items have field changes that can be merged.

#### **Actions:**

- Merge Some Fields: Merge fields specified in the Merge Actions configuration. By default this includes Title, Description and Test Steps, but this can be changed by an Administrator.
- Merge All Fields: Merge all fields of the Work Item that are supported for merging.

# **Fields Not Supported for Merging:**

When the Merge All Fields merge action is selected, all fields are merged in addition to **Title** and **Description** *except* the following fields:

- Comments
- Project
- Module
- Author
- Created
- ID
- Status
- Resolution
- Updated
- Location
- Planned Start

- Planned End
- Outline Number
- Linked Work Items Derived

# Non-Latin alphabets

This section provides reference information for users and administrators about features and configurations related to using Polarion with non-Latin alphabets.

The Polarion ALM architecture is designed to enable localization of the system to any language. Localized version of Polarion ALM are produced, provided, and supported by partners in various countries.

Localized versions of Polarion are currently available for the following non-Latin based languages:

- Chinese
- Japanese

# Non-Latin system properties

This section documents system properties that generally need to be set by administrators in the system configuration in order for some feature(s) to work with non-Latin characters or languages.

System properties are specified in the system configuration file polarion.properties.

```
\verb|com.polarion.customWordSpacingCharacters| \\
```

Language: Japanese. Enables Document comparison with languages that do not separate words with spaces. Specify a "break" character to be used when users compare Document versions in the Document history. Multiple characters can be specified and breaks occur on every one of them. **For example:** 

```
{\tt com.polarion.customWordSpacingCharacters=\u3000\u3001\u3002\uff08\uff09\uff0c\uff61\uff64$} \\ {\tt com.polarion.placeCommentsDirectly}
```

Language: Japanese. When set to true, then comments are placed directly under the cursor. If false, then comments are placed after current word.

#### Work Item queries

Users who switch between English and Japanese keyboard should note that when using the Japanese keyboard, entering Japanese characters in a graphical query element's panel (in the Work Items Table view) does not invoke the search action. The user must press Enter or Return to initiate searching. This is different behavior from English keyboard and characters, where search is invoked automatically with each character entered.

# **Diagram fonts**

Only diagrams that use Helvetica font can be properly rendered by the Document Editor into an image that is displayed in the content of a Document or Work Item. Making other fonts work with Asian or other non-Latin languages is fairly complex and requires considerable effort and will most

likely require assistance from your Polarion country partner or Polarion Professional Services. If you need this capability for diagrams, please contact Polarion Technical Support for guidance.

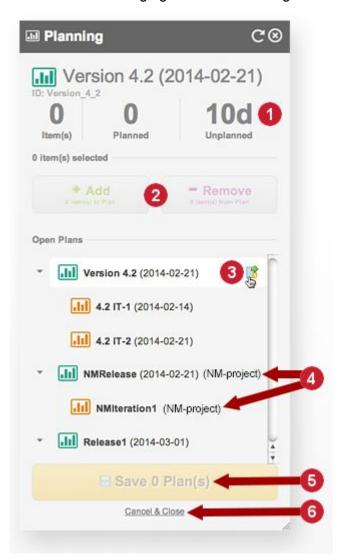
#### **Plans**

# Planning sidebar

This sidebar panel is available when viewing/editing a Plan in the Work Items Table, and when editing a Document. The Planning sidebar is opened automatically by the "Open in Table" action. (Actions menu, or Open in Table widget in a Plan. Cannot be invoked from a Document.)

The Planning sidebar enables you to load any open Plan in the table, or open its page. This can be useful when you are planning Work Items for a release or iteration. You can easily check any plan to see what Work Items have been added to it and what the workload is like, and to add or remove Work Items in different Plans until you have a release plan and any iteration sub-plans populated as you think best for successful a execution.

Refer to the following figure when browsing the next sections in this topic.



Selection in Open <b>Plans</b> . (Click to open this Plan.)	4 Plan from a different project.
Add or Remove item(s) selected in the table from the <b>Plan</b> selected in <b>Open Plans</b> .	Save changes to all modified Plans.
Current selection in Open Plans. (Click icon to load Plan in Table.)	6 Cancel changes to all modified Plans.

#### Selected Plan Title

The first element in the sidebar is the title of the Plan currently selected in the **Open Plans** list. The title is a hyperlink that will open the respective Plan in the Plans topic of the project. The ID of the Plan is also shown. This can be useful for developers needing to find the ID to cite it in a macro in a custom report page, or in an API call.

#### Selected Plan Work Statistics

After the Plan title and ID, the Planning sidebar shows several statistics about the Plan currently selected in **Open Plans**.

- **Items**: a count of the number of Work Items currently planned for processing in the selected Plan.
- **Planned**: the amount of work estimated for all Work Items in the selected Plan. The metric may be time, or something else defined in an estimation field, depending on the configuration in the Plan properties.
- **Unplanned**: Reports the Capacity (defined in plan properties) minus planned work. Planned work is the sum of all work already done and still to be done in the Plan.

#### Add / Remove buttons

These buttons can be used to add Work Items currently listed in the Table to the Plan currently selected in the Planning Sidebar. Keep in mind that the Plan currently loaded in the Table may not be the Plan currently selected in the Planning Sidebar. Observe the Query Builder in the table to see if the **Planned In** query element is present. If so, then the Work Items in the table are already added to the Plan specified in the query. Normally, you should not add items from that Plan to some other Plan selected in the sidebar, without first removing it from the queried Plan.

Assuming the query in the table is not fetching items that are already planned, you can add items from the table to any open Plan by first selecting it in the **Open Plans** tree in the sidebar. Select one or more items in the Table, and click the **Add** button. Alternatively, you can drag items from the Table to the Planning Sidebar to add them to the currently selected Plan.

If the Plan loaded in the Table is the same as the Plan selected in the sidebar, you can select items in that Plan for removal from it. Select the items to remove, and click the **Remove** button in the sidebar.

Remember that no Plan is actually updated with changes made in the Planning sidebar until the changes are saved via the **Save** button. The button shows a count of Plans that have changes pending.

### Open the Plans tree

This section of the Planning sidebar is a scrollable list of currently open Plans in the current project. If any Plan has child Plans, its node can be expanded or collapsed to show or hide the child Plans. If a plan is one in a different project that has been included in the currently-viewed plan, the ID of that project appears in parentheses after the Plan name.

Selecting a Plan in the tree shows its current work statistics. It is important to note that selecting a Plan here does *not* load its Work Items in the Table. If you want to do that, hover over the Plan when it is selected in the tree, and click on the icon that appears (Show Plan Work Items in Table).

#### The Save button

You can use the Planning sidebar to jump around among different Plans and make changes (add Work Items, remove Work Items). However, no Plan is actually updated until you click this button. The button label displays a count of how many Plans have changes pending.

# **Cancel & Close Link**

Clicking this link cancels all unsaved changes to all Plans, and closes the Planning sidebar. You can reopen the sidebar from the Sidebar drop-down menu, located to the right of the Query Builder in the Work Items Table.

#### Plan macros

#### Warning

The information in this section is now considered deprecated. It is still provided here for those who have legacy Plan reports implemented as Classic Wiki pages. New Plan reports should be created as LiveReport type Pages using the set of Plan-related Widgets. For more information see: Creating Plan Reports.

This topic lists the macros you can use in Classic Wiki pages (including the underling wiki page of Plans) to report information about Plans. Syntax details and usage examples for these macros can be found in the Plans section of the embedded Wiki Syntax Help.

# Work Items Board - {workitems-board}

The {workitems-board} macro renders an easy to understand summary of the Work Items in any Plan. The Work Items Board can be configured with columns summarizing the Plan's Work Items in different states, such as To Do, In Progress, Verified, and Done. Note that only following Work Items are shown on the Work Item board:

See Work Items Board to add a drag and drop board to like LiveReports, like Info Pages and like Plans.

- **Parent** items: Work Items added to the Plan specified in the plan parameter.
- Child items: Work Items with a link to those added to the specified Plan, and where such added Work Items have the *parent* link role to them (by default, if no roles parameter is specified)
- Work Items with status defined in the columns parameter of the macro .

# Burn Up and Burn Down - {plan-burnup} {plan-burndown}

The {plan-burndown} macro renders a Scrum-style "burn-down" chart of a Plan's Work Items. The {plan-burnup} macro, renders a "burn-up" chart of the Work Items.

In order for the Burn-down or Burn-up charts to show correct data, it is necessary to:

- Specify the Plan's Start Date and Due Date in the Plan properties.
- Add Work Items to the Plan (see: Planning Work Items)
- Be sure all Work Items in the Plan are estimated (i.e. the **Initial Estimate** field should be filled in for every Work Item).
- Start the Plan on the same date specified in the Plan's **Start Date** property. That is, the value in the **Started On** property should match that in the **Start Date** property.

After the Plan is started, avoid changing Start Date or Started On date to keep progress reporting accurate. You may, however, add Work Items to a Plan after it has started, and the total work and ideal progress statistics will be updated accordingly. However, adding Work Items after Plan start is not a recommended best practice.

#### Plan open in table - {plan-open-in-table}

The {plan-open-in-table} macro renders the Open in Table button in the Plan page, which, when clicked by a user, opens a new browser tab and loads the Work Items Table filtered to show the Work Items currently in the Plan (by means of the Planned In visual query element.)

#### Plans - {plans}

The {plans} macro displays open Plans that are child Plans of the one specified in the macro's plan parameter, and also Plans set by the query defined in the macro.

#### Plan status button - {plan-status-button}

The {plan-status-button} can be used in **Plans only**. It will not work in Wiki pages. It renders a button that shows current status of the Plan and enables the user to set a different status. If status is set to "Done", the Plans Finished On property is set to the current date/time.

# Plan label - {plan-label}

The {plan-label} macro shows the name and hierarchical structure of the Plan specified in the plan parameter of the macro. Users can click on the displayed parent or child Plans to jump directly to those Plans. The macro also provides an action for creating a new child Plan of the specified Plan.

#### Plan progress - {plan-progress}

The {plan-progress} macro displays the progress bar of the Plan specified in the plan parameter of the macro.

The macro starts showing the **Done** value only after the Plan is started. Otherwise, it shows only the ideal progress metric.

# **Ideal Progress statistic**

Plans report various statistics that reveal to the team and managers the current state and overall progress of the Plan. Most statistics are self-explanatory. One statistic, named Ideal Progress needs some background in order to correctly understand what it means.

The calculation formula for this statistic can be expressed as follows: Ideal Progress = (D1 / D2) \* T where...

- **D1** = the number of working days from Plan's start date to today
- **D2** = number of working days in Plan
- **T** = Total Work (aggregate of the Initial Estimate field values of all Work Items in the Plan)

Working days are per the Working Calender in the project configuration. Keep in mind that projects use the Global Working Calendar configuration by default. If working days for the project differ, be sure to create a project-scope Working Calendar so that Ideal Progress is calculated correctly for Plans in the project.

# **Previous Time Spent property**

The **Previous Time Spent** property of Plans stores the sum of all non-null values present in the **Time** Spent field of Work Items that have been added to the Plan. It affects statistics calculations of Plans in progress if these are calculated on the basis of time. In this type of Plan:

- When the Plan is not started yet, time spent is not reflected in the planned work. For example, if a user story is added to an Iteration Plan that is in progress from last iteration (e.g. 1d remaining, 2d spent), only 1d is included in the Plan's total work, not 3d.
- When the Plan status changed to In Progress (i.e. the Plan is started), the aggregated time spent value from all of the Plan's Work Items is written to the **Previous Time Spent** property. In the example above, the 2d time spent value would be written.

Once the iteration has started, the total planned work effort is calculated as: To Do + Current Time Spent - Previous Time Spent. If, during the iteration, a user reports an additional 6h done (so time spent = 2d 6h), but remaining To Do is 6h, then the planned work on Burn-up is Day 1: 1d, Day 2: 12h (22h + 6h - 16h = 12h). Users see the plan has been increased by 4h.

# **Templates**

This section provides reference information on different types of templates used in Polarion.

#### To access the Project templates

- 1. Click on the top left corner.
- 2. Click Administration.
- 3. Click **Project Templates**.

#### **Polarion Templates**



#### **Custom Templates**



My Custom Template

( Templates include demo data.)

We provide both empty and demo data versions of each of the templates. The demo data versions contain the sample Documents, Work Items and the additional data you need to test drive the workflow and get a jump on setting up your own projects.

See the 'Default Project Templates' table below for details on each template.

The default project templates are stored in your Polarion installation in polarion/plugins/com.polarion.alm.projects\_x.x.x/templates (where x.x.x are numbers). Folders under this one contain the default project templates distributed with Polarion.

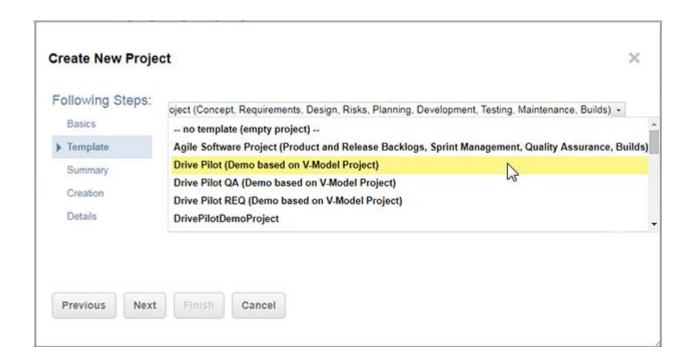
In 2017 our templates were rebuilt from the ground up using our LiveReport™ technology. This lets users make use of our growing library of report widgets like the drag and drop Work Items Board.

# Planned Items



Long-time users of Polarion can still download our Classic Wiki Project Templates and Plan Macros are still supported.

When selecting a new template in the Create New Project dialog, keep in mind that your system may be configured with custom project templates, or you don't have an ALM license, so you may only have some of the templates described in the table below.



# Tip

You can find more project templates on the Polarion ALM Extensions portal at <a href="http://extensions.polarion.com">http://extensions.polarion.com</a>.

Template Name	Product / License	Description
Agile Software Project  (Product and Release backlogs, Sprint Management, Quality Assurance, Builds)	ALM	An Agile based Project template that supports the full software application lifecycle from start to finish. It includes the management of product and release backlogs, sprint management and reporting, and test management with full quality assurance coverage.  It has a standard Java project structure that supports the automated build process managed by Maven-2.  Project with demo data: E - Library
V-Model Project  (Concept, Requirements, Design, Risks, Planning, Development, Testing, Maintenance, Builds)	ALM	A V-Model (Waterfall) based Project template that supports the full application lifecycle. It provides all that's needed to determine and define requirements, design then assess the risks, plan the development, monitor the development process, test the results and maintain your product.  It has a standard Java project structure that supports the automated build process managed by Maven-2.  Project with demo data: Drive Pilot

Template Name	Product / License	Description
V-Model Project QA  (Concept, Requirements, Design, Risks, Development, Testing, Maintenance, Builds)	ALM, QA	A V-Model (Waterfall) based Project template that includes: Requirements Specifications and their formal approval along with Risk and Test management. You can start by filling in the Requirement Specification documents, then target requirements for release and analyze potential risks by using the FMEA Worksheet templates. Then it's time to manage test execution. (Both manual and automated are supported.)
iviaintenance, builds)		It has a standard Java project structure that supports the automated build process managed by Maven-2.
		Project with demo data: Drive Pilot QA
V-Model Project REQ (Concept, Requirements, Design, Risks, Maintenance)	ALM, REQ	A V-Model (Waterfall) based Project template that supports Requirement and Test Specification documents. Start by filling in Requirement Specification documents. Then refine them in Test Case Specification documents and specify the validation and verification procedures using the Validation and Verification Plans. It supports a formal specification approval process.
		Project with demo data: Drive Pilot REQ
Specification Project with Variant Management (Requirements, Testing, Variants, Features)	ALM, QA, REQ	A V-Model (Waterfall) based Software Engineering Project configured for Feature, Requirements, and Test Specifications that includes Variant Management with a Propagation option. (If the Propagation option is enabled, the selection of Variant Work Items is determined by the relations to their parents and the restrictions of these parents.)
		Project with demo data: Weather Station

For more information about templates, see Customizing Project Templates.

# **Default Excel Round-trip templates**

Formatting of Excel Round-trip output is controlled by an export template. Polarion provides the following export templates by default:

Template File	Description	
Basic.xlsx	A generic export template providing basic formatting sufficient for many needs.	

Template File	Description
Empty.xlsx	A template in which no columns are specified for the export. User fully controls which columns are included by specifying them in the Export Work Items dialog.
FMEA Risk Worksheet.xlsx	A template specially configured with columns corresponding to fields for Risk type Work Items used in risk analysis specifications.
Time Report.xlsx	A template which provides fields and some calculations for work time reporting.

The default export templates may be customized, or customized copies created.

The default export templates are accessible in Administration Work Items Export Templates XISX: Microsoft Excel Round-trip Document.

See: Customizing Export Templates for additional information on customizing export templates.

# **Test Management Reference**

# **Permissions and license**

# ALM REQ QA

Your ability to access and execute Test Runs depends on the permissions assigned to your user account, which are normally tied to the user role(s) assigned to you by the system or project administrator. The license you are currently using may also affect your ability to use some test management features.

You must have the following permissions in order to execute manual Test Runs:

- Permission to EXECUTE You must have this permission for Test Runs in order to be able to execute a manual Test Run.
- Permissions for Work Items Because Test Runs can read, create and modify Work Items, you
  must have the following permissions for Work Items in addition to the EXECUTE permission
  for Test Runs:
  - o Permission to READ
  - Permission to MODIFY
  - Permission to CREATE NEW

#### **Test step attachments**

When manually executing a Test Run, you can optionally add a file attachment to each individual test step, or to a test record. For example, you might attach a screenshot image illustrating the result of a failed test step.

This section describes several important points you need to know about these attachments.

Click Add Attachment to attach a reference file.

- Hover over an already attached file and click to remove it.
- You can attach a file to a test step or a test record, or remove an attachment from these, while text execution is in progress.
- Users can download attachments from Test Steps or the **Test Records** section of Test Case type Work Items in the Work Item Tracker.
- Users can download files attached to a test step result or a test record if the test record is visible on the page of a Test Run.
- Users can download files attached to a test record if the test record is visible in the Test Run Records view.
- Attachments are displayed in any Defect items created as a result of failed test steps.
- Attachments are not exported to, or imported from Excel during round-trip operations.

# xUnitFileImport job parameters

This section documents the parameters of the xUnitFileImport job which enables importing results of externally executed tests into Polarion, creating a Test Run in the process.

Parameters are optional unless otherwise specified. Parameters for importing test results during Mavin builds are also documented, and an example of the job configuration is also provided.

Name	Description		
path	<b>Required.</b> Path to xUnit file, or to a folder containing xUnit files. (All XML files are imported and then deleted.)		
project	<b>Required.</b> ID of the project in which the job will create test case and defect Work Items as output of the job.		
idPrefix	Prefix for the ID (composed of prefix+timestamp) of Test Run created as output of the job.		
userAccountVaultK	Reference to user authentication key contained in the User Account Vault in  **Administration    **Description** User Management   User Account Vault		
maxCreatedDefects	The number of failed tests before a summary Defect type Work Item will be created instead of individual Defect items for every failed test. (The default is 20.)		
maxCreatedDefects	The percentage of tests that can fail before a summary Defect type Work Item Petwill be created instead of individual Defect items for every failed test. (The default is 10%.)		
templateTestRunId	ID of the template Test Run on which the Test Run created by the job is based. The template Test Run must exist in the same project specified in project.		

Name	Description	
templateTestCaseI	ID of template Test Case type Work Item used as basis for Test Case items created by the job. The template Test Case must exist in the same project specified in project.	
templateDefectId	ID of template Defect type Work Item used as basis for Defect items created by the job. The template Defect must exist in the same project specified in project.	
idRegex	Fills ID of the created Test Run with regex matching the xUnit filename.	
groupIdRegex	Fills group.id of the created Test Run with regex matching the xUnit filename.	
field[N]	Replace [N] with a number between 1 and 3. Specify up to 3 fields that can additionally filled with regex matching xUnit filename.	
Replace [N] with a number between 1 and 3. When field[N] is specthis parameter to fill Test Run field[N] with regex matching xUnit filence.		

See the Monitor Topic and Configuring the Scheduler for more information.

# Parameters for import in Maven Builds

Test results may be imported to a created Test Run during Maven builds by configuration in .polarion/builder/build.properties.

The parameters are identical to the parameters for the xUnitFileImport job, except that all must be prefixed with polarion.build.xUnit.import.

Unlike the xUnitFileImport job, however, xUnit files are not deleted after import.

The following parameters must also be specified in the Maven build configuration.

Parameter	Description	
polarion.build.xUnit.import	Set to true to enable import of test results during the Maven build process.	
polarion.build.xUnit.import	When import is enabled, specified the relative path to relative path to artifactId\target\surefire-reports in the job's working directory.	
polarion.build.xUnit.import	Set to true to import all xUnit files to a single Test Run. Set false to import different xUnit files to separate Test Runs.	

# Job example

```
<maxCreatedDefectsPercent>5</maxCreatedDefectsPercent>
    <templateTestRunId>20110718-151106/templateTestRunId>
    <templateTestCaseId>WI-12</templateTestCaseId>
    <templateDefectId>WI-5</templateDefectId>
    <idRegex>(.*).xml</idRegex>
    <groupIdRegex>[^-]*-(.*).xml</groupIdRegex>
   <typeRegex>([^-]*)-.*.xml</typeRegex>
    <field1>fieldId</field1>
    <field1Regex>[^-]*-(.*).xml</field1Regex>
</job>
```

# xUnit file example

Here is an example of an xUnit file showing only the tags and attributes read by Polarion. There follows a screenshot of the result of importing the example file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example shows only tags, attributes and content which is
currently read by Polarion
     See http://windyroad.org/dl/Open%20Source/JUnit.xsd -->
<testsuites>
<!--Name of the root tag does not matter, but it must not be same
as the ones below -->
    <testsuite timestamp="2011-10-17T23:05:06">
        <!-- testsuite tags can be nested, timestamp is not required and
format is "yyyy-MM-dd'T'HH:mm:ssZ" -->
        <testsuite>
           <testcase name="someMethod" classname="SomeClass" time="0.285"/>
<!-- test case id is "SomeClass.someMethod", time is not required
and is in seconds -->
        </testsuite>
        <testcase name="otherMethod" classname="SomeClass" time="0.001">
            <failure message="failure message" type="package.Exception">
                <!-- message and type are not required, all text content is
added to the created defect -->
                Failure details
            </failure>
        </testcase>
        <testcase name="otherMethod" classname="OtherClass" time="0.002">
            <error message="error message" type="package.OtherException">
                Error Details
            </error>
        </testcase>
    </testsuite>
</testsuites>
```

# Tests - example (xUnit Test Manual Upload)

Result	Test Case	Defect	Duration	Executed By	Executed
√ Passed	EL-387 - SomeClass.someMethod		0.285 s	System Administrator	2011-10-18 01:05
<b>X</b> Failed	€ EL-388 - SomeClass.otherMethod	<b>≅</b> EL-389	0.001 s	System Administrator	2014-05-22 11:31
	ackage.Exception e: failure message				
37.52.0	ilure details		0.002 e	Sustam Administrator	2014_05.22 11:31
Blocked	ilure details  BEL-390 - OtherClass.otherMethod ackage.OtherException e: error message		0.002 s	System Administrator	2014-05-22 11:31

# **Test Records query extender**

Polarion has an embedded SQL database layer that can facilitate Work Item queries in traceability reports and other complex querying needs. One common need is some query function that allows you to search for Work Items (typically test cases, or defects) according to test records... finding "failed test cases executed by me in last week", or "find Defects triggered by Test Cases",for example.

The TEST\_RECORDS keyword, with parameters, can be used in the Work Items table for such kinds of queries. It bypasses the usual Lucene query engine and queries the index of the SQL database layer.

Syntax: TEST\_RECORDS: (testrunId, result, executedBy, executed)

#### Parameters:

- testrunId The ID of a Test Run, qualified with the ID of a project. For example: PRJ1/TR-200
- result An option from the Result enumeration (configured in Administration > Testing). Accepts wildcard '\*' (meaning 'any').
- executedBy The ID of a user who executed the Test Run. Accepts wildcard '\*' (meaning 'any').
- executed Time frame of the Test Run execution. Can be a single date or date range. Accepts wildcard '\*' (meaning 'any').

# **Examples:**

TEST RECORDS: (proj1/testRun1)

```
TEST_RECORDS: (proj1/testRun1, "good result, but not enough")
TEST_RECORDS: (proj1/testRun1, failed,*, 20120130)
TEST_RECORDS: ("proj1/testRun1", failed, johndoe, 20120101 TO 20120201)
```

# **Additional examples**

The following table shows several examples for the TEST RECORDS element.

Example	Notes
TEST_RECORDS:("myProject/Smoke21","failed")	Finds all failed test cases in Test Run Smoke21
TEST_RECORDS:("myProject/Smoke22","failed",	Finds all Defects linked from failed test records *for Test Run Smoke22
<pre>TEST_RECORDS:("myProject/Smoke23",*,"johnd' TO 20120703")</pre>	r, Finds alb test cases of Test Run Smoke 23 that were executed by user johnd on 2nd or 3rd July, 2012

#### Note

The TEST RECORDS element may be used as a sub-query inside another query. For example:

```
TEST_RECORDS:("myProject/FullTest15","failed") AND severity:smoke AND
(assignee.id:johnd OR assignee.id:janeb)
```

# **Work Items**

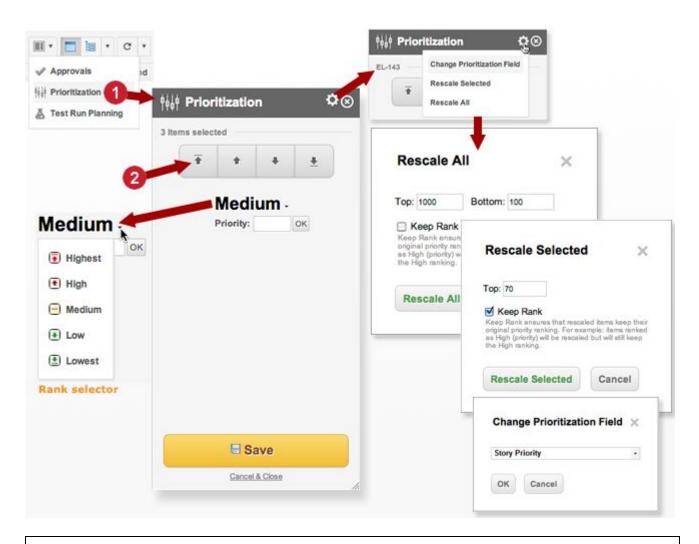
#### Work Items overview

This section provides reference information related to Work Items and the Work Items index and index fields. There is also information for those using Polarion's implementation of the Apache Lucene query language for embedded queries in configuration files, Wiki pages, API calls, etc.

- Prioritization sidebar,
- Work Item field permission restrictions,
- Work Item and Index fields.
- Advanced Work Item Querying.

#### **Prioritization sidebar**

This panel can be used to assign a Priority field value to multiple selected Work Items in the Table or Tree views of the Work Item Tracker (**Navigation** — **Work Items**). For conceptual and procedural information, see Prioritize Work Items. This topic provides information about the components of the panel. Please refer to the following figure:



- Prioritization sidebar
- 2 Reposition items

# **Reposition buttons**

These buttons move a selected Work Item or a selected group of Work Items upward or downward in the table. On move, selected items are rescaled according to the new position in the table. The buttons, left to right:

- Move selected item(s) to the top of the table.
- Move selected item(s) up one row in the table.
- Move selected item(s) down one row in the table.
- Move selected item(s) to the bottom of the table.

#### Rank selector

Displays a drop-down menu showing the priority ranks currently configured for the project. The priority rank of all selected Work Items is set to the selected rank on Save.

# **Priority field**

Accepts a float value. The numeric priority value of all selected Work Items is set to the specified value on Save, overriding any value previously set in the Rank Selector.

# Change prioritization field

Enables you to prioritize the items in the table on a custom field rather than the standard Priority field. The pick list displays all fields that are valid to use for prioritization in the current context. For more information, see Prioritize on a Custom Field.

# Dialog: Rescale selected

Rescales the numeric priority value of all Work Items currently selected in the table, limiting the maximum value to that specified in the Top field, and/or the minimum value to that specified in the **Bottom** field (if that field is present in the dialog... it is only present when relevant).

If Keep Rank is checked, none of the selected items will have their value set above the maximum value for their current rank. For example, if a selected item currently has rank "Medium", and rescaling would set the value higher than the maximum for that rank (thereby giving it a rank of "High"), that item will not be rescaled to any value not within the "Medium" range.

# Dialog: Rescale all

Rescales the numeric priority value of all Work Items in the prioritization table. Items will be rescaled within the range specified in the **Top** and **Bottom** fields.

Keep Rank functions in the same way as in **Rescale Selected**, except that it operates on all items in the prioritization table.

# Tip

If you are prioritizing a set of Work Items that all have the same priority, as is usually the case when the items were created by import from a Microsoft Office document, it is advisable to run **Rescale All** before beginning to manually prioritize the items.

#### Work Item field permission restrictions

Administrators can configure restrictions on permission to read and/or modify Work Item fields. In particular, the permission to MODIFY a Work Item field can impact users' ability to perform some operations.

The following sections describe this impact:

#### **New Work Items**

If a user is denied permission to MODIFY any Work Item fields, the ability to create new Work Items is limited as follows:

Required field?	Default value configured?	Result for user
YES	YES	Create is allowed. Default value is used, and user cannot modify it.
YES	NO	Create is disallowed. Message is displayed to user.
NO	YES	Create is allowed. Default value is used, and user cannot modify it.
NO	NO	Create is allowed. No value is set. User cannot modify field.

## Imported or re-imported Work Items

If a user is denied permission to MODIFY some Work Item field(s), and the user imports new Work Items or re-imports round-trip Work Items which hold values for the restricted fields, those fields are ignored by the importer, the field values are not imported, and the user sees a message about the restriction.

#### Work Item and Index fields

This topic documents the standard Work Item data fields. The field names are typically shown in Work Item form and in the column labels of the Work Items table. They also appear in the helper panel of the visual query builder.

When constructing textual queries using the Lucene query language, you will need to reference Work Item field IDs (rather than field names). IDs are needed for some configurations, macro parameters, API calls, etc. The table below lists all available fields and their ID values. Due to the way Polarion indexes Work Item data, there are some Work Item fields whose field IDs *must not be referenced in queries*. Such fields have one or more searchable Index IDs, which should be used in queries. These are noted in the **Index IDs** column of the reference table. For example:

```
{workitems: timePoint.time:[20120101 TO 20120212] AND assignee.id:$me | fields=id, title, severity, assignee, timePoint | sortby=timePoint created}
```

## **Note**

The field IDs of the Time Point and Assignee fields are not referenced in the query portion of the macro. Rather, the index IDs' timePoint.time and assignee.id are used.

There are also some IDs that can be used in queries, but which are not fields of the Work Item itself. Rather, they are fields in the Work Item *index*. Consequently there is no Field ID, name, or description. These are noted in the **Index ID** column of the reference table.

When a field is tokenized, it means that is it possible to search objects by sub-strings contained in the field. This is most useful for querying for values in text fields such as **Title** and **Description**.

Sortable index fields can be used for ordering of the query results or searching by range. These index fields contain special converted values (e.g. lowercase) of the fields, which are computed by special index functions.

Field name	Description	Field ID	Index IDs	On form
Approvals	Field that stores record(s) of approval(s) of the Work Item.	approvals	approvals (tokenized) approvalState Uses a special index function and therefore special syntax. See Query for Approvals.	yes
	This field defines the person responsible for a Work Item. The specified person will receive automatic notification(s) about the assignment and any changes to the Work Item.			
Assignee	A Work Item can be automatically assigned to a user specified in the auto-assignment configuration by selecting the value automatic in this field's combo box. This is the default for all new Work Items.	assignee	assignee.idassignee.name (sortable, lowercase)	yes
	For information about the auto-assignment configuration, see: Configure Auto-assignment.			

Field name	Description	Field ID	Index IDs	On form
			attachments	
			(tokenized)	
			attachments.author.id	
			(sortable)	
			attachments.author.name	
			(sortable)	
			attachments.content	
	Field that stores		(tokenized)	
Attachments	table of attachments,	attachments	attachments.fileName	yes
	including link to attached file(s)		(tokenized)	
			attachments.length attachments.length.1 (one word)	
			(sortable)	
			attachments.title	
			(tokenized)	
			attachments.updated	
			(sortable)	
Backlinked Work Items	Derived field containing links pointing from other Work Items to the referenced one (e.g. WI-1). These links are not stored in the WI-1 item, but are displayed in the Linked Work Items section of WI-1.	backlinked WorkItems (All one word)	backlinkedWorkItems	via linked WorkItems (All one word)

Field name	Description	Field ID	Index IDs	On form
Categories	This field enables you to flexibly categorize Work Items according to subsystem, customer, or whatever criteria you need. Category values provide an additional parameter that can be used in search queries. You can assign one or multiple Category values to any Work Item. For information on configuring Categories.  See: Configure Categories.	categories	categories.id categories.name (sortable, lowercase)	yes
Comments	This multiple entry field stores free-form text comments. Users can create multiple comments and comment threads in this field throughout the life of a Work Item.	comments	comments.author.id comments.text (tokenized) comments.title (tokenized)	yes
Created	The date a Work Item was created. The non-editable date value of this field is automatically assigned by Polarion and appears in the footer of the Work Item Viewer/Editor	created	created (date only) created.1 (datetime) created.2 (sortable)	permanent

Field name	Description	Field ID	Index IDs	On form
Custom Field (general)	Custom Fields can be defined by administrators in  Administration  Work Items  Custom Fields.  There are different types of custom fields such as string, text, enumerations, etc. All custom fields use three different index fields, where the first one is tokenized, the second one is sortable and the third one is a keyword (that is to say, not tokenized).	customFieldId	customField (tokenized) customField.1 (sortable) customField.KEY	yes
Description	Field that stores free-form text that describes what assignee(s) must do to resolve the Work Item.	description	description (tokenized)	yes
Document	The Document in which the Work Item is contained.	_document _space Each must be configured on the Work Item form separately.	document.id (sortable) Uses special index function. See Query for Documents.	yes
Due Date	Date when the Work Item should be competed. It overrides any completion date calculated by the LivePlan planning engine.	dueDate	dueDate (sortable)	yes

Field name	Description	Field ID	Index IDs	On form
Externally Linked Work Items	Field that stores links to Work Items on another Polarion instance (if the system is so configured).	externally Linked WorkItems (All one word)	externallyLinkedWorkItems (URL) externallyLinkedWork Items.host (All one word) (hostname)	via linkedWork Items (All one word)
Hyperlinks	Field that stores hyperlinks to other resources inside the Polarion portal, or remote (internet URLs, for example)	hyperlinks	Not searchable	yes
ID	Identifier string that uniquely identifies a Work Item in the Polarion repository.  No field label in the Work Item detail.  May be displayed as <b>ID</b> in table columns, dialogs etc. elsewhere in the user interface.	id	id id.1 (tokenized) id.2 (sortable)	permanent
Initial Estimate	This field is used in Polarion's project and time planning and planning accuracy metrics features. Here users specify an estimate of the amount of time that will be needed to resolve and close the Work Item.  Time unit is specified with shorthand these characters: d for day(s), and h for hour(s).  Time duration is specified by an integer value, or	initialEstimate	initialEstimate initialEstimate.1 (sortable)	yes

Field name	Description	Field ID	Index IDs	On form
	the literal 1/2 (or any other fraction). Examples:  • 3d - 3 days  • 7h - 7 hours  • 1/2d - 1/2 day			
Linked Revisions	Field that stores links to repository revisions that are stored with the Work Item.	linkedRevisions	linkedRevisions	yes
Linked Revisions Derived	Field that stores links to repository revisions that are derived from recognized IDs in commit messages.	linkedRevisions Derived (All one word)	linkedRevisionsDerived	via linked Revisions (All one word)
Linked Work Items	Field that stores links to other Work Items, either in the same repository or another repository hosted on another Polarion instance (if the system is so configured).	linkedWorkItems	linkedWorkItems Uses special syntax. See Query for Linked Work Items.	yes
Location	Field that stores repository ID and path where the Work Item is stored in the repository. Can be used e.g. to filter only items that are contained in Documents from certain space.	location	location location.tokenized (tokenized) location.tree	no

Field name	Description	Field ID	Index IDs	On form
Outline Number	A derived field that stores the outline number of a Work Item, if the item is contained in a Document. Users can search Work Items by outline numbers (also works for partial searches, e.g. 1-2*).	outlineNumber	outlineNumber Uses special syntax. See Query for Documents.	yes
Parent Work Item	A derived field that stores the IDs of parent items to the current Work Item.	parentWorkItem	parentWorkItem  Must be specified.  Is in the form  parentWorkItem:  ProjectId/WorkItemId  (All one word)	no
Planned In	Field of type 'plans', stores the names of all Plans to which the Work Item has been added.	plannedIn	plannedIn (not sortable)	yes
Planned End	Field that stores a date when work on the Work Item is planned to end. Value is normally supplied by the LivePlan project planning engine.	plannedEnd	plannedEnd (sortable)	via plannedTo
Planned Start	Field that stores a date when work on the Work Item is planned to start. Value is normally supplied by the LivePlan project planning engine.	plannedStart	plannedStart (sortable)	via plannedTo

Field that stores a constraint on date when work on the Work Item is planned planning
Constraints to start or end. Value forces the LivePlan project planning engine to plan the item according to the value set.  Constraints  Constraints  (All one word)  Yes
The Priority field indicates the relative importance and the order in which a Work Item should be processed. This field is used by programmers and engineers to prioritize their work. Work Item with the highest priority should be processed first.  The default values for the Priority field are configured in enumerations. In the system, the value is actually a float number. In the configuration you can define names for certain numerical values. For example: Highest - 90, High - 70, Medium - 50, Low -30 and Lowest - 10. Define values and names in the relevant enumeration configuration. For information, see: Configure

Field name	Description	Field ID	Index IDs	On form
	When editing a Work Item, you can specify your own numeric value in the Priorityfield - just enter a float value in the field.			
Project	This field is derived and stores the Project where the Work Item is stored.	project	project.id (sortable)	yes
Remaining	This field is used in Polarion's project and time planning and planning accuracy metrics features. Here users specify the time remaining before the Work Item will be resolved. When a value is initially specified in the Initial Estimate field, the value of this field is automatically synchronized with it. Later, as the item is in progress, the assignee can modify the value here to that the progress on the item is reflected in the reports, metrics, and audits.  Time duration is specified by an integer value, or the literal 1/2 (or any other fraction). Examples:  3d - 3 days  7h - 7 hours	remaining Estimate (All one word)	remainingEstimate remainingEstimate.1 (sortable)	yes

Field name	Description	Field ID	Index IDs	On form
	• 1/2d - 1/2 day			
Resolution	Indicates the final outcome of a Work Item. The default values are defined in the global and/or project configuration. Keep in mind that values in projects may be from the Project Template from which the project was derived. Also keep in mind that the visibility of the field and/or the ability to change the value may be controlled by the workflow configuration and bound to change of the Status field.  You can customize the values in the relevant enumerations configuration file. See: Configure Enumerations.	resolution	resolution resolution.1 (sortable)	yes
Resolved On	This is a date field which is updated automatically when a Work Item is resolved. The value is the date the item was resolved. Not usually shown in the user interface but ID may be referenced in reports, queries, etc.	resolvedOn	resolvedOn (sortable)	no

Field name	Description	Field ID	Index IDs	On form
	The <b>Severity</b> field describes the impact (on the end user or customer) of the issue that triggered creation of the Work Item. Values in your configuration(s) should be descriptive in your context: Critical, or Must Have, for example.			
Severity	In the default configuration for this field, the IDs of Work Items with high severity appear in Red font in the user interface.	severity	severity severity.1 (sortable)	yes
	The Severity values and respective font colors can be configured in the severity-enum.xml enumeration configuration file. See:			
	Configure Enumerations.			
Status	The Status field indicates the Work Item's process phase: "New", "Resolved", "Closed", etc. Status values can be configured in the relevant enumeration configuration. See: Configure Enumerations. The status changes are driven by the workflow configuration (See:	status	status status.1 (sortable)	yes

Field name	Description	Field ID	Index IDs	On form
	Configure Work Item Workflow).			
	The field's combo box has an option Next workflow action which automatically sets the correct value for the field and triggers the next workflow actions including notifications, if any.			
	Changing the Work Item status resets the Work Item's workflow to the initial status. This affects the planning of the Work Item.			
Suspect	Field that stores a Boolean value indicating if the Work Item has an outgoing link to another Work Item with the Suspect attribute set on it or not.	suspect	suspect (sortable)	no
Time Point	Time Points are similar to milestones (versions) or development iterations. The list of Time Points can be customized to suit your situation. For information, see: Configure Time Points.	timePoint	timePoint.id timePoint.time (sortable) timePoint.earliest PlannedStart (All one word) (sortable)	yes

Field name	Description	Field ID	Index IDs	On form
Time Spent	This field is used in Polarion's project and time planning and planning accuracy metrics features. Here users specify the actual time spent to resolve the Work Item.  Time unit is specified with shorthand these characters: <b>d</b> for day(s), and <b>h</b> for hour(s).  Time duration is specified by an integer value, or the literal 1/2 (or any other fraction). Examples:  3d - 3 days  7h - 7 hours	timeSpent	timeSpent timeSpent.1 (sortable)	yes
Title	Each Work Item receives a unique ID automatically so <b>Title</b> should contain a short description of the Work Item. The item can be described further in the <b>Description</b> field.	title	title (tokenized) title.1 (sortable, lowercase)	permanent

Field name	Description	Field ID	Index IDs	On form
Туре	The <b>Type</b> of a Work Item is descriptive of its nature. For example: "Bug", "Task", "Change request". Depending on the Type of Work Item, some type-specific fields can be present or absent in the detail display. For more information, see: Configure Custom Fields.  Changing the type of a Work Item resets the Work Item's workflow to the default Status value which can affect the automated planning of the item.	type	type type.1 (sortable)	yes
Updated	The date when a Work Item was last changed. The non-editable date value is automatically assigned by Polarion.	updated	updated (date only) updated.1 (datetime) updated.2 (sortable)	permanent
WatchedBy	Lets you search for Work Items watched by a specific user.  (watchedBy:username)	watchedBy		yes

(tokenized)

## **Fields Not Imported from Excel**

Some fields cannot be imported from an Excel workbook.

- 1. Read-only, or only internally modified fields:
  - author
  - created
  - id
  - linkedWorkItemsDerived
  - linkedRevisionsDerived
  - location
  - module
  - outlineNumber
  - plannedEnd
  - plannedIn
  - plannedStart
  - previousStatus
  - project
  - updated
- 2. Other fields for which import is not supported:
  - approvals
  - attachments

- externallyLinkedWorkItems
- linkedRevisions
- planningConstraints
- timePoint
- workRecords

## **Advanced Work Item querying**

This section is for advanced users who use command-line queries in the Query Builder, wiki macros, or the API.

Polarion's query language is essentially the same as that of Apache Lucene. The syntax is documented in the Apache Lucene Syntax Documentation, available on the Apache web site.

## The following tips are relevant to Lucene queries specified in Polarion:

- Operators must be typed in upper case: AND, OR, NOT.
- White space has the same function as the OR operator, so the following queries are equivalent: severity: (blocker OR critical) OR priority: (highest OR high) will return the same result as severity: (blocker critical) priority: (highest high). White space can be also used to separate different values for one term.
- There is a special field HAS\_VALUE, which can be used to determine whether or not an object has a value set in a field or not. For example, to query for items with no value in the *resolution* field, use NOT HAS\_VALUE:resolution.
- Search terms cannot start with an asterisk (\*). For example, something\* is a valid query, but \*something is not.
- Queries, including sub-queries, can begin with the NOT operator: NOT severity:blocker, and severity:blocker AND (NOT priority:low OR HAS VALUE:assignee.id) are valid queries.
  - However queries like QUERY1 OR NOT QUERY2 are problematic and must be written in the form QUERY1 OR (NOT QUERY2).
- Operators have no defined priority, so complex queries should be surrounded by parentheses. For example, (type:userstory AND targetRelease.KEY:1.0) OR (type:defect AND priority:highest).
- Using many parenthetical expressions in queries may negatively impact performance because more time is spent in query pre-processing. To optimize the performance of your queries, consult the Apache Lucene Syntax Documentation.

The same query language is applicable in a number of administrative tasks such as customization of configuration files that contain query expressions or elements or attributes that consist of a query expression.

### **Deprecated Query Elements:**

Existing queries that use any of the constructs mentioned below still work, but are now considered deprecated and long-term support for them is not guaranteed. It is recommended that you update all such usages in saved queries, macro and page parameters in Wiki pages, API calls, etc. to the recommended form.

Work Item field CUSTOM FIELDS - Querying the HAS VALUE field is preferred to the CUSTOM FIELDS field. For example, use: HAS VALUE: targetRelease rather than CUSTOM FIELDS:targetRelease.

Also, custom fields of type enumeration now use the .KEY variant of the field. Example: targetRelease.KEY:2.1.0, and not targetRelease:2.1.0

- Query constant ######NULL For querying objects with empty field values, you should no longer use the ######NULL query constant. Use the HAS VALUE field instead. For example, to query for items with no value in Severity use NOT HAS VALUE: severity instead of severity: ######NULL.
- Activities field FIELDS The HAS VALUE field is also preferred to the FIELDS field for Activities. For example, use HAS VALUE: field instead of FIELDS: field for Activities.
- Query parameters ALL:ALL\_VALUE and ALL:NO\_VALUE Replace ALL:ALL VALUE with \*.\*, and replace ALL:NO VALUE with NOT \*.\*.

### Search syntax basics

Polarion's query engine is based on Apache Lucene. Consequently, query syntax in Polarion is essentially the same as that of Lucene. You may want to download the Lucene Documentation. The following table lists some syntax constraints that are commonly needed when constructing queries in Polarion.

System Constraint	Example
	"Word", "Phrase with several words"
Word or phrase ":	title:word
	title:"Phrase with several words"
	? – replaces single character
	* - severaltitle:ab* - finds Work Items having words starting with 'ab' in the title fieldtitle:ab*c - finds Work Items having words in the title field starting with 'ab' and finishing with 'c'
Wildcards: *, ?	The same with '?'
	Lucene doesn't allow using wildcards as first character
	Lucene expands all the wildcards to matching statements. Thus, if you type title:t*it will be expanded by Lucene to: title:the OR title:test OR title:task OR

System Constraint	Example	
	Polarion's default is 2048 clauses maximum in one Lucene query (Lucene's default is 1024). Using too generic a wildcard may exceed this maximum number resulting in an error reported back to the user.	
Fuzzy/proximity search ~	Tilde (~) is used for a fuzzy/proximity search. You should seldom need to use it.	
	[] – Specifies the range in dates or numbers,	
	{ } – Will find all Documents whose titles are between specified terms, exclusive.	
Range Searches { }, [ ]	Example:	
	created:[20120208 TO 20120222]	
	finds Work Items created between 08 Feb 2012 and 22 Feb 2012	
	{ } Rarely needed, not recommended.	
Boosting a Term ^	^ is used for relevant searching. In most scenarios you don't need it with Polarion	
	Lucene supports AND, "+", OR, NOT and "-" as Boolean operators(Note: Boolean operators must be ALL CAPS)	
	AND == && OR ==    NOT == !	
Boolean Operations + -    &&!	title:xxx AND severity:blocker finds Work Items where title contains 'xxx' and severity is set to 'blocker'. Similar syntax using OR.	
	Title:xxx AND NOT severity:blocker finds elements with 'xxx' in title and with severity other than 'blocker'	
	+/- may be used for the explicit requirement of having the record found or excluded.	
	Group statements using parentheses.	
	severity:(blocker OR critical) has the same result as severity:blocker OR severity:critical	
Grouping ()	Grouping also makes prioritization of constraints (as in mathematics):	
	severity:blocker AND (title:xxx OR title:yyy) will make the OR operation higher priority than comparing blocker severity with title:xxx.	

System Constraint	Example
Escaping with backslash (\)	When using any characters above in text search, use '\'. For example, title.1:\(1+1\)\:2 will search Work Items with the literal value '(1+1):2' in the "title" field.

#### **Tokenization**

Polarion supports two different tokenization algorithms. System property search.wordBoundaries can be used to select from them.

### White space based:

This is the default tokenization. However, it is not convenient for languages that do not use white space between words. The indexed text is split around white spaces, and from the remaining character sequences all leading and trailing non-alphanumeric characters (anything that is not a letter or number) are removed.

#### Note

For advanced users: This splitting is done using a regular expression and <code>java.lang.String.split(String)</code>. It is possible to configure a custom regular expression that should be used via the system property <code>search.wordBoundaries.splitByPattern</code>.

### **Examples:**

(Term for title.1 field is always the same as the title itself.)

Item Title	Terms for "title" field	Query	Result (standard tokenization)
Mary had a little 'lamb'.	Mary, had, little, lamb	title:mary	found
		title.1:mary	not found (title.1's term is "Mary had a little lamb")
		title:mary*	found
		title.1:mary*	found
SW_ngcb_simulati	oßW_ngcb_simulation	title:SW	not found
		title:ngcb	not found
		title:SW_ngcb	not found
		title:SW_ngcb*	found
		title.1:SW_ngcb	not found (title.1's term is "SW_ngcb_simulation")

Item Title	Terms for "title" field	Query	Result (standard tokenization)
		title.1:SW_ngcb*	found
WI-1234	WI-1234	title:WI	not found
		title:1234	not found
		title:WI-1234	found

### **Standard Tokenization:**

This method splits text into words according to the Unicode standard. To use this method of tokenization, use the system property search.wordBoundaries=standard.

# **Examples:**

(Term for title.1 field is always the same as the title itself.)

Item Title	Terms for "title" field	Query	Result (standard tokenization)
Mary had a little lamb.	Mary, had, little, lamb	title:mary	found
		title.1:mary	not found (title.1's term is "Mary had a little lamb")
		title:mary*	found
		title.1:mary*	found
SW_ngcb_simulati	oßW_ngcb_simulation	title:SW	not found
		title:ngcb	not found
		title:SW_ngcb	not found
		title:SW_ngcb*	found
		title.1:SW_ngcb	not found (title.1's term is "SW_ngcb_simulation")
		title.1:SW_ngcb*	found
WI-1234	WI-1234	title:WI	found
		title:1234	found
		title:WI-1234	found (matches titles containing "WI" or "1234")

### Combining text with visual queries

If you have used the visual elements of the Query Builder to create a guery, you can optionally append free-form textual query syntax. When combining visual and textual query elements, there are a few points to keep in mind:

- The free-form part of query is usually connected syntactically with the visual elements by the AND operator.
  - For example, if you have two visual elements [Type:Requirement] [Assignee:Me] (visual elements denoted here by square brackets), if you append text severity: must have, your Query Builder line will look like [Type:Requirement] [Assignee:Me] + severity:must\_have, and will be treated by the **query parser as:** type:requirement AND assignee.id:\$me AND severity:must have.
- Appending text to visual elements with the OR operator causes the leading visual elements to be treated syntactically as if they were surrounded by parentheses.
  - For example, if you have two visual elements [aa:xx] [bb:yy] (visual elements denoted here by square brackets), if you append text OR CC:ZZ, your Query Builder line will look like [aa:xx] [bb:yy] + OR cc:zz, and will be treated by the query parser as: (aa:xx AND bb:yy) OR cc:zz.
- If the Query Builder is cleared of all visual elements, you can start the free-form part of a query with an operator. However, AND and OR are ignored. Operator NOT at the beginning of a query is parsed as you would expect.

### Edit an opened query:

You may sometimes access the Work Items table via a link in an email or a shortcut containing a query, via a saved guery. If the guery you are opening was originally constructed with visual guery elements, the visual elements appear in the Query Builder when you access the table via the link or shortcut, and you can subsequently edit the query by modifying the visual elements.

If the query you open contains any elements that cannot be parsed into visual elements, the elements which can be rendered visually are so rendered until an element that cannot be parsed into a visual element is encountered. The remaining elements are rendered as a text string. You can then modify the query any way you want, editing, adding or removing visual elements, and/or editing, removing, or modifying the text string.

#### Queries using special index functions

Polarion implements a number of special index functions and "query expander" variables, which can be referenced to simplify queries on complex objects... Documents or Test Records, for example. This section describes the main use cases and related special index functions.

#### Current-date based searches:

The Polarion query system provides a special \$today\$ constant that can be used in query strings to retrieve Work Items updated on the current date. This can be useful for creating shortcuts.

### Allowed formats:

\$today\$: actual date

Modifiers can be used to shift the date to a specified time frame:

- \$today SHIFT\$: actual date minus time frame specified by SHIFT (see examples below)
- \$today + SHIFT\$: actual date plus time frame specified by SHIFT

The SHIFT parameter can take the following form:

- Nd: N days, where N is an integer value. For example, 14d represents a time frame of 14 days
- Nw: N weeks
- Nm: N months
- Ny: N years

The following example searches for items created during the previous week:

```
created:[$today - 2w$ TO $today - 1w$]
```

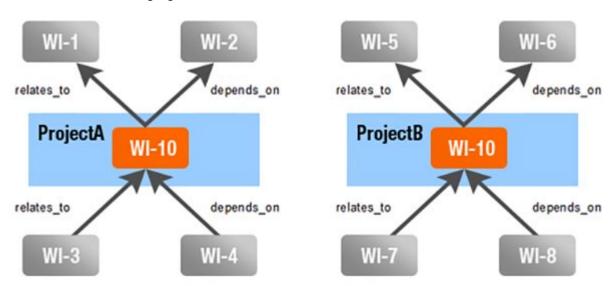
Spaces between \$today and SHIFT\$ are ignored, so expressions like \$today-3w\$ will work.

### **Query for Linked Work items:**

There can sometimes be a need to query for Work Items that are linked by another Work Item according to the link role. You can formulate a query around the following elements:

- linkedWorkItems
- backlinkedWorkItems

Consider the following figure:



There are two Work Items having an ID of WI-10. These exist in different projects, and both have links to/from other Work Items (for the purposes of this example, it doesn't matter in which projects the linked items live).

Query	Matched Items
backlinkedWorkItems:WI-10	WI-1, WI-2, WI-5, WI-6
backlinkedWorkItems:ProjectA/WI-10	WI-1, WI-2
backlinkedWorkItems:depends_on=WI-10	WI-2, WI-6
backlinkedWorkItems:depends_on=ProjectA/WI-10	WI-2

Query	Matched Items
linkedWorkItems:WI-10	WI-3, WI-4, WI-7, WI-8
linkedWorkItems:ProjectA/WI-10	WI-3, WI-4
linkedWorkItems:depends_on=WI-10	WI-4, WI-8
linkedWorkItems:depends_on=ProjectA/WI-10	WI-4

## **Query for Test Record Work Items:**

You can build queries to search for Work Items linked to a test record in a particular Test Run using the  ${\tt TEST\_RECORDS}$  element. This can be particularly useful when creating report pages. For example, a test manager might use it to query for what defects were part of some test results.

# The element can take the following parameters:

Component	Description	Require	d Attributes
test run	Identifier of the Test Run whose test records should be queried for linked Work Items. Format: project ID/Test Run ID.	YES	N/A
test result	The kind of results to search for.	ON	testResultId - identifier of a kind of result, for example, "failed", "passed", "blocked"  @any - search for test records with any type of result  @null - search for test records with no result  * - search for all test records
executed by	Search for test records executed by some specified user(s)	NO	userId - the user ID of a user who executed the tests of the test record  * - search for all test records

Component	Description	Require	d Attributes
when executed	Date or time interval when execution of the Test Run of a test record occurred	NO	yyyymmdd - search for test records executed on particular day  yyyymmdd TO yyyymmdd - search for test records executed in time interval  * - search for all test records
linked work item type	Type of Work Item linked to test records	NO	false (default value) - search for Test Cases linked to test records  true - search for Defects linked to test records

#### **Examples:**

Search for test cases executed in Initial System Verification Test:

TEST RECORDS: ("drivepilot/Initial System Verification Test", @any)

Search for test cases planned for Initial System Verification Test, but not executed yet:

TEST RECORDS: ("drivepilot/Initial System Verification Test", @null)

Search for test cases executed by mTest in last week in Full System Verification Test:

TEST\_RECORDS:("drivepilot/Full System Verification Test",@any,"mTest","\$today-1w\$ TO \$today\$")

Search for defects triggered by test failures in Initial System Verification Test:

TEST RECORDS: ("drivepilot/Initial System Verification Test", "failed", \*, \*, "true")

### **Query for Documents:**

Use the *document.id* index field to query for a specific Document. You can use the <code>outlineNumber</code> index field to query for items contained in Documents according to their outline number.

Syntax: Syntax: document.id: "Space Name/Document Name"

**Examples:** document.id: "Functional Requirements" searches for the named Document in the default space. Note that the \_default space name should not be specified. You only need to specify it in the query parameter of wiki macros such as the {workitems} macro.

Example: document.id: "Requirements/Functional Specification" searches for a Document named Functional Specification in the Requirements space.

You can query for **Outline numbers** in Documents using the special index field <code>outlineNumber</code>. The Outline Number field can be sorted on this index field. Sort is by document.outlineNumber, and wildcards are supported. Examples:

All items in current scope with outline number on level 1.1: outlineNumber:1.1.\*

Unresolved items with outline number on level 1.1 in Document named *FunctionalSpec* located in the *Specification* space: NOT HAS\_VALUE:resolution AND document.id:Specification/Functional Specification AND outlineNumber:1.1\*

When searching for content of a particular outline number, the document to search within can also be specified.

```
outlineNumber:("PROJECT/SPACE/ID", "2*")
```

When working within the Document scope (Using the filter in the document editor, the table or tree tabs), the document is automatically added to the query.

### Tip

Specify a document whenever possible (in all places other than in the Document scope), otherwise the outline query will search through all the documents in the entire repository.

#### **Note**

"OutlineNumber" queries only work when the number of Work Items that contain matching outline numbers is not greater than the value of the "luceneMaxClauseCount" Polarion Property. (The default is set to 20 000.)

## **Query for Approvals:**

Querying for items in a particular state with regard to approval has a special syntax - see the following examples.

Items awaiting approval by a specific user (user ID "rProject"): approvals:fullxxxapprovedxxxrProject

All items disapproved by anyone: approvals:fullxxxdisapprovedxxx\*

All items awaiting approval by the current user: approvals:fullxxxwaitingxxx\$[user.id]

#### Tip

In the Work Items table, create a query using the visual guery builder, selecting the Approval field and setting the desired parameters. In the helper panel, click Copy to **Clipboard** to access the query including the special syntax.

# **Mathematical formula syntax**

You can create or edit formulas in Polarion with LaTeX syntax.

You can find extensive resources on the MathJax library for LaTex syntax at the following external sites:

- https://docs.mathjax.org/en/latest/tex.html#supported-latex-commands
- http://www.onemathematicalcat.org/MathJaxDocumentation/TeXSyntax.htm

## **Caution**

The external links may contain some commands from versions of MathJax that are not supported in Polarion.

(See the list of supported symbols below.)

# Tip

You can use third-party software or an online editor to create a formula and copy and paste the LaTeX syntax they output into Polarion.

# **Examples**

Formula	LaTeX Syntax
$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
$F(x)=\int_b^a\frac{1}{3}x^3$	$F(x) = \int_{a_b} \frac{1}{3}x^3$
$ x  = \begin{cases} x & \text{if } x \ge 0 \\ -x & \text{if } x < 0 \end{cases}$	x  = \begin{cases} x & \text{ if } x\ge 0 \\ -x & \text{ if } x \lt 0 \end{cases}
$\left(\sum_{k=1}^n a_k b_k\right)^2 \leq \left(\sum_{k=1}^n a_k^2\right) \left(\sum_{k=1}^n b_k^2\right)$	\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
$A = egin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \ a_{21} & a_{22} & \dots & a_{2n} \ dots & dots & \ddots & dots \ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$	A = \pmatrix{     a_{11} & a_{12} & \ldots & a_{1n} \cr     a_{21} & a_{22} & \ldots & a_{2n} \cr     \vdots & \vdots & \vdots \cr     a_{m1} & a_{m2} & \ldots & a_{mn} \cr }

Formula	LaTeX Syntax
$\sin A \cos B = \frac{1}{2} [\sin(A - B) + \sin(A + B)]$ $\sin A \sin B = \frac{1}{2} [\sin(A - B) - \cos(A + B)]$ $\cos A \cos B = \frac{1}{2} [\cos(A - B) + \cos(A + B)]$	\sin A \cos B = \frac{1}{2}\left[ \sin(A-B)+\sin(A+B) \right] \\ \sin A \sin B = \frac{1}{2}\left[ \sin(A-B)-\cos(A+B) \right] \\ \cos A \cos B = \frac{1}{2}\left[ \cos(A-B)+\cos(A+B) \right]
$\frac{d}{dx} \left( \int_0^x f(u)  du \right) = f(x)$	\frac{d}{dx}\left(\\int_{0}^{x} f(u)\du\\right)=f(x)
$\frac{d}{dx}\arctan(\sin(x^2)) = -2\frac{\cos(x^2)x}{-2 + (\cos(x^2))^2}$	{\frac {d}{dx}}\arctan(\sin({x}^{2}))=-2{\frac {\cos({x}^{2})x} {-2+\left (\cos({x}^{2})\right )^{2}}}
$egin{cases} \lim_{x o\infty} \exp(-x) = 0 \ k_{n+1} = n^2 + k_n^2 - k_{n-1} \ a mod b, x \equiv a \pmod b \ rac{n!}{k!(n-k)!} = inom{n}{k} \end{cases}$	\begin{cases} \lim\limits_{x \to \infty} \exp(-x) = 0\\ k_{n+1} = n^2 + k_n^2 - k_{n-1}\\ a \bmod b,x \equiv a \pmod{b}\\ \frac{n!}{k!(n-k)!} = \binom{n}{k}\\ \end{cases}
$\frac{(x_1x_2)}{\times (x_1'x_2')} \frac{(y_1y_2y_3y_4)}{(y_1y_2y_3y_4)}$	\frac{ \begin{array}[b]{r} \left( x_1 x_2 \right)\\ \times \left( x'_1 x'_2 \right) \end{array} }{ \left( y_1y_2y_3y_4 \right) }
$\int\limits_a^b P\left(A=2\Big rac{A^2}{B}>4 ight)>\int_0^\infty \mathrm{e}^{-x}\mathrm{d}x$	\int\limits_a^bP\left(A=2\middle \frac{A^2} {B}>4\right)>\int_0^\infty \mathrm{e}^{-x}\mathrm{d}x
$\left\{\frac{x^2}{y^3}\right\} \to \frac{x^3}{3}\bigg _0^1 \to \left(\left(\left(\frac{\mathrm{d}}{\mathrm{d}x}(kg(x))\right)\right)\right)$	\left\{\frac{x^2}{y^3}\right\}\to \left.\frac{x^2}{3}\right _0^1\to \big( \Big( \Bigg( \Bigg( \frac{\mathrm d}{\mathrm d x} \left( k g(x) \right) \Bigg) \bigg) \Big) \big)

Formula	LaTeX Syntax
$M = egin{bmatrix} rac{5}{6} & rac{1}{6} & 0 \ rac{5}{6} & 0 & rac{1}{6} \ 0 & rac{5}{6} & rac{1}{6} \end{bmatrix}$	M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\[0.3em] \frac{5}{6} & 0 & \frac{1}{6} \\[0.3em] 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}
$\binom{n}{r} = \frac{n!}{r!(n-r)!}$	\left( \begin{array}{c} n \\ r \end{array} \right) = \frac{n!}{r!(n-r)!}
$egin{bmatrix} xz & xw \ yz & yw \end{bmatrix} = egin{bmatrix} x \ y \end{bmatrix}  imes [z & w]$	\begin{bmatrix} xz & xw \\ yz & yw \end{bmatrix} = \left[ \begin{array}{c} x \\ y \end{array} \right] \times \left[ \begin{array}{cc} z & w \end{array} \right]
$f(x)=\int_{-\infty}^{\infty}\hat{f}\left( \xi ight) ,e^{2\pi i\xi x}\ d\xi$	$ f(x) = \int_{-\infty}^{\int_{-\infty}^{\infty}} f(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x) = \int_{-\infty}^{\infty} f(x) = \int_{-\infty}^{\infty} f(x) dx $
$\left\{egin{aligned} B' &= - abla  imes E, \ E' &=  abla  imes B - 4\pi j, \ F &: X  ightarrow Y \ x  ightarrow 2x \end{aligned} ight.$	\begin{cases} B'=-\nabla \times E,\\ E'=\nabla \times B - 4\pi j,\\ F\colon X \rightarrow Y \\ x \mapsto 2x \end{cases}

# Symbols

Command	Output	How it's used	Note
#			
~	a b	a~b	non-breaking space
1	f'(x)	f'(x)	
%	a+b	a + b % comment	single-line comment

Command	Output	How it's used	Note
&	$egin{array}{cccccccccccccccccccccccccccccccccccc$	\begin{matrix} -1 & 2 \\ 3 & -4\end{matrix}	column separation
٨	$a^b$	a^b	superscript
_	$a_b$	a_b	subscript
{	$\frac{1}{2}$	\frac{1}{2}	
}	$\frac{1}{3}$	\frac{1}{3}	
1	$ig  \ a \ b$	a\ b	
\!	$a^2/b$	a^2\!/b	negative thin space
\#	#		
\\$	\$		
\%	%		
\&	&		
١,	a b	ab	thin space
<i>\</i> :	a b	a\:b	medium space
\;	a b	a\;b	thick space
\>	a b	a\>b	alternate medium space
			line separator
\_	_		
\{	{		
И			
\}	}		

## Α

Command	Output	How it's used	Note
\above	$\frac{a}{b}$	a \above 1pt b	
\abovewithdelims	$\left[\frac{a}{b}\right]$	a \abovewithdelims [] 1pt b	
\acute	$\dot{a}$	\acute a	
\aleph	х		
\alpha	α		
\amalg	П		
∖And	&		
\angle	_		
\approx	$\approx$		
\approxeq	$\approx$		
\arccos	arccos		
\arcsin	arcsin		
\arctan	arctan		
\arg	arg		
\array	$egin{array}{ccc} a & b & & & & & & & & & & & & & & & & &$	a & b \cr c & d }	
\Arrowvert			
\arrowvert	1		
\ast	*		
\asymp	$\asymp$		
\atop	$egin{array}{c} a \ b \end{array}$	a \atop b	

Command	Output	How it's used	Note
\atopwithdelims	$\begin{bmatrix} a \\ b \end{bmatrix}$	a \atopwithdelims [ ] b	

## В

Command	Output	How it's used	Note
\backepsilon	Э		
\backprime	1		
\backsim	~		
\backsimeq	∽		
\backslash	\		
\bar	$\bar{a}$	\bar{a}	
\barwedge	$\overline{\wedge}$		
\Bbb	Q	\Bbb Q	
\Bbbk	k		
\bbox	math	\bbox[10px]{math}	
\bcancel	X2.	\bcancel{X^2}	
\because	:		
\begin	$egin{array}{cccc} -1 & 3 \ 2 & -4 \end{array}$	\begin{matrix} -1 & 3 \\ 2 & -4\end{matrix}	
\begingroup			
\beta	β		
\beth	٦		
\between	Ŏ		

Command	Output	How it's used	Note
\bf			
\Big	(x)	\Big(x\Big)	
\big	(x)	\big(x\big)	
\bigcap	Λ		
\bigcirc	0		
\bigcup	U		
\Bigg	$\bigcap a$	\Bigg {\uparrow}a	
\bigg	$\bigcap a$	\bigg {\uparrow}a	
\BiggI			
\biggI			
\Biggm			
\biggm			
\Biggr			
\biggr			
\BigI	(x)	\Bigl(x\Bigr)	
\bigI	(x)	\bigl(x\bigr)	
\Bigm			
\bigm			
\bigodot	0		
\bigoplus	$\oplus$		

Command	Output	How it's used	Note
\bigotimes	$\otimes$		
\Bigr	(x)	\Bigl(x\Bigr)	
\bigr	(x)	\bigl(x\bigr)	
\bigsqcup	П		
\bigstar	*		
\bigtriangledown	$\nabla$		
\bigtriangleup	Δ		
\biguplus	+		
\bigvee	V		
\bigwedge	٨		
\binom	$\binom{a}{b}$	\binom a b	
\blacklozenge	<b>•</b>		
\blacksquare			
\blacktriangle	<b>A</b>		
\blacktriangledown	▼		
\blacktriangleleft	<b>◄</b>		
\blacktriangleright	•		
\bmod	mod		
\boldsymbol	$oldsymbol{x}$	\boldsymbol{x}	
\bot	Τ		
\bowtie	$\bowtie$		

Command	Output	How it's used	Note
\Box			
\boxdot			
\boxed	x	\boxed{x}	
\boxminus			
\boxplus	⊞		
\boxtimes			
\brace	{}		
\bracevert	1		
\brack			
\breve	$reve{x}$	\breve{x}	
\buildrel	def =	\buildrel \rm {def} \over {=}	
\bullet	•		
\Bumpeq	≎		
\bumpeq	<u></u>		

# С

Command	Output	How it's used	Note
\cal			
\cancel	7	\cancel{=}	
\cancelto	text some	\cancelto{some}{te	xt}
\cap	n		
\Cap			

Command	Output	How it's used	Note
\cases	$\left\{egin{array}{l} x^2+2y+6=\ x^2-7y+13=\ \end{array} ight.$	1 x^2+2y+6=1 \cr x^2-7y+13=-9 }	
\cdot	•		
\cdotp	•		
\cdots	• •		
\ce	x	\ce{x}	
\cee	x	\cee{x}	
\centerdot	•		
\cf	x	\cf{x}	
\cfrac	$oxed{rac{x}{y}}$	\cfrac{x}{y}	
\check	Ň	\check{X}	
\checkmark	✓		
\chi	$\chi$		
\choose	()		
\circ	0		
\circeq	<u>•</u>		
\circlearrowleft	Q		
\circlearrowright	Ò		
\circledast	*		
\circledcirc	0		
\circleddash	Θ		
\circledR	®		

Command	Output	How it's used	Note
\circledS	<u>s</u>		
\class			
\clubsuit	*		
\colon	:		
\color	red text	\color{red}{\text{red}} text}}	
\colorbox	σ	\require{color}\colo	rbox{yellow}{\$\sigma\$}
\complement	C		
\cong	$\cong$		
\coprod	П		
\cos	cos		
\cosh	cosh		
\cot	cot		
\coth	coth		
\cr	$\left\{egin{array}{l} x^2-1=y \ x^2-7y-3= \end{array} ight.$	x^2-1=y ()cr x^2-7y-3=0 }	
\csc	csc		
\cssId			
\cup	U		
\Cup	U		
\curlyeqprec	4		
\curlyeqsucc	$\nearrow$		
\curlyvee	Υ		

Command	Output	How it's used	Note
\curlywedge	人		
\curvearrowleft			
\curvearrowright	>		

# D

Command	Output	How it's used	Note
\dagger	†		
\daleth	٦		
\dashleftarrow	<b>←</b>		
\dashrightarrow	<b></b> →		
\dashv	-		
\dbinom	$\binom{2}{3}$	\dbinom{2}{3}	
\ddagger	‡		
\ddddot	·ä·	\ddddot a	
\dddot	ä	\dddot a	
\ddot	$\ddot{a}$	\ddot a	
\ddots	·		
\DeclareMathOper	at $\int_a^b$	\DeclareMathOper \dec_a^b	ator{\dec}{\int}
\definecolor	text	\require{color} \definecolor{Bright 1.0, 0.0} BrightGreen	
\def			
\deg	deg		

Command	Output	How it's used	Note
\Delta	Δ		
\delta	δ		
\det	det		
\dfrac			
\diagdown	\		
\diagup	/		
\diamond	<b>♦</b>		
\Diamond	$\Diamond$		
\diamondsuit	<b>♦</b>		
\digamma	F		
\dim	dim		
\displaylines	a+b=0 $a=1$	\displaylines{ a + b = 0 \cr a = 1 }	
\displaystyle			
\div	•		
\divideontimes	*		
\dot	$\dot{a}$	\dot a	
\doteq	÷		
\Doteq	÷		
\doteqdot	<b>÷</b>		
\dotplus	÷		
\dots	•••		
\dotsb	•••		

Command	Output	How it's used	Note
\dotsc	•••		
\dotsi	•••		
\dotsm	•••		
\dotso	•••		
\doublebarwedge	₹		
\doublecap			
\doublecup	U		
\Downarrow	₩		
\downarrow	<b>↓</b>		
\downdownarrows	$\downarrow \downarrow$		
\downharpoonleft	1		
\downharpoonright			

# Ε

Command	Output	How it's used	Note
\ell	$\ell$		
\emptyset	Ø		
\enclose			
\end			
\endgroup			
\enspace			
\epsilon	$\epsilon$		
\eqalign			
\eqalignno			
\eqcirc	<del></del>		

Command	Output	How it's used	Note
\eqref			
\eqsim	~		
\eqslantgtr	>		
\eqslantless	<		
\equiv	=		
\eta	$\eta$		
\eth	ð		
\exists	3		
\exp	exp		

# F

Command	Output	How it's used	Note
\fallingdotseq	<b>=</b>		
\fbox			
\fcolorbox			
\Finv	4		
\flat	b		
\forall	$\forall$		
\frac			
\frac			
\frak			
\frown			

# G

Command	Output	How it's used	Note
\Game	Э		

Command	Output	How it's used	Note
\Gamma	Γ		
\gamma	γ		
\gcd	$\operatorname{gcd}$		
\gdef			
\ge	2		
\genfrac			
\geq	<u>&gt;</u>		
\geqq	$\geq$		
\geqslant	>		
\gets	<del></del>		
/gg	>>		
/ggg	<b>&gt;&gt;&gt;</b>		
\gggtr	<b>&gt;&gt;&gt;</b>		
\gimel	I		
\global			
\gnapprox	<del>≈</del>		
\gneq	<i>&gt;</i>		
\gneqq	≩		
\gnsim	<i>≥</i>		
\grave			
\gt	>		
\gtrapprox	\approx		

Command	Output	How it's used	Note
\gtrdot	>		
\gtreqless	>		
\gtreqqless	<b>&gt;</b>    <b>&gt;</b>		
\gtrless	≷		
\gtrsim	≳		
\gvertneqq	<b>\Rightarrow</b>		

## Н

Command	Output	How it's used	Note
\hat	$\hat{a}$	\hat a	
\hbar	$\hbar$		
\hbox			
\hdashline			
\heartsuit	$\Diamond$		
\hline			
\hom	hom		
\hookleftarrow	$\leftarrow$		
\hookrightarrow	$\hookrightarrow$		
\hphantom			
\href			
\hskip			
\hslash	ħ		
\hspace			
\Huge			

Command	Output	How it's used	Note
\huge			
\idotsint	$\int \cdots \int$		

I

Command	Output	How it's used	Note
\iff	$\iff$		
\iiiint	<b>IIII</b>		
\iiint	<b>SSS</b>		
\iint	<i>ff</i>		
\lm	I		
\imath	ı		
\impliedby	←		
\implies	$\Rightarrow$		
\in	$\in$		
\inf	inf		
\infty	$\infty$		
\injlim	inj lim		
\int	ſ		
\intercal	Т		
\intop	ſ		
\iota	L		
\it			

#### J

Command	Output	How it's used	Note
\jmath	J		
\Join	$\bowtie$		

## Κ

Command	Output	How it's used	Note
\kappa	$\kappa$		
\ker	ker		
\kern			

#### L

Command	Output	How it's used	Note
\label			
\Lambda	Λ		
\lambda	λ		
\land	$\land$		
\langle	<		
\LARGE			
\Large			
\large			
\LaTeX	$ \mathcal{L}T_{E}X $		
\lbrace	{		
\lbrack	[		
\lceil	Γ		
\ldotp	•		

Command	Output	How it's used	Note
\ldots	•••		
\le	<u> </u>		
\leadsto	~ <del>&gt;</del>		
\left			
\Leftarrow	←		
\leftarrow	<del></del>		
\leftarrowtail	←<		
\leftharpoondown	_		
\leftharpoonup			
\leftleftarrows	⊭		
\Leftrightarrow	$\Leftrightarrow$		
\leftrightarrow	$\leftrightarrow$		
\leftrightarrows	$\stackrel{\longleftarrow}{\longrightarrow}$		
\leftrightharpoons	<del></del>		
\leftrightsquigarrow	, <del>~~</del>		
\leftroot			
\leftthreetimes	$\lambda$		
\leq	<u> </u>		
\leqalignno			
\leqq	$\leq$		
\leqslant	<		
lessapprox	×≈		
\lessdot	<		
\lesseqgtr	<u> </u>		

Command	Output	How it's used	Note
\lesseqqgtr	<b>\left\{   \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \</b>		
\lessgtr	\$		
\lesssim	$\lesssim$		
\let			
\lfloor	L		
\lg	lg		
\lgroup	(		
\lhd	⊲		
\lim	lim		
\liminf	lim inf		
\limits			
\limsup	$\limsup$		
\II	«		
\llap			
\llcorner	L		
\Lleftarrow	<b></b>		
\III	<b>«</b>		
\Illess	<b></b>		
\lmoustache	ſ		
∖ln	ln		
\lnapprox	≨		
\lneq	<i>\$</i>		

Command	Output	How it's used	Note
\lneqq	≨		
\Inot	¬		
\Insim	<del></del>		
\log	log		
\Longleftarrow	<b>(</b>		
\longleftarrow	<del></del>		
\Longleftrightarrow	$\iff$		
\longleftrightarrow	$\longleftrightarrow$		
longmapsto	$\longmapsto$		
\Longrightarrow	$\Rightarrow$		
\longrightarrow	$\longrightarrow$		
\looparrowleft	↔		
\looparrowright	↔		
\lor	V		
lower			
\lozenge	$\Diamond$		
\lrcorner	_		
\Lsh	↰		
\lt	<		
\Itimes	K		
\IVert			
\lvert			
\lvertneqq	≨ #		

# M

Command	Output	How it's used	Note
\maltese	*		
\mapsto	$\mapsto$		
\mathbb			
\mathbf			
\mathbin			
\mathcal			
\mathchoice			
\mathclose			
\mathfrak			
\mathinner			
\mathit			
\mathop			
\mathopen			
\mathord			
\mathpunct			
\mathrel			
\mathring			
\mathrm			
\mathscr			
\mathsf			
\mathstrut			
\mathtip			
\mathtt			
\matrix	$egin{array}{ccc} a & b & & & & & & & & & & & & & & & & &$	a & b \cr c & d }	

Command	Output	How it's used	Note
\max	max		
\mbox			
\measuredangle	∠		
\mho	υ		
\mid			
\middle			
\min	min		
\mit			
\mkern			
\mmlToken			
\mod			
\models	F		
\moveleft			
\moveright			
\mp	干		
\mskip			
\mspace			
\mu	$\mu$		
\multimap	<b>─</b> ○		

# N

Command	Output	How it's used	Note
\nabla	$\nabla$		
\natural	Ц		
\ncong	<b>#</b>		

Command	Output	How it's used	Note
\ne	<i>≠</i>		
\nearrow	7		
\neg	一		
\negmedspace			
\negthickspace			
\negthinspace			
\neq	<i>≠</i>		
\newcommand			
\newenvironment			
\Newextarrow			
\newline			
\nexists	A		
\ngeq	≱		
\ngeqq	≱		
\ngeqslant	*		
\ngtr	*		
\ni	€		
\nLeftarrow	#		
\nleftarrow	<del><!---</del--></del>		
\nLeftrightarrow	<b>⇔</b>		
\nleftrightarrow	↔		
\nleq	≰		

Command	Output	How it's used	Note
\nleqq	≢		
\nleqslant	≰		
\nless	*		
\nmid	1		
\nobreakspace			
\nolimits			
\normalsize			
\not	/		
\notag			
\notin	⊭		
\nparallel	#		
\nprec	*		
\npreceq	≠		
\nRightarrow	<i>⇒</i>		
\nrightarrow	<i>→</i> >		
\nshortmid	ł		
\nshortparallel	Ħ		
\nsim	~		
\nsubseteq	⊈		
\nsubseteqq	≨		
\nsucc	7		

Command	Output	How it's used	Note
\nsucceq	≠		
\nsupseteq	⊉		
\nsupseteqq	⊉		
\ntriangleleft	A		
\ntrianglelefteq	<b>⊉</b>		
\ntriangleright	×		
\ntrianglerighteq	⊭		
\nu	ν		
\nVDash	¥		
\nVdash	<b></b>		
\nvDash	¥		
\nvdash	⊬		
\nwarrow			

# 0

Command	Output	How it's used	Note
\odot	$\odot$		
\oint	∮		
\oldstyle			
\Omega	Ω		
\omega	ω		
\omicron	o		

Command	Output	How it's used	Note
\ominus	$\Theta$		
\operatorname			
\oplus	$\oplus$		
\oslash	$\oslash$		
\otimes	$\otimes$		
\over	-		
\overbrace	$\overbrace{a+\cdots+a}^{n \text{ times}}$	a+\cdot times}}	s+a}^{n
\overleftarrow	$\stackrel{\longleftarrow}{ab}$	\overleftarrow {ab}	
\overleftrightarrow	$\stackrel{\longleftrightarrow}{ab}$	\overleftrightarrow {ab}	
\overline	$oxed{ab}$	\overline {ab}	
\overrightarrow	$\overrightarrow{ab}$	\overrightarrow {ab}	
\overset	$egin{array}{c} a \ b \end{array}$	\overset {a} {b}	
\overwithdelims			
\owns	∋		

# Ρ

Command	Output	How it's used	Note
\parallel			
\partial	$\partial$		
\perp	上		
\phantom			

Command	Output	How it's used	Note
\Phi	Φ		
\phi	$\phi$		
\Pi	П		
\pi	$\pi$		
\pitchfork	ф		
\pm	土		
\pmatrix	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	a & b \cr c & d}	
\pmb			
\pmod			
\pod			
\Pr	Pr		
\prec	$\prec$		
\precapprox	<b>∀</b> ≋		
\preccurlyeq	$\preceq$		
\preceq	$\preceq$		
\precnapprox	<b>₹</b>		
\precneqq	¥		
\precnsim	ズ		
\precsim	$\stackrel{\sim}{\sim}$		
\prime	1		
\prod	П		

Command	Output	How it's used	Note
\projlim	proj lim		
\propto	$\propto$		
\Psi	Ψ		
\psi	$\psi$		

# Q

Command	Output	How it's used	Note
\qquad			

# R

Command	Output	How it's used	Note
\raise			
\rangle	>		
\rbrace	}		
\rbrack	]		
\rceil	]		
∖Re	R		
\ref			
\renewcommand			
\renewenvironmen	t		
\require			
\restriction	1		
\rfloor			

Command	Output	How it's used	Note
\rgroup	)		
\rhd	$\triangleright$		
\rho	ρ		
\right			
\Rightarrow	$\Rightarrow$		
\rightarrow	$\rightarrow$		
\rightarrowtail	$\rightarrow$		
\rightharpoondown	$\overline{}$		
\rightharpoonup			
\rightleftarrows	ightharpoonup		
\rightleftharpoons	$\rightleftharpoons$		
\rightleftharpoons	ightharpoonup		
\rightrightarrows	$\Rightarrow$		
\rightsquigarrow	~ <del>&gt;</del>		
\rightthreetimes	<i>X</i>		
\risingdotseq	<b>≓</b>		
\rlap			
\rm			
\rmoustache	Ì		
\root	$\sqrt[a]{b}$	\root a \of b	
\Rrightarrow	⇒		
∖Rsh	1		
\rtimes	×		

Command	Output	How it's used	Note
\Rule			
\rVert			
\rvert			

# S

Command	Output	How it's used	Note
\S			
\scr			
\scriptscriptstyle			
\scriptsize			
\scriptstyle			
\searrow	>		
\sec	sec		
\setminus	\		
\sf			
\sharp	#		
\shortmid	1		
\shortparallel	П		
\shoveleft			
\shoveright			
\sideset			
\Sigma	Σ		
\sigma	$\sigma$		
\sim	~		

Command	Output	How it's used	Note
\simeq	~		
\sin	sin		
\sinh	sinh		
\skew			
\small			
\smallfrown	$\sim$		
\smallint	ſ		
\smallsetminus	\		
\smallsmile	$\smile$		
\smash			
\smile	$\sim$		
\Space			
\space			
\spadesuit	•		
\sphericalangle	∢		
\sqcap	П		
\sqcup	Ц		
\sqrt			
\sqsubset			
\sqsubseteq	⊑		
\sqsupset			
\sqsupseteq	⊒		
\square			
\stackrel			

Command	Output	How it's used	Note
\star	*		
\strut			
\style			
\subset	C		
\Subset	€		
\subseteq	$\subseteq$		
\subseteqq	$\subseteq$		
\subsetneq	Ç		
\subsetneqq	⊊ ≠		
\substack			
\succ	>		
\succapprox	₩		
\succcurlyeq	≽		
\succeq	<u></u>		
\succnapprox	<b></b>		
\succneqq	<del>\</del> ≠		
\succnsim	<b>≻</b> ∻		
\succsim	<i>≿</i> ?		
\sum	Σ		
\sup	sup		
\supset	$\supset$		
\Supset	∋		

Command	Output	How it's used	Note
\supseteq	$\supseteq$		
\supseteqq	⊇		
\supsetneq	⊋		
\supsetneqq	⊋		
\surd	$\checkmark$		
\swarrow	✓		

# Т

Command	Output	How it's used	Note
\tag			
\tan	tan		
\tanh	tanh		
\tau	τ		
\tbinom			
\TeX	$T_{EX}$		
\text			
\textbf			
\textit			
\textrm			
\textsf			
\textstyle			
\texttt			
\texttip			
\tfrac			

Command	Output	How it's used	Note
\therefore	<i>:</i> .		
\Theta	Θ		
\theta	$\theta$		
\thickapprox	2		
\thicksim	~		
\thinspace			
\tilde	$ ilde{a}$	\tilde a	
\times	×		
\tiny			
\Tiny			
\to	$\rightarrow$		
\toggle			
\top	Т		
\triangle	Δ		
\triangledown	$\nabla$		
\triangleleft	◁		
\trianglelefteq	$\triangleleft$		
\triangleq	<u></u>		
\triangleright	$\triangleright$		
\trianglerighteq			
\tt			
\twoheadleftarrow	<b>«</b> —		
\twoheadrightarrow	/ →		

# U

Command	Output	How it's used	Note
\ulcorner	Г		
\underbrace	$\underbrace{a+\cdots+a}_{n \text{ times}}$	a+\cd times}}	ots+a}_{n
\underleftarrow			
\underleftrightarrov	v		
\underline			
\underrightarrow			
\underset			
\unicode			
\unlhd	⊴		
\unrhd	⊵		
\Uparrow	介		
\uparrow	<b>↑</b>		
\Updownarrow	<b>\$</b>		
\updownarrow	<b>‡</b>		
\upharpoonleft	1		
\upharpoonright	1		
\uplus	₩		
\uproot			
\Upsilon	Υ		
\upsilon	v		
\upuparrows	$\uparrow\uparrow$		
\urcorner	٦		

# ٧

Command	Output	How it's used	Note
\varDelta	Δ		
\varepsilon	ε		
\varGamma	Γ		
\varinjlim	$\stackrel{\lim}{\longrightarrow}$		
\varkappa	х		
\varLambda	Λ		
\varliminf	lim		
\varlimsup	$\overline{\lim}$		
\varnothing	Ø		
\varOmega	Ω		
\varphi	$\varphi$		
\varPhi	Φ		
\varpi	$\varpi$		
\varPi	П		
\varprojlim	lim ←		
\varpropto	$\propto$		
\varPsi	$\Psi$		
\varrho	ρ		
\varsigma	ς		
\varSigma	Σ		
\varsubsetneq	Ş		

Command	Output	How it's used	Note
\varsubsetneqq	≨		
\varsupsetneq	⊋		
\varsupsetneqq	⊋		
\vartheta	$\vartheta$		
\varTheta	Θ		
\vartriangle	Δ		
\vartriangleleft	◁		
\vartriangleright	$\triangleright$		
\varUpsilon	$\gamma$		
\varXi	Ξ		
\vcenter			
\vdash	<b></b>		
\Vdash	⊩		
\vDash			
\vdots	:		
\vec			
\vee	V		
\veebar	<u> </u>		
\verb			
\Vert			
\vert			
\vphantom			

Command	Output	How it's used	Note
\Vvdash	≡		

# W

Command	Output	How it's used	Note
\wedge	$\land$		
\widehat			
\widetilde			
\wp	В		
\wr	}		

# X

Command	Output	How it's used	Note
١Xi	[1]		
\xi	ξ		
\xcancel			
\xleftarrow			
\xlongequal			
\xmapsto			
\xrightarrow			
\xtofrom			
\xtwoheadleftarrow			
\xtwoheadrightarro	w		

# Υ

Command	Output	How it's used	Note
\yen	¥		

# Z

Command	Output	How it's used	Note
\zeta	ζ		

# Chapter 6: Polarion SDK

# ALM REQ QA

The Polarion SDK (Software Development Kit) contains documentation of available APIs and the embedded database schema, plus libraries, source code, and code examples for developers who want to develop custom extensions to Polarion ALM's standard features and functionality.

The SDK content is installed to the file system along with all other programs and data. You need to explore the file system of your installation in order to access it. You may wish to copy the SDK from the installed location to developers' local computers, or host it on a file server on your network.

#### **Default SDK location**

You can find the Polarion SDK content in the <code>\$POLARION\_HOME\$/polarion/sdk</code> folder of your installation's file system. Documentation is located in the doc sub-folder. (On Windows, substitute backslash for forward slash.)

The <code>\$POLARION\_HOME\$/polarion/sdk/index.html</code> file provides links to all SDK resources. You can access the SDK index page via your web browser by appending <code>polarion/sdk/</code> to the URL of your Polarion server.

# **Example:**

http://polarion.mydomain.com/polarion/sdk/

# **Chapter 7: Glossary**

#### **Definitions:**

## **\$POLARION\_HOME\$**

Denotes the root of the Polarion installation in documentation. For Windows systems, the default value is C:\Polarion. For Linux systems it is the location of the binaries, usually <code>/opt/polarion/</code>.

## Administration

The area of Polarion used by administrators for managing the system configuration, customization, user accounts, and so forth. Access requires administrator permissions on the user's account. Permission may be granted globally for the repository, or for one or more specific projects. When a user account has the required permission for the scope, the **Administration** link appears in the Polarion header enabling the user to access administration topics and features.

#### **Artifact**

A piece of information that 1) is produced, modified or used by a process, 2) defines an area of responsibility, and 3) is subject to version control. An artifact can be a model, a model element, or a piece of information that is used or produced by a software (or application) development process.

In Polarion documentation, the term can apply to Work Items. It is generally applied to units of blocks of information in an external office document which become Work Items when the document is imported into Polarion.

#### **Baseline**

A work product (such as software or documentation) that has been formally reviewed, approved, and delivered and can only be changed through formal change control procedures.

In Polarion, a Baseline record for a project or repository created in the **Baselines** topic.

## **Bi-directional Traceability**

Achieved when the traceability chains can be demonstrated in both directions, ensuring that the end product meets all of the requirements, and only the requirements specified.

In Polarion, the possibility to trace from an artifact/Work Item up to a linking parent item or down to a linked item.

#### **Build Artifact**

The smallest buildable entity. Comprised of resources - usually source code and support files.

Builds of projects are configured in Administration in the **Builds** topic.

## Category

A standard data field of Work Items for the purpose of categorizing them. When Work Items are assigned one or more Category values, they can be queried by Category. For example, if category User Interface is defined, Work Items relating to user interface development can be assigned that category. It is then simple to query for Work Items having a Category value matching that "User Interface".

Categories can be defined for each project in **Administration \*\* \*\* Work Items \*\* Categories**.

#### Classic Wiki

See: Wiki Page.

#### **Content Pane**

Content Pane is a part of user interface, where you can read project information (wiki pages or Work Items, for example) and edit content.

#### **Dashboard**

A special wiki-based page that rolls up and displays key status information based on actual progress data automatically maintained by Polarion. Information can be viewed for an entire repository, a group of projects, or a single project.

(Not all Polarion products have this feature.)

#### **Document**

Document (capitalized) in documentation refers to a Live Document.

#### **Enumeration**

A list of values that users can select from and apply to some Work Item field. For example, Work Item types are defined in an enumeration (Administration > Work Items > Enumerations).

#### **Initial Estimate**

A standard data field of Work Items, it represents the amount of time initially estimated for completing a Work Item. Values in this field are used by the Live Plan planning engine when generating a project plan.

See also Remaining Estimate.

## Info Page

A type of Page that does not contain user-configurable report Widgets which retrieve display project and Work Item data. The term is simply to distinguish such Pages from LiveReport Pages that do contain Widgets. Technologically, there is no difference. A Page created as an Info Page can contain Widgets.

#### Information Scope

Polarion enables viewing and management of information for an entire repository, for a group of projects, or a single project. The information scope refers to the breadth of information you are

currently working with. The Repository scope means just that: you are working with or looking at information from all projects in a repository. For example, the Dashboard topic in this scope (in products which have the feature) rolls up status information from all projects to provide a view on the overall status of development across all projects managed with Polarion.

#### Interface

A named version of the Polarion user interface that displays some subset of all available information to facilitate some role or work context, for example, "Manager" or "Developer". An Interface can show or hide Navigation topics, invoke some particular view on Work Items, show or hide Work Item fields, and control the layout of Work Item fields.

Interfaces are configured in the **Interfaces** topic of **Administration**.

#### Job

A predefined system process that may be initiated either by system or by user. Examples of jobs include backups and builds. Administrators can define jobs and job scheduling.

#### Link Role

A descriptive word or words applied to a link between two Work Items that describes the relationship between those items. For example, a link role "depends on" may be defined and applied to a link between two Work Items when one of the linked items has some dependency upon the other one.

#### Link Role Rule

A control that limits the application of link roles between Work Items of one or more specified types. For example, if a link role should be applicable to links between Work Items of type A and B, but not to links between Work Items of type A and C, a link role rule can be defined that enforces this. When this link role rule exists, users are prevented from specifying such a link.

#### LiveDoc

An artifact-aware online document that can contain Work Items of one or more types. LiveDocs, (referred to in documentation as "Documents") are ideal for creating specifications for such things as requirements or test cases. Work Items defined in a LiveDoc can be managed in the same way as those defined directly in the integrated tracker (that is the Work Items topic in Navigation).

Prior to version 2011, the term referred to Microsoft Office Word and Excel documents, based on special document templates, which could define and store Polarion Work Items. While useful for some, this approach had drawbacks, especially in terms of scalability. Beginning with version 2011, Polarion refactored the technology completely. Users having projects containing pre-2011 Microsoft Word and Microsoft Excel documents (created with Polarion version 2010 or earlier) are recommended to migrate to the current LiveDocs implementation. Legacy documents are no longer indexed and consequently do not appear in query results. If you have legacy documents and need to have them included in the index, please contact Polarion ALM technical support.

"LiveDoc" may also refer to Polarion's artifact-aware online document technology.

#### **Live Document**

See LiveDoc above.

#### Live Plan

The feature of some Polarion products which automatically calculates a project plan based on a number of input parameters including project start and end dates, Time Points and Work Items assigned to them, estimated time and time worked on Work Items, human capacity based on working calendars, and many other factors.

The term can refer not only to the Live Plan engine, but also the GANTT-style chart of the project plan rendered by the engine in the Live Plan view of the Work Items topic, and in the Live Plan section of the Dashboard topic (in products that provide these features).

#### LiveReport

A type of Page that contain user-configurable report Widgets which retrieve display project and Work Item data. The term is simply to distinguish such Pages from others that do not contain Widgets. Technologically, there is no difference.

#### Module

A Module was a special container for Work Items in Polarion versions prior to version 2011. It is now deprecated in favor of Polarion LiveDoc Documents.

## **Navigation Pane**

Navigation pane is the part of user interface where a user is able to select available topics the present features, functionality, and information.

#### **Orphaned Work Item**

An orphaned Work Item is a Work Item which was originally defined in a LiveDoc and later unmarked or merged with another Work Item in the Document. The orphaned Work Item no longer appears as such in the document, but the artifact still exists in the underlying repository. It is therefore subject to workflow control and all other management features. The Recycle Bin pane of the Document Sidebar provides tools for managing orphaned Work Items.

## Page

When capitalized, refers to a portal web page that contains information that is not tracked and workflow-managed through a process.

Content of Pages cannot be marked and managed as Work Items. However, Pages may display Work Item data via special report Widgets. Pages can have hyperlinks to other internal and external URLs. Pages have a palette of rich formatting tools, and can display images and contain file attachments.

#### **Process**

A set of repeatable steps with a clearly defined beginning and end.

In Polarion, a process is encapsulated in a project's workflow configuration and managed by setting different values in the Status field of Work Items as the item tracks through the process steps.

#### **Project**

An undertaking requiring effort that is constrained by limited resources, and is planned, executed, and controlled.

In Polarion, the term can also refer to a folder in a Polarion repository that stores and manages all artifacts relating to some specific development activity or goal. A project's folder is marked as a project by Polarion when an administrator creates a new project.

## **Remaining Estimate**

A standard data field of Work Items, it represents the difference between the amount of time initially estimated for completing a Work Item, and the amount of time already reported spent working on it. Values in this field are used by the Live Plan planning engine when generating a project plan.

See also: Initial Estimate.

## Repository

Polarion's underlying data storage and history mechanism. Currently, the open source version control system Subversion is used for this purpose. All development artifacts including Work Items, wiki pages and documents are stored and versioned in the repository.

By default, a single Subversion repository is created when Polarion is installed. With appropriate licensing and IT infrastructure, Polarion can support multiple repositories

## Requirement

A specific business result that is needed to solve a problem or achieve an objective.

In Polarion, the term often refers to an artifact or Work Item type which defines a requirement as per the definition above.

# **Saved Query**

A query string either manually entered or built by the Query Builder which is saved for later reuse. Saved queries appear in the drop-down panel of the Query Builder in Work Item views in the Work Items topic. Saved queries can be shared globally for all platform users, or for all users of a specific project. Individual users can also create "private" saved queries which are available only to them.

Saved Queries differ from *Shortcuts* in that they only retrieve Work Items and are available only in the Work Items page.

#### Scope

See Information Scope.

#### **Shortcut**

A special link in Navigation that takes the user to specified saved subset and presentation of information. Shortcuts can be shared globally for all platform users, or for all users of a specific project. Individual users can also create "private" shortcuts which are only available to them.

Shortcuts differ from *Saved Queries* in that they are not limited to retrieving Work Items, and they can retrieve presentation as well as data.

## SVN

An abbreviation for Subversion, a leading open source version control system used by Polarion to store all artifacts of the application development process.

#### **Test Case**

A type of Work Item which defines a set of steps for testing some defined functionality or feature of an application. Polarion project templates which provide QA/testing support pre-define a Work Item type named "Test Case". (Project administrators may change the actual name of this type, or create some other type which is analogous to it.)

Test Case items may created and defined either in a Document or directly in the integrated tracker (Navigation - Www.ltems).

#### **Test Run**

A business object which encompasses a set of Test Cases (which usually define manual tests) or unit tests (normally automated tests) to be executed, together with information about the status of the Test Run, results of execution of the Test Run and the Test Cases or unit tests which it encompasses, the test environment, and the build tested.

#### Time Point

A named milestone or development deadline associated with a date. Time Points are most commonly created for projects by a project manager/administrator. Work Items can be assigned to a Time Point by project managers or other users with the necessary permissions. When assigned to a Time Point, Work Items are processed by the Live Plan engine and shown in the project plan (Live Plan view of the Work Items topic and the Live Plan section of the Dashboard topic).

#### User

A person or computer-based process that interacts with Polarion. An "end user" is a Polarion user who does not have administrator permissions.

#### Variant

For a glossary of terms related to variants and variant management, see Terminology.

#### View

A named representation of a set of Work Items in which the retrieved items are presented in some specific format or context. For example, the Table view renders Work Items as table, the Tree view renders Work Items as a hierarchical tree of linked items, the Matrix view renders two sets of Work Items in a matrix format. The Road Map view renders Work Items associated with a defined point in time.

#### Widget

A user-configurable visual report component that can be inserted into LiveReport Pages and which retrieve and visually format project and Work Item data in useful ways in real time.

# Wiki Page

DEPRECATED. A unitary online document that uses Polarion's initial wiki technology from prior to version 2015. Now referred to in documentation as "Classic Wiki page".

Content of Wiki Pages cannot be marked and managed as Work Items. However, Wiki Pages may display Work Item data from the integrated tracker by means of macros which contains query

language that retrieves some set of Work Items and dynamically displays it in a Wiki Page. Pages can have hyperlinks to other Pages and/or external URLs, and/or to tracker pages. Pages can display images and contain file attachments.

#### Work Item

Polarion's term for a tracked and managed system artifact that represents some issue to be processed, or element of work to be done. Work Items are a key element in the workflow of a project.

# Work Item Import Rule

A rule, defined (and optionally saved for reuse) when importing a Microsoft Office Word document into Polarion as a LiveDoc, which causes Polarion to recognize content in the source document as a Work Item.

# **Work Item Type**

Work Items have a property *type* which describes the basic nature of the issue or work. Examples of common Work Item types include:

- Defect
- Change Request
- Requirement
- Task

Administrators can create custom Work Item types to support any process, semantics, and workflow.

## Workflow

The feature of Polarion that encapsulates a process and subsequently controls and automates the progress, and reflects the status of Work Items from their inception until they are resolved/closed.

Polarion products come with predefined workflows suitable for many types projects. Administrators can customize workflow to support any process or methodology used by an organization to develop applications.

#### **Workflow Action**

An action invoked by a user or a job that changes the status of a Work Item. For example, suppose a project has statuses Open and Accepted defined for Work Items. Suppose also that it has a workflow action Accept defined. When set by a user, the Accept action triggers transition of the Work Item status from Open to Accepted.

#### **Workflow Transition**

An artifact defined for Work Item workflows which transitions them from one defined status to another (from *Open* to *Accepted*, for example), resulting in notifications, history entry, etc.

## **Work Record**

A multivalued data field for Work Items. It enables the assignee(s) of a Work Item to record multiple blocks of time spent working on the item. The aggregated time from all work records is added to any value in the Time Spent field, and that value is used to update the Remaining Estimate field. Using Work Records is optional.

# **Chapter 8: About Polarion**

**Polarion AG** is a Siemens Company. Since Polarion's inception in 2004, our mission has been to help companies advance the development, governance and maintenance of software via a unified solution for Requirements, Quality, and Application Lifecycle Management. Polarion Software was acquired by Siemens PLM Software in January 2016, expanding their support for the growing ALM market. The acquisition is the culmination of a relationship between Siemens and Polarion that started several years ago through which Siemens has been an investor, a partner and customer of Polarion.

## Here are some links to the product web site which may be of interest:

- Product Home Page: https://polarion.plm.automation.siemens.com
- Extensions Portal: http://extensions.polarion.com
- Product Downloads: https://polarion.plm.automation.siemens.com/downloads

#### COPYRIGHT STATEMENT

This software and related documentation are proprietary to Polarion AG (a Siemens company).

© 2019 Polarion AG

## **Trademarks and Acknowledgments**

Polarion is a registered trademark of **Polarion AG**. **Polarion ALM**, **Polarion Requirements**, **Polarion QA**, **Polarion Pro**, **Polarion Reviewer**, and **Polarion Variants** are trademarks of **Polarion AG**.

Siemens and the Siemens logo are registered trademarks of Siemens AG. NX, Solid Edge, and Teamcenter are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or their subsidiaries in the United States and in other countries.

All other trademarks, registered trademarks, or service marks belong to their respective holders.

Polarion software products use icon images from the Silk Icons set, <a href="http://www.famfamfam.com/lab/icons/silk/">http://www.famfamfam.com/lab/icons/silk/</a> and the Copenhagen icon set, <a href="https://www.iconfinder.com/">https://www.iconfinder.com/</a>.

Polarion software products may use licensed icon images from the Glyphish PRO icon set.

This software and related documentation are proprietary to Siemens Product Lifecycle Management Software Inc. © 2019 Polarion AG. Polarion is a registered trademark of Polarion AG. Polarion ALM, Polarion REQUIREMENTS, Polarion QA

and Polarion VARIANTS are trademarks or registered trademarks of Polarion AG.

registered trademarks, or service marks belong to their respective holders.

Siemens and the Siemens logo are registered trademarks of Siemens AG. NX, Solid Edge, and Teamcenter are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or their subsidiaries in the United States and in other countries. All other trademarks,