Teamcenter 11.5 lifecycle visualization

ClearanceDB Administration Guide
Contents

Getting started .............................................. 1-1
  ClearanceDB overview ..................................... 1-1
  System requirements ...................................... 1-2
  Supported platforms ...................................... 1-2
  Version compatibility .................................... 1-3
  Supported locales ....................................... 1-3
  ClearanceDB system components .......................... 1-4
  Steps to install, configure, and run ClearanceDB analysis .................................. 1-6
  ClearanceDB issue management process ................ 1-7
  ClearanceDB managed workflow ......................... 1-8
  Perl and SQL script reference ........................... 1-9
  Other resources .......................................... 1-12

Installing ClearanceDB components .......................... 2-1
  Install ClearanceDB components on Windows ........... 2-1
  Install ClearanceDB components on UNIX ................ 2-2
  Install a ClearanceDB patch ............................... 2-4
  Upgrade an existing database ............................. 2-4
  Upgrade to Oracle 11gR2 .................................. 2-5

Configuring ClearanceDB Server ............................. 3-1
  ClearanceDB Server overview ............................ 3-1
  Create the ClearanceDB database ....................... 3-2
  Tune the database memory parameters ................. 3-9
  Creating the ClearanceDB tablespaces .................. 3-11

Creating user accounts ....................................... 4-1
  Creating user accounts .................................. 4-1
  Create Oracle users ...................................... 4-2
  Update Oracle user accounts ............................. 4-4
  Understanding roles ...................................... 4-5
  Create ClearanceDB auxillary users .................... 4-7

Creating the ClearanceDB Work Area ......................... 5-1
  Setting up the work area ................................ 5-1
  Create the ClearanceDB Work Area ...................... 5-2
  Global configuration options ............................ 5-3
  Configure the global DBC file ........................... 5-19
  Configure the global .vvi file ........................... 5-20

Configuring and running ClearanceDB Proxy ................ 6-1
ClearanceDB Proxy overview .............................................. 6-1
Install the Oracle Instant Client (Windows) .......................... 6-2
Install the Oracle Instant Client (UNIX and Linux) ............... 6-2
Configure the default ClearanceDB Proxy settings ............... 6-4
Encrypt the Oracle user password .................................... 6-5
Create a ClearanceDB Proxy ........................................... 6-6
Configure a ClearanceDB Proxy ....................................... 6-6
Run a ClearanceDB Proxy as a Windows service ................. 6-8
Start a ClearanceDB Proxy ............................................ 6-10
Ping a ClearanceDB Proxy ............................................ 6-11
Ping the database ....................................................... 6-11
List products in the database ......................................... 6-12
Stop a ClearanceDB Proxy ............................................. 6-12

Setting up products for analysis ...................................... 7-1
Setting up products for analysis ....................................... 7-1
Configuring the ClearanceDB default product ................... 7-2
  Understanding the ClearanceDB default product ................ 7-2
  Create the ClearanceDB default product ......................... 7-2
Create a new ClearanceDB product .................................. 7-4
Configuring the ALL product for issue inheritance ............. 7-7
  Set up the ALL product for issue inheritance ................ 7-7
  Define groups for the ALL product ............................... 7-7
  Example configuration file for the ALL product ............... 7-8
Deploying the issue inheritance mechanism ....................... 7-8
  Enable the issue inheritance mechanism ....................... 7-8
  Disable the issue inheritance mechanism ..................... 7-9
  Modifying the issue inheritance mechanism .................. 7-9
Using the product-specific configuration file ...................... 7-9
ClearanceDB product configuration attributes ................... 7-11
  Overview of ClearanceDB product attributes ................. 7-11
  ClearanceDB product attributes reference ................... 7-12
  Merging product attributes with the database ............... 7-25
  Example configuration file ....................................... 7-25
Specifying rules and conditions .................................... 7-26
  Specifying rules and conditions ............................... 7-26
  ClearanceDB rules ................................................ 7-26
  ClearanceDB conditions .......................................... 7-28
  Merging rules and conditions with the database ............. 7-31
  Examples of rules and conditions .............................. 7-31
Defining clearance zones ............................................. 7-50
  Defining clearance zones ........................................ 7-50
  Understanding clearance zones .................................. 7-51
  Merging zones with the database ............................... 7-51
  Example zones file ............................................... 7-52

Performing analysis .................................................... 8-1
Accuracy of clearance calculations .................................. 8-1
Enabling rules and conditions ....................................... 8-2
Chapter 1: Getting started

ClearanceDB overview

ClearanceDB enables you to analyze and manage clearance issues according to criteria specified by your organization. It provides a flexible mechanism to control analysis, utilizing rules and conditions that you create based upon part metadata. Focusing analysis on specific parts minimizes extraneous results, which in turn helps to optimize the clearance analysis process, especially when working with large sets of data. ClearanceDB integrates with an Oracle database, providing a powerful, secure environment for the clearance management processes.

ClearanceDB can help you to detect, resolve, and manage spatial interference issues that arise as your product design evolves throughout its lifecycle. By automating the clearance analysis process, you can identify design flaws that would otherwise delay the release of new product offerings, increase development costs, and jeopardize quality.

ClearanceDB's rules-based capabilities are especially appropriate for determining how changing design decisions impact pairs of parts in your product design. ClearanceDB is so effective because it enables you to establish a part pairs database that defines which specific parts should be checked against each other on a repeated basis.
**System requirements**

**Perl requirements**
The ClearanceDB client and proxy tiers require Perl version 5.03 or later. If Perl is not already installed on your system or you have a version earlier than 5.03, many sources are available for you to manually install the latest distribution, such as from http://www.activestate.com.

**ClearanceDB Server requirements**
ClearanceDB Server is supported on Oracle 11g R2 (version 11.2.0.3) and 12c Enterprise and Standard Editions.

**ClearanceDB Proxy requirements**
The ClearanceDB proxy tier requires the Oracle Instant Client Basic (not Basic Lite) package, version 11.2.0.3.

**ClearanceDB Client SQL script requirements**
The ClearanceDB client tier requires Oracle Net Services, sqlplus, and sqlldr, to run the ClearanceDB SQL scripts.

**Clearance Calculator Client requirements**
For Clearance Calculator, which is part of the client tier, a minimum of 16 GB RAM and 4 CPU cores is recommended. Performance increases with additional RAM and CPU resources. For high end installations, it is not unusual to have 128 GB of RAM and 32 cores (or more).

**Supported platforms**
You can run ClearanceDB on the platforms listed below in addition to the standard Lifecycle Visualization platforms, except Mac. For more information about system hardware and software requirements, see the hardware and software certifications page on GTAC.

**Note**
The list of supported platforms is applicable to all ClearanceDB components, including Clearance Calculator, ClearanceDB Client, ClearanceDB Proxy, and ClearanceDB Server.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Chipset</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSE Linux Enterprise Server</td>
<td>11 SP2 and 12</td>
<td>x86-64</td>
</tr>
</tbody>
</table>

**Note**
Includes both physical and virtual machines.
<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Chipset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Solaris</td>
<td>10 and 11</td>
<td>Sun Ultra</td>
</tr>
</tbody>
</table>
| Windows Server 64-bit | Windows Server 2008 R2 SP1 Standard and Enterprise  
 |                   | Windows Server 2012 R2 Standard and Enterprise  
 |                   | Windows Server 2016        | x86-64 (both Intel and AMD) |

For more information about supported virtual machines, please contact your Siemens PLM Software GTAC representative.

**Version compatibility**

**Version compatibility with Teamcenter**

ClearanceDB support for Teamcenter versions is consistent with Lifecycle Visualization support. Please see the Teamcenter lifecycle visualization interoperability page for details.

**Version compatibility within ClearanceDB tiers**

- **Client** (Mockup, Clearance.exe, ClearanceDbClient.exe) and **Proxy** (ClearanceDbProxyServer.exe, ClearanceDbProxyClient.exe)— Must all come from the same exact Lifecycle Visualization version.
- **Server** — Must be the same or later version than the client and proxy within the same major release (for example, 10.x). In some cases, a server that is a higher major version than the client (for example, an 11.1 server and a 10.1 client) may work, but specific cases need to be confirmed through your Siemens PLM Software GTAC representative.

**Supported locales**

In general, ClearanceDB results can be loaded and managed using localized Teamcenter and Teamcenter lifecycle visualization in any 8-bit locale. However, the ClearanceDB application is only *partially localized*. Strings from the ClearanceDB database (some of which are visible in Teamcenter and Teamcenter lifecycle visualization) are not localized. Perl scripts are also not localized.

- All files under the ClearanceDB working directory, such as .cfg files, .csvclodb files, and product names, must be entered in a single byte locale.
- Any user data entered into the ClearanceDB database, such as product names, issue dispositions, rules, and conditions, must be in a single byte locale.
- Clearance DB items in Teamcenter and Teamcenter lifecycle visualization are localized, except text from the ClearanceDB server. For example, this includes ClearanceDB Reports column names, Oracle errors, and ClearanceDB server errors. These items always appear in English.
ClearanceDB system components

The basic ClearanceDB architecture is shown below, along with descriptions of each component's role in the ClearanceDB process.
Note

- You can install each of the ClearanceDB software components on a separate machine, or group any combination of components on a single machine or multiple machines. All of these components also can be installed on a single machine for testing purposes.

- In general, the ClearanceDB software components running on each tier must be from the same major release, with the exception of the client and proxy components, which must be from exactly the same release. Additionally, within a major release, the server component must be at least as recent as the client and proxy components. For example, an installation where the server component is from the 8.1 release and the client and proxy components are from the 8.0 release is supported; an installation where the server component is from a different major release than the client and proxy components, or where the client and proxy components are not from the exact same release, is not supported.

Client tier

The Client tier consists of ClearanceDB applications such as the Clearance Calculator, Mockup or the Teamcenter Rich Client with the RDV option, and the ClearanceDbClient.

- Clearance Calculator - The Clearance Calculator accesses product information from the database in order to perform rules-based clearance analysis. A command line application, the Calculator is typically run on a predetermined schedule using a batch file or script. The Calculator generates a results file, which is merged with the database via the ClearanceDB Client or SQL*Plus on a machine with an Oracle Client installation.

- Mockup or the Teamcenter Rich Client - Mockup or a supported Teamcenter application such as DesignContext loads the ClearanceDB analysis results associated with a particular product for the purpose of managing relevant issues within a visual environment. End users can review ClearanceDB issues and update their status within the database.

- ClearanceDB Client - The ClearanceDB Client communicates with the proxy tier components, enabling ClearanceDB administrators to upload product configurations, rules and conditions, zones, and results to the database.

Proxy tier

The proxy tier consists of the ClearanceDB Proxy and the Oracle Instant Client.

- ClearanceDB Proxy - This component consists of two applications, the ClearanceDbProxyServer and the ClearanceDbProxyClient. Only the ClearanceDbProxyServer needs to be started by the administrator; the ClearanceDbProxyClient is invoked automatically by the ClearanceDbProxyServer. The ClearanceDB Proxy facilitates communication between the ClearanceDB client tier and the Oracle database.

- Oracle Instant Client - The Oracle Instant Client handles communication between the proxy and the database.
Server tier

The server tier is an Oracle Database server with a ClearanceDB database instance.

- ClearanceDB Server - This component consists of an Oracle Database installation populated with appropriate user information, product configurations, rules and conditions, zones, results, and issue dispositions.

Steps to install, configure, and run ClearanceDB analysis

1. Setup the ClearanceDB system components. This is a series of tasks which includes:
   - Installing the ClearanceDB Server, ClearanceDB Proxy, ClearanceDB Client, and Clearance Calculator. These components can be installed independently from one another on different machines or together in any combination on a single machine.
   - Installing and configuring additional system components that make up the ClearanceDB server tier. This includes configuring the Oracle Server, creating tablespaces on the server for ClearanceDB data, and creating ClearanceDB user accounts.
   - Configuring the system components that make up the ClearanceDB proxy tier. This includes creating the ClearanceDB Work Area, and configuring and starting one or more instances of the proxy.
   - Configuring the system components that make up the ClearanceDB client tier. This includes configuring the ClearanceDB Client and the Clearance Calculator. It may also include installing the Mockup client and configuring it to access and display ClearanceDB results.

2. Create products for analysis. This entails the following:
   - Creating a default product. The default product is used in the creation of new products.
   - Creating product configurations for your models and uploading them to the database.
   - Creating rules, conditions, and zones to focus the analysis according to your unique criteria.

3. Setup and run the analysis. This process includes the following:
   - Configuring the batch mode analysis.
   - Running the Clearance Calculator against your product data to obtain results.
   - Uploading the results to the ClearanceDB database.
ClearanceDB issue management process

1. Define context
2. Check clearance
3. Review concerns
4. Submit issues
5. Modify design
6. Validate and close issues
ClearanceDB managed workflow

1. Author/Edit data in CAD system
2. Evaluate results in viewer and create issues
3. Check clearance
4. Store results in ClearanceDB database
5. Store data in Teamcenter
6. Create rules to reduce false positive clearance results
# Perl and SQL script reference

Use the following scripts to manage the ClearanceDB database, the proxy, and your products.

**Note**
The ClearanceDB SQL scripts are case-sensitive.

## Database management scripts

<table>
<thead>
<tr>
<th>Use this script</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>cre_aux_ClearanceDB_user.sql</td>
<td>Create auxiliary ClearanceDB users.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> sqlplus system/manager@connection @cre_aux_ClearanceDB_user logfile user password dfft_tablespace temp_tablespace owner</td>
</tr>
<tr>
<td>cre_ClearanceDB_user.sql</td>
<td>Create ClearanceDB users.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> sqlplus system/manager@connection @cre_ClearanceDB_user logfile user password dfft_tablespace temp_tablespace</td>
</tr>
<tr>
<td>del_aux_ClearanceDB_user.sql</td>
<td>Delete auxiliary ClearanceDB users.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> sqlplus system/manager@connection @del_aux_ClearanceDB_user logfile user</td>
</tr>
<tr>
<td>upd_aux_ClearanceDB_user.sql</td>
<td>Update auxiliary ClearanceDB users.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> sqlplus user/passwd@connection @upd_aux_ClearanceDB_user logfile auxuser action role</td>
</tr>
<tr>
<td>upd_ClearanceDB_user.sql</td>
<td>Update ClearanceDB users.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> sqlplus user/passwd@connection @upd_ClearanceDB_user logfile dfft_tablespace indx_tablespace</td>
</tr>
</tbody>
</table>
### Proxy management scripts

<table>
<thead>
<tr>
<th>Use this script</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_proxy.pl</td>
<td>Create a new ClearanceDB Proxy.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> create_proxy.pl <code>&lt;proxy name&gt;</code></td>
</tr>
<tr>
<td>ping_database.pl</td>
<td>Test database connectivity.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> ping_database.pl <code>[&lt;proxy host name&gt; &lt;port&gt; &lt;connect data&gt;]</code></td>
</tr>
<tr>
<td>ping_proxy.pl</td>
<td>Test to determine if the proxy is running.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> ping_proxy.pl <code>[&lt;proxy host name&gt; &lt;port&gt;]</code></td>
</tr>
<tr>
<td>start_proxy.pl</td>
<td>Start a ClearanceDB Proxy.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> start_proxy.pl <code>&lt;proxy configuration file&gt;</code></td>
</tr>
<tr>
<td>stop_proxy.pl</td>
<td>Stop a ClearanceDB Proxy.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> stop_proxy.pl <code>&lt;proxy configuration file&gt;</code></td>
</tr>
</tbody>
</table>

### Product management scripts

<table>
<thead>
<tr>
<th>Use this script</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>analyze.managed_product.pl</td>
<td>Perform analysis on a Teamcenter managed product.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> analyze.managed_product.pl <code>&lt;product&gt;</code></td>
</tr>
<tr>
<td>analyze_product.pl</td>
<td>Perform analysis on a product.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> analyze_product.pl <code>&lt;product&gt;</code></td>
</tr>
<tr>
<td>copy_product.pl</td>
<td>Copy an existing ClearanceDB product and use it to create a new product in both your work area and the database.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> copy_product.pl <code>&lt;product name&gt; &lt;new product name&gt;</code></td>
</tr>
</tbody>
</table>
### Use this script

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_product.pl</td>
<td>Create a new product.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>create_product.pl &lt;name&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-uc:-ur:-ud:-uz:-ua] [-ds &lt;datasource&gt; -cd &lt;connection name&gt;]</td>
</tr>
<tr>
<td>delete_product.pl</td>
<td>Delete a ClearanceDB product from both your work area and the database.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>delete_product.pl &lt;product name&gt;</td>
</tr>
<tr>
<td>list_all_products.pl</td>
<td>List all of the ClearanceDB products in the database.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>list_all_products.pl [&lt;proxy host name&gt; &lt;port&gt; &lt;connect data&gt;]</td>
</tr>
<tr>
<td>list_product.pl</td>
<td>Display information about a ClearanceDB product.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>list_product.pl &lt;product&gt;</td>
</tr>
<tr>
<td>rename_product.pl</td>
<td>Rename a ClearanceDB product in both your work area and the database.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rename_product.pl &lt;product name&gt; &lt;new product name&gt;</td>
</tr>
</tbody>
</table>

### General administration scripts

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_ClearanceDB_working_dir.pl</td>
<td>Create the ClearanceDB Work Area.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>create_ClearanceDB_working_dir.pl &lt;path&gt; [-df]</td>
</tr>
<tr>
<td>create_default_product.pl</td>
<td>Create the default product.</td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>create_default_product.pl &lt;path&gt;</td>
</tr>
</tbody>
</table>
Use this script | To do this
--- | ---
update_product.pl | Upload product information to the database.

Usage:
update_product.pl [product]
-uc:-ur:-ud:-uz:-ua [-df]

Other resources

For detailed information on installing and licensing Mockup, refer to Teamcenter lifecycle visualization Installation Guide.

For documentation on working with clearance results in stand-alone Mockup or the Teamcenter Lifecycle Viewer, refer to Customizing the display of clearance results within Teamcenter lifecycle visualization help.

For documentation on working with clearance results in DesignContext, refer to Performing clearance analysis and proximity filtering in the DesignContext Guide.

For help on working with Oracle software, refer to the extensive documentation resources located at http://docs.oracle.com/en/database/. Oracle provides a wealth of material, including getting started guides and tutorials.
Chapter 2: Installing ClearanceDB components

Install ClearanceDB components on Windows

1. Locate the Teamcenter lifecycle visualization installation image.

2. Run setup.exe.

3. In the Teamcenter Visualization - InstallShield Wizard dialog box, click Next.

4. Click the + icon next to ClearanceDB.

5. Click the icon next to each ClearanceDB component that you want to install and choose This feature will be installed on local hard drive.
   You can choose to install any combination of the following components on a single machine:
   • Clearance Calculator
   • ClearanceDB Proxy
   • ClearanceDB Client
   • ClearanceDB Server

   Tip
   You can also install ClearanceDB example data by navigating to Help and Miscellaneous Features→Example Files→ClearanceDB Example Files.

6. To change the destination, click Change, browse to a new location, and click OK.

7. Click Next.

8. In Language Selection, click Next.

9. In License Type, do one of the following:

<table>
<thead>
<tr>
<th>Choose this option</th>
<th>Then do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install a node-locked license from this file</td>
<td>Enter the file path, or click Browse to locate the appropriate file and then click Open.</td>
</tr>
<tr>
<td>Choose this option</td>
<td>Then do this</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Obtain a license from a license server machine</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• If you are setting up a single license server, enter the Server Name and Server Port. The license.dat files in the license folder of the client will be updated.</td>
</tr>
<tr>
<td></td>
<td>• If you are setting up a redundant license server:</td>
</tr>
<tr>
<td></td>
<td>a. Enter the server name and port numbers in the Server Name box:</td>
</tr>
<tr>
<td></td>
<td>port@host1;port@host2;port@host3</td>
</tr>
<tr>
<td></td>
<td>b. Leave Server Port blank.</td>
</tr>
<tr>
<td></td>
<td>The registry keys will be updated and the license.dat files in the license folder for the client will not be updated (the client will have no server information listed).</td>
</tr>
<tr>
<td></td>
<td>For more on redundant license servers, see the guide, FLEXnet Licensing End User Guide, available in the installation directory.</td>
</tr>
<tr>
<td>Do not modify the existing license file(s)</td>
<td>No further action is necessary.</td>
</tr>
</tbody>
</table>

**Note**

For more information on licensing Teamcenter lifecycle visualization, see the Installation Guide.

10. In **Ready to Install the Program**, click **Next**.

11. Click **Install**.

   The selected components are installed. This may take several minutes.

12. Click **Finish**.

**Install ClearanceDB components on UNIX**

1. Log in as root.
2. Browse to the Teamcenter lifecycle visualization software distribution image directory.

3. Type ./setup to run the installer.

4. Click Install Software.

5. In Teamcenter lifecycle visualization Setup window, click Next.

6. In the Choose Location window, specify the installation location and click Next.

7. In the Select Platforms window, select the platform on which to install the software and click Next.

8. In the Language Selection window, select the languages to install and click Next.

9. In the Select Components window, select the products and components to install.

   You can choose to install any combination of the following ClearanceDB components on a single machine:

   • Clearance Calculator
   • ClearanceDB Proxy
   • ClearanceDB Client
   • ClearanceDB Server

   **Tip**

   To install ClearanceDB example data, choose Help and Miscellaneous Product Features→ClearanceDB Example Files.

10. Click Next.

11. In the License Type, do one of the following:

<table>
<thead>
<tr>
<th>Choose this option</th>
<th>Then do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not modify the existing license file(s)</td>
<td>No further action is necessary.</td>
</tr>
<tr>
<td>Obtain a license from a license server machine</td>
<td>Enter the Server Name and Server Port.</td>
</tr>
</tbody>
</table>

   **Note**

   For more information on licensing Teamcenter lifecycle visualization, see the Installation Guide.

12. Click Next.
13. In the Ready to Install window, review the current settings (click Back to make any changes) and when ready to continue, click Next.
   The installation process begins and a window displays the progress and lists the files being installed.


15. To close the installer, click Exit.

Install a ClearanceDB patch

Typically there are several steps required to apply a ClearanceDB patch to an existing installation.
1. Review the ReadMe file included with the patch for possible additional instructions.

2. If the patch is a Teamcenter for lifecycle visualization Maintenance Patch or Service Release, run the upgrade installer as described in the Teamcenter for lifecycle visualization Installation Guide or ReadMe file.

3. Run the upd_ClearanceDB_user.sql database upgrade script.

Upgrade an existing database

You must run the upd_ClearanceDB_user.sql script to update an existing database to work with a newer version of ClearanceDB.

1. Open a command prompt and navigate to the ClearanceDB SQL scripts, which are located at the following location within the stand-alone Lifecycle Visualization installation:
   • (Windows) <Mockup installation directory>\Products\Mockup\ClearanceDB\Server
   • (UNIX) <Mockup installation directory>/Visualization/bin/ClearanceDB/Server

2. From the command prompt, type
   sqlplus "<owner>/<password>@<connection>" @ upd_ClearanceDB_user.sql <logfile> <dflt_tablespace> <index_tablespace>
   where

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>The user name of the ClearanceDB Oracle user account.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the Oracle user account.</td>
</tr>
<tr>
<td>connection</td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td>logfile</td>
<td>The name that you want to give the log file that is created when you run the upd_ClearanceDB_user.sql script. For example, updatelog.txt.</td>
</tr>
<tr>
<td>For this parameter</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
</tr>
<tr>
<td>dflt_tablespace</td>
<td>CL_TABLE. This is the name of a ClearanceDB tablespace.</td>
</tr>
<tr>
<td>indx_tablespace</td>
<td>CL_INDEX. This is the name of a ClearanceDB tablespace.</td>
</tr>
</tbody>
</table>

Example

```
sqlplus "clowner/clowner@cldata" @upd_ClearanceDB_user.sql updatelog.txt
CL_TABLE CL_INDEX
```

Note

The ClearanceDB SQL scripts are case-sensitive.

Tip

If the database is updated, the end of the log file displays a message similar to this:

```
On 2007/07/11 at 23:20:00 THE INSTALLATION SUCCEEDED. PL/SQL procedure successfully completed.
```

Upgrade to Oracle 11gR2

For information on upgrading a database to the Oracle 11gR2 release, see Oracle Database Upgrade.
Chapter 3: Configuring ClearanceDB Server

ClearanceDB Server overview

The ClearanceDB Server tier consists of an Oracle Server installation with a database instance configured with the ClearanceDB schema. The ClearanceDB database stores user information, product configurations, rules, conditions, zones, results, and issue dispositions.
Create the ClearanceDB database

On your Oracle Server, create a database instance to store ClearanceDB information.

1. Start the Oracle Database Configuration Assistant.

2. On the Welcome page, click Next.

3. On the Step 1 of 12: Operations page, ensure Create a Database is selected, and then click Next.
4. On the Step 2 of 12: Database Templates page, select **General Purpose**, and then click **Next**.

5. On the Step 3 of 12: Database Identifier page, in the **Global Database Name** box, type a name for the ClearanceDB database.

6. In the **SID** box, type a name for the Oracle System Identifier (SID).

7. Click **Next**.
8. On the Step 4 of 12: Management Options page, accept the defaults, and then click **Next**.

![Database Configuration Assistant, Step 4 of 12: Management Options](image)

9. On the Step 5 of 12: Database Credentials page, select **Use the Same Password for All Accounts**, and then type the password you want to use for the SYS, SYSTEM, DBSNMP, and SYSMAN accounts.

![Database Configuration Assistant, Step 5 of 12: Database Credentials](image)
10. On the Step 6 of 12: Storage Options page, accept the default, and then click Next.

![Database Configuration Assistant, Step 6 of 12: Storage Options](image)

11. On the Step 7 of 12: Database File Locations page, accept the default, and then click Next.

![Database Configuration Assistant, Step 7 of 12: Database File Locations](image)
12. On the Step 8 of 12: Recovery Configuration page, accept the defaults, and then click Next.

![Recovery Configuration Assistant](image)

13. On the Step 9 of 12: Database Content page, ensure Sample Schemas is not selected, and then click Next.

![Database Content Assistant](image)
14. On the Step 10 of 12: Initialization Parameters page, accept the defaults, and then click Next.

15. On the Step 11 of 12: Database Storage page, accept the defaults, and then click Next.
16. On the Step 12 of 12: Creation Options page, ensure **Create Database** is selected, and then click **Finish**.

17. Review the **Database Details** report, and then click **OK**.
Creation of the database begins. When the database is created, the **Database Configuration Assistant** dialog box appears.

18. Click **Exit**.

You are now ready to create the listener.

**Tune the database memory parameters**

Modify the following settings for optimal database performance.

1. Using the Oracle Enterprise Manager **Database Control**, login as a user with SYSDBA privileges to your ClearanceDB database instance.

   ![Database Configuration Assistant](image)

   * User Name: SYS
   * Password: ********
   Connect As: SYSDBA

   ![Login Button](image)

2. On the Administration page, in the **Database Configuration** section, click **Memory Parameters**.

**Database Administration**

**Storage**
- Control Files
- Tablespace
- Temporary Tablespace Groups
- Datafiles
- Rollback Segments
- Redo Log Groups
- Archive Logs

**Database Configuration**
- Memory Parameters
- Undo Management
- All Initialization Parameters
- Database Feature Usage
3. On the Memory Parameters page, in the **Current Allocation** section, click **Disable**.

**Current Allocation**

<table>
<thead>
<tr>
<th>SGA Component</th>
<th>Current Allocation (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Pool</td>
<td>212</td>
</tr>
<tr>
<td>Buffer Cache</td>
<td>356</td>
</tr>
<tr>
<td>Large Pool</td>
<td>4</td>
</tr>
<tr>
<td>Java Pool</td>
<td>4</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total SGA</strong></td>
<td><strong>584</strong></td>
</tr>
</tbody>
</table>

4. On the Disable Automatic Shared Memory Management page, in the **New Size (MB)** box for the **Shared Pool** parameter, type **100**.

<table>
<thead>
<tr>
<th>SGA Component</th>
<th>Current Size (MB)</th>
<th>New Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Pool</td>
<td>212</td>
<td>100</td>
</tr>
<tr>
<td>Buffer Cache</td>
<td>356</td>
<td>600</td>
</tr>
<tr>
<td>Large Pool</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Java Pool</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total SGA</strong></td>
<td><strong>584</strong></td>
<td><strong>584</strong></td>
</tr>
</tbody>
</table>

5. In the **New Size (MB)** box for the **Buffer Cache** parameter, type **600**.

<table>
<thead>
<tr>
<th>SGA Component</th>
<th>Current Size (MB)</th>
<th>New Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Pool</td>
<td>212</td>
<td>100</td>
</tr>
<tr>
<td>Buffer Cache</td>
<td>356</td>
<td>600</td>
</tr>
<tr>
<td>Large Pool</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Java Pool</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total SGA</strong></td>
<td><strong>584</strong></td>
<td><strong>584</strong></td>
</tr>
</tbody>
</table>

6. Click **OK**.

You are now ready to create the ClearanceDB database tablespaces.
Creating the ClearanceDB tablespaces

ClearanceDB requires the following tablespaces within your database instance:

- **CL_TABLE**
- **CL_INDEX**

Create the CL_TABLE tablespace

1. Using the Oracle Enterprise Manager **Database Control**, login as a user with SYSDBA privileges to your ClearanceDB database instance.

   * User Name: SYS
   * Password: ********
   Connect As: SYSDBA

   ![Login button](image)

2. On the Administration page, in the **Storage** section, click **Tablespaces**.

   **Database Administration**

   - **Storage**:
     - Control Files
     - Tablespaces
     - Temporary Tablespace Groups
     - Datafiles
     - Rollback Segments
     - Redo Log Groups
     - Archive Logs
   - **Database Configuration**:
     - Memory Parameters
     - Undo Management
     - All Initialization Parameters
     - Database Feature Usage

3. On the Tablespaces page, click **Create**.

4. On the Create Tablespace page, in the **Name** box, type **CL_TABLE**.

   **General**

   | * Name | CL_TABLE |

5. Click **Add**.

6. On the Add Datafile page, in the **File Name** box, type **CL_TABLE**.
Chapter 3: Configuring ClearanceDB Server

7. In the **File Size** box, type 600.
8. Click **Continue**.
9. On the Create Tablespace page, click **OK**.
   The **CL_TABLE** tablespace is created.

**Create the CL_INDEX tablespace**
1. On the Tablespaces page, click **Create**.
2. On the Create Tablespace page, in the **Name** box, type **CL_INDEX**.
3. Click **Add**.
4. On the Add Datafile page, in the **File Name** box, type **CL_INDEX**.
5. In the **File Size** box, type 180.
6. Click **Continue**.

7. On the Create Tablespace page, click **OK**.
   
The **CL_INDEX** tablespace is created.
Chapter 4: Creating user accounts

Creating user accounts

A ClearanceDB user account is essentially an Oracle user configured to store ClearanceDB products, which include configuration information, unique definitions of rules and conditions, zones, and analysis results. ClearanceDB products are also associated with the status of design issues as specified by the end user in the viewer.

Note
You cannot query across database instances, users, or products.
Create Oracle users

Create Oracle user accounts to store product configurations, unique definitions of rules and conditions, zones, analysis results, and issue dispositions. Execute the `cre_ClearanceDB_user.sql` script to create a ClearanceDB Oracle user.

1. Configure ClearanceDB Server.

2. Open a command prompt and navigate to the ClearanceDB SQL scripts, which are located at the following location within the stand-alone Lifecycle Visualization installation:
   - (Windows) `<Mockup installation directory>\Products\Mockup\ClearanceDB\Server`
   - (UNIX) `<Mockup installation directory>/Visualization/bin/ClearanceDB/Server`

3. At the command prompt, type

```
sqlplus 
"<dba>/<password>@<connection> AS SYSDBA" 
@cre_ClearanceDB_user.sql
<logfile> <owner> <password> <dflt_tablespace> <temp_tablespace>
```

where

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dba</td>
<td>The user name of the SYS or SYSTEM user.</td>
</tr>
<tr>
<td>password</td>
<td>The SYS or SYSTEM password.</td>
</tr>
<tr>
<td>connection</td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td>logfile</td>
<td>The name that you want to give the log file that is created when you run the <code>cre_ClearanceDB_user.sql</code> script. For example, <code>create_owner_log.txt</code>.</td>
</tr>
<tr>
<td>owner</td>
<td>The name you want to give the Oracle user account. For example, <code>CLOWNER</code>.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the Oracle user account. For example, <code>CLOWNER</code>.</td>
</tr>
<tr>
<td>dflt_tablespace</td>
<td><code>CL_TABLE</code>. This is the name of the ClearanceDB tablespace.</td>
</tr>
<tr>
<td>temp_tablespace</td>
<td><code>TEMP</code>. This is the name of the area in the Oracle database for temp data.</td>
</tr>
</tbody>
</table>

Example

```
sqlplus 
"sys/infodba@cldata AS SYSDBA" 
@cre_ClearanceDB_user.sql
ownerlog.txt CLOWNER CLOWNER CL_TABLE TEMP
```

Note

The ClearanceDB SQL scripts are case-sensitive.
Tip

If the user is successfully created, the contents of the log file will look like this:

```
old  1: CREATE USER "&2" IDENTIFIED BY "&3" DEFAULT TABLESPACE "&4"
      TEMPORARY TABLESPACE "&5" PROFILE DEFAULT
new  1: CREATE USER "CLOWNER" IDENTIFIED BY "CLOWNER" DEFAULT TABLESPACE "CL_TABLE" TEMPORARY TABLESPACE "TEMP" PROFILE DEFAULT
User created.
old  1: GRANT "CONNECT" TO "&2"
new  1: GRANT "CONNECT" TO "CLOWNER"
Grant succeeded.
old  1: GRANT "RESOURCE" TO "&2"
new  1: GRANT "RESOURCE" TO "CLOWNER"
Grant succeeded.
old  1: GRANT UNLIMITED TABLESPACE TO "&2"
new  1: GRANT UNLIMITED TABLESPACE TO "CLOWNER"
Grant succeeded.
old  1: GRANT CREATE TABLE TO "&2"
new  1: GRANT CREATE TABLE TO "CLOWNER"
Grant succeeded.
old  1: GRANT CREATE VIEW TO "&2"
new  1: GRANT CREATE VIEW TO "CLOWNER"
Grant succeeded.
old  1: GRANT EXECUTE ANY PROCEDURE TO "&2"
new  1: GRANT EXECUTE ANY PROCEDURE TO "CLOWNER"
Grant succeeded.
old  1: ALTER USER "&2" DEFAULT ROLE ALL
new  1: ALTER USER "CLOWNER" DEFAULT ROLE ALL
User altered.
```

Note

You must now run the `upd_ClearanceDB_user.sql` script.
Update Oracle user accounts

After the creation of the ClearanceDB database, ClearanceDB Oracle user accounts must be updated using the `upd_ClearanceDB_user.sql` script. You must also run this script whenever you want to update to a newer version of ClearanceDB.

1. Open a command prompt and navigate to the ClearanceDB SQL scripts, which are located at the following location within the stand-alone Lifecycle Visualization installation:
   
   - (Windows) `<Mockup installation directory>\Products\Mockup\ClearanceDB\Server`
   - (UNIX) `<Mockup installation directory>/Visualization/bin/ClearanceDB/Server`

2. At the command prompt, type

   ```sql
   sqlplus "<owner>/<password>@<connection>" @upd_ClearanceDB_user.sql <logfile> <dflt_tablespace> <index_tablespace>
   ```

   where

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>The user name of the Oracle user account. For example, CLOOWNER.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the Oracle user account. For example, CLOOWNER.</td>
</tr>
<tr>
<td>connection</td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td>logfile</td>
<td>The name that you want to give the log file that is created when you run the <code>upd_ClearanceDB_user.sql</code> script. For example, updatelog.txt.</td>
</tr>
<tr>
<td>dflt_tablespace</td>
<td>CL_TABLE. This is the name of a ClearanceDB tablespace.</td>
</tr>
<tr>
<td>index_tablespace</td>
<td>CL_INDEX. This is the name of a ClearanceDB tablespace.</td>
</tr>
</tbody>
</table>

   **Example**

   ```sql
   sqlplus "clowner/clowner@cldata" @upd_ClearanceDB_user.sql updatelog.txt CL_TABLE CL_INDEX
   ```

   **Note**

   The ClearanceDB SQL scripts are case-sensitive.
Understanding roles

ClearanceDB roles allow the owner of a ClearanceDB database to grant or restrict access to specific functions within the database. Roles can be associated with multiple users, and unique roles can be combined into a "super" role to provide access to multiple areas of functionality.

To associate ClearanceDB users with roles, the following process must be followed:

1. Designate ClearanceDB users as auxiliary users with the `cre_aux_ClearanceDB_user.sql` installation script.

2. Assign roles to auxiliary users with the `upd_aux_ClearanceDB_user.sql` installation script.

Once auxiliary users are created and associated with roles, use the `upd_aux_ClearanceDB_user.sql` script to revoke roles, and the `del_aux_ClearanceDB_user.sql` script to delete auxiliary users.

The database administrator defines the Oracle user names during the user creation step of the installation process. The DBA also creates the ClearanceDB auxiliary users. ClearanceDB roles may be assigned to auxiliary users by the database administrator or the ClearanceDB database owner.

A ClearanceDB auxiliary user can be given access to a single ClearanceDB database only. The link between an auxiliary user and a ClearanceDB database is established during the creation process of the auxiliary users and cannot be modified. However, an auxiliary user can be deleted from a ClearanceDB database and subsequently recreated with a link to a different database.

An auxiliary user cannot delete a ClearanceDB database nor change its grants. Objects such as tables can be selectively viewed or modified only. Temporary objects can be created and deleted as needed but only indirectly in an encapsulated way.
Roles

The database administrator or the owner of the ClearanceDB database can assign the following roles to ClearanceDB auxiliary users.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG</td>
<td>This role allows auxiliary users to define and maintain the ClearanceDB process configuration (per product) for all of the products in the database.</td>
</tr>
<tr>
<td>RULES</td>
<td>This role allows auxiliary users to define and maintain ClearanceDB rules and conditions (per product) for all of the products in the database.</td>
</tr>
<tr>
<td>ZONES</td>
<td>This role allows auxiliary users to define and maintain the ClearanceDB process configuration (per product and configurations) for all of the products and their configurations in the database.</td>
</tr>
<tr>
<td>MAINT</td>
<td>This role allows auxiliary users to maintain the ClearanceDB database in terms of its content. In particular, the role supports product (metadata) reset in order to force a complete reevaluation of the product, product data deletion, copy, and renaming tasks.</td>
</tr>
<tr>
<td>CALC</td>
<td>This role allows auxiliary users to perform the incremental clearance analysis against the ClearanceDB requirement rulebase either from Mockup or with the Clearance Calculator.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>This role allows auxiliary users to update the ClearanceDB database with clearance analysis results.</td>
</tr>
<tr>
<td>ISSUES</td>
<td>This role allows auxiliary users to query the ClearanceDB database for the clearance issues associated with a given product and its configuration, with the analysis results possibly subject to server side filtering. Also, ClearanceDB reports can be requested and history logs queried.</td>
</tr>
<tr>
<td>CLDB</td>
<td>This role combines all of the basic ClearanceDB roles. The privileges of the user granted the role are still a very strict subset of the privileges of the owner of the schema. It consists of the following basic roles: CONFIG, RULES, ZONES, MAINT, CALC, UPDATE, ISSUES.</td>
</tr>
<tr>
<td>ADMIN</td>
<td>This role allows auxiliary users to administer the ClearanceDB process within the ClearanceDB database. It consists of the following basic roles: CONFIG, RULES, ZONES, and MAINT.</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>VIS</td>
<td>The role allows auxiliary users to perform clearance analysis, persist and manage issues within a particular ClearanceDB environment. It consists of the following basic roles: CALC, UPDATE, and ISSUES.</td>
</tr>
</tbody>
</table>

**Note**

The above roles can be combined as needed to match the responsibilities of individual users.

---

### Create ClearanceDB auxiliary users

Execute the `cre_aux_ClearanceDB_user.sql` script after the ClearanceDB Oracle user account and the corresponding schema are created.

Usage (all on a single command line):

```sql
sqlplus <dba>/<password>@<connection> @cre_aux_ClearanceDB_user.sql <logfile> <AuxClearanceDBUserName> <AuxUserClearanceDBPassword> <DefaultTableSpaceName> <TemporaryTableSpaceName> <ClearanceDBUserName>
```

where

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dba</code></td>
<td>The user name of the SYS or SYSTEM user.</td>
</tr>
<tr>
<td><code>password</code></td>
<td>The SYS or SYSTEM password.</td>
</tr>
<tr>
<td><code>connection</code></td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td><code>logfile</code></td>
<td>The name that you want to give the log file that is created when you run the <code>cre_aux_ClearanceDB_user.sql</code> script. For example, <code>create_auxuser_log.txt</code>.</td>
</tr>
<tr>
<td><code>AuxClearanceDBUserName</code></td>
<td>The name that you want to use for the Oracle ClearanceDB auxiliary user.</td>
</tr>
<tr>
<td><code>AuxUserClearanceDBPassword</code></td>
<td>The password that you want to use for the Oracle ClearanceDB auxiliary user.</td>
</tr>
<tr>
<td><code>DefaultTableSpaceName</code></td>
<td>Oracle default tablespace name to be assigned to the ClearanceDB auxiliary user (cannot be of temporary type). For example, <code>USERS</code>.</td>
</tr>
<tr>
<td><code>TemporaryTableSpaceName</code></td>
<td>Oracle temporary tablespace name to be assigned to the ClearanceDB auxiliary user (must be of temporary type). For example, <code>TEMP</code>.</td>
</tr>
<tr>
<td><code>ClearanceDBUserName</code></td>
<td>Oracle ClearanceDB Oracle user account name. For example, <code>CLOWNER</code>.</td>
</tr>
</tbody>
</table>
Chapter 4: Creating user accounts

Example

```
sqlplus SYSTEM/_MANAGER@ORCL @cre_aux_ClearanceDB_user.sql cre_vis_orcl.log
VIS VIS1234 USERS TEMP CLOWNER
```

Note

The ClearanceDB SQL scripts are case-sensitive.

Privileges

The auxiliary ClearanceDB user is granted the following system privileges at the time of creation:

```
ALTER SESSION CREATE SESSION CREATE SYNONYM UNLIMITED TABLESPACE
```

Synonyms

There are a number of synonyms created for the auxiliary ClearanceDB user at the time of creation. The synonyms are not public. The synonyms are for the ClearanceDB schema owner objects. The set of synonyms is the same for all ClearanceDB auxiliary users regardless of the roles that may be granted in the future.

Example

```
CREATE SYNONYM VIS.EAICL_RULE_BUFFER FOR CLOWNER.EAICL_RULE_BUFFER;
```

Managing User Roles

Execute the `upd_aux_ClearanceDB_user.sql` script after the ClearanceDB auxiliary user is created.

Usage (all on a single command line):

```
sqlplus <owner>/<password>@<connection> @upd_aux_ClearanceDB_user.sql <logfile> <AuxClearanceDBUserName> vRoleAction <RoleName>
```

where

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>The user name of the Oracle user account. For example, CLOWNER.</td>
</tr>
<tr>
<td>password</td>
<td>The password for the Oracle user account. For example, CLOWNER.</td>
</tr>
<tr>
<td>connection</td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td>logfile</td>
<td>The name that you want to give the log file that is created when you run the <code>upd_aux_ClearanceDB_user.sql</code> script. For example, <code>updateauxlog.txt</code>.</td>
</tr>
<tr>
<td>AuxClearanceDBUserName</td>
<td>The Oracle ClearanceDB auxiliary user name.</td>
</tr>
<tr>
<td>RoleAction</td>
<td>GRANT or REVOKE.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RoleName</td>
<td>Any defined ClearanceDB role name. For example, VIS. If empty all roles are revoked.</td>
</tr>
</tbody>
</table>

**Example**

```
sqlplus CLOWNER/CLOW1234@ORCL @upd_aux_ClearanceDB_user.sql upd_vis_orcl1.log VIS GRANT VIS

sqlplus CLOWNER/CLOW1234@ORCL @upd_aux_ClearanceDB_user.sql upd_vis_orcl2.log VIS REVOKE

sqlplus CLOWNER/CLOW1234@ORCL @upd_aux_ClearanceDB_user.sql upd_vis_orcl3.log VIS GRANT CLDB
```

**Note**
The ClearanceDB SQL scripts are case-sensitive.

**Deleting ClearanceDB Auxiliary Users**

Execute the `del_aux_ClearanceDB_user.sql` script after the ClearanceDB auxiliary user is created.

**Usage (all on a single command line):**

```
sqlplus <dba>/<password>@<connection> @del_aux ClearanceDB_user.sql <logfile> <AuxClearanceDBUserName>
```

where

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dba</td>
<td>The user name of the SYS or SYSTEM user.</td>
</tr>
<tr>
<td>password</td>
<td>The SYS or SYSTEM password.</td>
</tr>
<tr>
<td>connection</td>
<td>The Local Net Service Name for the ClearanceDB Oracle database.</td>
</tr>
<tr>
<td>logfile</td>
<td>The name that you want to give the log file that is created when you run the <code>del_aux_ClearanceDB_user.sql</code> script. For example, <code>delauxlog.txt</code>.</td>
</tr>
<tr>
<td>AuxClearanceDBUserName</td>
<td>Oracle ClearanceDB auxiliary user name. For example, VIS.</td>
</tr>
</tbody>
</table>

**Example**

```
Sqlplus SYSTEM/_MANAGER@ORCL @del_aux_ClearanceDB_user.sql del_vis_orcl.log VIS
```
**Note**

The ClearanceDB SQL scripts are case-sensitive.
Chapter 5: Creating the ClearanceDB Work Area

Setting up the work area

The ClearanceDB Work Area is the location on your machine where you can create and manage ClearanceDB products. Created by the create_ClearanceDB_working_dir.pl script, the ClearanceDB Work Area contains configuration files that you use to create a connection with the database, run the proxy, and update the database with product information such as rules, conditions, zones, and results.

Note
To administer ClearanceDB, you must create and work from a ClearanceDB Work Area.

Global configuration files are located in the root of the work area

Each of your products and proxies is located in a separate directory
Create the ClearanceDB Work Area

When working with the Clearance Calculator, ClearanceDB Client, and ClearanceDB Proxy, you must create a specific ClearanceDB Work Area on your machine, from which you can create and manage ClearanceDB products and instances of the ClearanceDB Proxy.

1. Install the Clearance Calculator, ClearanceDB Client, or ClearanceDB Proxy.

2. (UNIX) Update your PATH definition to include the ClearanceDB installation directory.

   ```
   Example
   <Mockup installation directory>/Visualization/bin/ClearanceDB
   ```

3. Open a command prompt and type

   ```
   create_ClearanceDB_working_dir.pl <path> [-df]
   ```

   where
For this parameter | Type |
--- | --- |
path | The path to the location on your machine where you want to create the work area.  
**Note**  
The path parameter is used to specify the location of the work area, not the name of the work area, which is always ClearanceDB_Work_Area. |
—df | The option to populate the work area with a default product.  
**Note**  
The ClearanceDB Work Area must have a default product, which is used for the creation of new products. You can create the default product now, or run the create_default_product.pl script to create it later. |

Example

create_ClearanceDB_working_dir.pl c: \ -df

A new directory named ClearanceDB_Work_Area is created at the specified location.

**Global configuration options**

The ClearanceDB global configuration file (Clearance.cfgglobal), located in the root of the ClearanceDB Work Area, controls the default settings for the ClearanceDB Proxy, ClearanceDB Client, and Clearance Calculator. The settings from this file are applied to all of the clearance products in the ClearanceDB Work Area, unless the settings are manually copied into the Clearance.cfgproduct for a specific product. Any settings specified in the Clearance.cfgproduct take precedence over the Clearance.cfgglobal file.

**Section 1: Teamcenter Configuration**

**TC_ROOT**  
Specifies the Teamcenter root directory.

**TC_DATA**  
Specifies the Teamcenter TC_DATA directory.
TeamcenterUserId
Specifies a Teamcenter user name with Teamcenter system administration privileges. If no user name is provided, ClearanceDB uses the operating system user name.

**Note**
This is the same option as the `bomwriter -u=` command line argument.

TeamcenterUserPassword
This setting is deprecated. The Teamcenter user password is now stored in a separate file, and the path to the file is specified with the TeamcenterUserPasswordFile setting. Now if the password is specified with TeamcenterUserPassword, it is written to an unencrypted file in the user’s home directory with a randomly generated name beginning with CLDB.

TeamcenterUserPasswordFile
Specifies the path to a file containing the Teamcenter user password.

**Note**
This is the same option as the `bomwriter -pf=` command line argument.

TeamcenterUserGroupId
Specifies the Teamcenter group ID. If no group ID is provided, ClearanceDB uses the default group of the specified Teamcenter user.

**Note**
This is the same option as the `bomwriter -g=` command line argument.

RevisionRule
Specifies the revision rule for the product’s top level product structure node.

**Note**
This is the same option as the `bomwriter -revision_rule=` command line argument.

SavedVariantRule
Specifies the saved variant configuration to pass to the `bomwriter`.

LexicographicalVariantAnalysis
Defines the mode of the evaluation of variant condition relational expressions. Choose one of the following:

**No** — If undeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

**Yes** — If undeterminable variant conditions exist, evaluate them lexicographically.
Note
Variant conditions that cannot be evaluated based on textual representations are undeterminable.

**VariantAnalysisClientTraceFile**
Specifies the name for the log file of the variant analysis. By default, this is not set, and a log file is not generated.

**Example**
VariantAnalysisClientTraceFile=cldb_variant_analysis.log

**VariantAnalysisClientTraceFlags**
Specified the content of the log file of the variant analysis. The trace flags are additive, concatenated with the plus (+) character.

**Example**
VariantAnalysisClientTraceFlags=CONFIGURATION+STATISTICS

Add any of the following flags:

- **CONFIGURATION** Lists the variant analysis configuration attributes.
- **STATISTICS** Provides the basic variant analysis statistics.
- **VARIANTCONDITIONS** Provides details of textual variant conditions.
- **DETERMINABILITY** Lists the determinability of variant conditions.
- **SATISFIABILITY** Lists the satisfiability of variant conditions.
- **EXCLUDEDVCPAIRS** Lists excluded variant condition pairs.
- **EXCUSEDUIDPAIRS** Lists excluded clearance element pairs as ABSOCCs.
- **EXCLUDEDNGIDPAIRS** Lists excluded clearance element pairs as NGIDs.
- **IMPACT** Lists the impact of variant conditions on the product structure.

**Caution**
Enabling the **EXCUSEDUIDPAIRS** and **EXCLUDEDNGIDPAIRS** trace flags may result in extremely large log files.

**VariantLogicalExpression**
Defines the variant condition UserValue title and BOM line property pair in the PLM XML file. This has the form of:
“Variant Condition”: bl_variant_condition

“Variant Formula”: bl_formula

Example

VariantLogicalExpression="VC:bl_variant_condition"

Note

This must match the content of the **BomWriterUserAttributes** setting.

**RunLevel**

Specifies the actions of the **analyze_managed_product.pl** script. Use a value from 1 to 6.

1 — In the product directory, a.vvi file is created, which is used by the **bomwriter** to generate a .plmxml file referencing the managed data.
2 — Using the .vvi and .plmxml files in the product directory, the Clearance Calculator performs analysis upon the managed product data and generates a results file.
3 — The results file is uploaded to the ClearanceDB database.
4 — Both the **RunLevel 1** and 2 actions are performed.
5 — Both the **RunLevel 2** and 3 actions are performed.
6 — All of the **RunLevel** actions are performed, 1, 2, and 3.

**BomWriterUserAttributes**

Specifies the variant conditions for the **bomwriter** to include in the generated .plmxml file. Type these according to the following syntax:

**target:Instance,key:myAttribute,literal:\"My Attribute Value\"**

Note

This is the same option as the **bomwriter -ua=** command line argument.

Example

Use this option to include variant model related BOM line properties in the generated .plmxml file. For example:

**BomWriterUserAttributes=target:Instance,key: VC,prop: bl_variant_condition**

**TeamcenterWebServerPath**

Specifies the Teamcenter web server path, including the protocol, host name, and port number. Type this according to the following syntax:

**TeamcenterWebServerPath=http://machine_name:port**
Example

JtDataStagingProcess
Specifies to use the Teamcenter load_fcccache utility to download the model data from the Teamcenter server to the local system for clearance analysis. Enabling this option pre-populates the FMS client cache (FCC), which leads to faster and more reliable analysis.

Valid values are 0 (off) or 1 (on). The default value is 0.

JtDataStagingProcessErrorLimit
Specifies when to abort the analyze ManagedProduct.pl script, based on the following custom error codes:

FILECOPY_FAILED
INVALID_DAKID_FORMAT
FCC_OPENFILE_FAILED
FCC_DOWNLOAD_FAILED
GET_READ_TICKET_FAILED
DATASET_READ_FAILED
PLMXML_MISSING_JT
CHMOD_FAILED
COPYOUT_CLEANUP_FAILED

You can specify a numerical value for each error code. By default, the error codes are given values that correspond to the severity of the problem, with the lowest value representing the most severe failure. If a value is not specified for the JtDataStagingProcessErrorLimit setting, the script stops whenever an error occurs during the staging process.

The default value is 30.

Note
The default values are recommended for the JtDataStagingProcessErrorLimit setting and the related error codes. Essentially, with these defaults the clearance analysis will abort when any of these errors occur, and you can use the reported error message to troubleshoot the problem.

FILECOPY_FAILED
Specifies the error code that indicates a copy operation to the output directory has failed.

The default value is 20.

INVALID_DAKID_FORMAT
Specifies the error code that indicates an invalid DAKID was found.

The default value is 21.

FCC_OPENFILE_FAILED
Specifies the error code that indicates the FCC failed to open the file using the ticket.
The default value is 22.

**FCC_DOWNLOAD FAILED**
Specifies the error code that indicates a failure most likely due to a missing file in the volume.
The default value is 23.

**GET_READ_TICKET FAILED**
Specifies the error code that indicates a read ticket failed.
The default value is 24.

**DATASET_READ FAILED**
Specifies the error code that indicates no read access on the dataset.
The default value is 25.

**PLMXML_MISSING_JT**
Specifies the error code that indicates a missing JT file reference in the .plmxml file.
The default value is 26.

**CHMOD FAILED**
Specifies the error code that indicates a failure to set the access mode during copy out.
The default value is 27.

**COPYOUT_CLEANUP FAILED**
Specifies the error code that indicates a failure to remove a file during lifetime cleanup.
The default value is 28.

**CopyOutLocation**
Specifies the location for the dataset files downloaded from Teamcenter server.
The default location is the product directory in the ClearanceDB Work Area.

**Note**
Do not use special characters in folder names.

**BucketCount**
Specifies how many directories to use for the cached files. Spreading the files over multiple directories can lead to better performance.
The default value is 30.

**UseAbsoluteLocation**
Specifies to use an absolute value for the location attribute in the .plmxml file generated by the bomwriter. It is usually better to have a relative reference, although there are instances where an absolute reference is required, such as when the .plmxml file is moved to a different location from the referenced files.
The default value is No.
**DirAccessMode**
Specifies the access mode setting for the directories created to hold the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**FileAccessMode**
Specifies the access mode setting for the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**BucketPrefix**
Specifies a prefix to add to the names of directories created to hold the cached files.

The default value is **RW**.

**FilenamePrefix**
Specifies a prefix to add to the names of the cached files.

The default value is **fmsr_**.

**CopyOutLifetime**
Specifies the lifetime of the files cached in the **StagingProcessDownloads** directory. The directory is scanned for files older than the specified value, which are removed. The lifetime value is specified in seconds, where one day is equal to 86400 seconds and two weeks is equal to 1209600 seconds.

The default value is **1209600** (two weeks).

**Note**
This option requires the **FilenamePrefix** option to be set since it uses the prefix as validation of ownership to prevent the accidental removal of files.

**LifetimeCheck**
Specifies to scan the **StagingProcessDownloads** directory for files older than the **CopyOutLifetime** value.

Valid values are **0** (off) or **1** (on). The default value is **0**.

**LifetimeCheckInterval**
Specifies how often to scan the **StagingProcessDownloads** directory for files older than the **CopyOutLifetime** value. If the directory holds many files and it is not important to check the lifetime each time the **analyze_managed_product.pl** script is run, you can improve performance by increasing the value so the check is made less frequently. If the specified value is 10, the lifetime check occur once over the course of 10 script executions.

The default value is **10**.

**LifetimeProcessLimit**
Specifies the maximum number of seconds the file lifetime check is allowed to continue. The lifetime check randomly examines cached files. If the **StagingProcessDownloads** directory
consists of many files, this option has the effect of randomly processing a subset of files each
time the lifetime check takes place. Over time, all of the files are examined.

The default value is **300**.

**LogTypes**
Specifies the type of logging to be reported. The following are valid log types:

- **NONE**
- **ERROR**
- **WARNING**
- **INFORMATION**
- **DEBUG**
- **PERFORMANCE**
- **ALL**

**Note**
Use the + sign to use multiple log types. For example, **ERROR+WARNING**.

**RulesObject = Requirement Rules from ClearanceDB Server**
Specify to perform analysis using the product's rules and conditions from the ClearanceDB
database. To enable this option, remove the number sign symbol (#) that precedes the**RulesObject = Requirement Rules from ClearanceDB Server** line.

**RulesObject = Variant Analysis**
Specify to perform analysis using Teamcenter variant conditions. To enable this option, remove
the number sign symbol (#) that precedes the **RulesObject = Variant Analysis** line.

**ResultsObject = ClearanceDB Results**
Specify to generate a ClearanceDB results file (**ClearanceResultsDbUpload.csvcldb**). To
enable this option, remove the number sign symbol (#) that precedes the **ResultsObject = ClearanceDB Results** line.

**Attribute=__PLM_ABSOCC_UID, Part and Parents (Part First)**
Specify to use product structure absolute occurrences. To enable this option, remove the number
sign symbol (#) that precedes the **Attribute=__PLM_ABSOCC_UID, Part and Parents (Part First)** line.

**Section 2: ClearanceDB Proxy Configuration**

**OracleClientDirectory**
Specifies the full directory path containing the Oracle Instant Client files. For example, **OracleClientDirectory=c:\apps\instantclient_11_2** or **OracleClientDirectory=/opt/instantclient_11_2**.
LogFile
Specifies the name of the file where ClearanceDB Proxy messages are logged.

Compress
Specifies whether or not the communications between the ClearanceDB client applications and the ClearanceDB Proxy are compressed. Compression results in higher performance. Choose one of the following:
True — Compress ClearanceDB Proxy messages.
False — Do not compress ClearanceDB Proxy messages.

Port
Specifies the port number for ClearanceDB Proxy to use, along with the number of threads that you want to listen to the port. For example, 7206,8.

Note
You can specify multiple port numbers like this:

```
port1,number of threads
port2,number of threads
```

ProxyClientPassword
Specifies an optional password for ClearanceDB Proxy.

Note
If you require a password for clients to access the database, it must be specified in the DBC file like this:

```
DATASOURCE=<proxy_name>,<port>/<password>
```

CONNECT_DATA
Specifies the Oracle connection information. Type the information according to the following syntax:

```
<connect_name>,<user>/<password>@<service_name>
```

Note
The value used for `<connect_name>` must match the value specified in the DBC file.

If you are using an encrypted password in an external file, type the information according to the following syntax:

```
<connect_name>,<user>@<service_name>
```

Use the ClearanceDBUserPasswordFile setting to specify the full path and file name for the file containing the encrypted Oracle user password.
ClearanceDBUserPasswordFile
Specifies the path to a file containing the Oracle user password.

AdminProxyPassword
Specifies an optional password used to stop a ClearanceDB Proxy via ClearanceDB Client. This password should be restricted to the proxy administrator only.

Section 3: ClearanceDB update/upload settings

ClearanceDBResultsUploadFile
The name and, optionally, the path of the results file generated by the Clearance Calculator. By default, the name of this file is ClearanceResultsDbUpload.csvcldb.

UploadMethod
The option specifying how to merge data with the ClearanceDB database. You can specify one of the following:

- **Proxy** — Perform updates using the ClearanceDB Proxy.
- **NoProxy** — Perform updates using the Oracle Client only.

**Note**
To use this option, the full Oracle Client must be installed on the machine from which you want to make database updates.

OracleSqlLoaderExecutable
The name of the Oracle SQL*Loader executable.

**Note**
This is required only if UploadMethod is set to NoProxy.

OracleSqlPlusExecutable
The name of the Oracle SQL*Plus executable.

**Note**
This is required only if UploadMethod is set to NoProxy.

ClearanceDBOracleUserCredentials
The Oracle user account information. Type this according to the following syntax:

**username/password**

**Note**
This is required only if UploadMethod is set to NoProxy.
ClearanceDBOracleNetServiceName
The Oracle Net Service Name.

Note
This is required only if UploadMethod is set to NoProxy.

Section 4: Clearance Analysis configuration

ClearanceExeOptions
-\texttt{d <value>} — Specifies the clearance distance, in model units.
-\texttt{n} — Checks clearance using NURBS data, if present in your model.
-\texttt{c} — Checks for points of contact and penetration.
-\texttt{p} — (Valid with -c option only) Calculates the depth of penetrations.
-\texttt{t <value>} — (Valid with -c and -p options only) Specifies the contact tolerance.
-\texttt{e} — Automatically sends e-mail messages to owners, as specified in your Clearance Manager preferences.
-\texttt{s} — Shows only error messages as the Clearance Calculator runs.
-\texttt{r} — Specifies the maximum distance requirement, in model units.
-\texttt{q <dbc>} — Specifies to perform analysis against the ClearanceDB database. You must specify a DBC file to use this option.

Note
You only need to use this option if manually running the Clearance Calculator from the command line, without using the analyze_product.pl or analyze_managed_product.pl Perl script.

-\texttt{Q <dbc>} — Specifies to perform a ClearanceDB query. A DBC file is optional.

Note
You only need to use this option if manually running the Clearance Calculator from the command line, without using the analyze_product.pl or analyze_managed_product.pl Perl script.

-\texttt{m <crl>} — Includes material thickness in the clearance check. Specify the \texttt{crl} option for more accurate results.
-\texttt{l} — Applies the default layer filter when checking clearance.
-\texttt{i} — Generates 2D images of the element pairs involved in clearance violations.

Note
To generate 2D images of clearance issues, you must set the \texttt{ResultsObject} option to \texttt{General Clearance Results} in the Clearance.cfgglobal file.
-f <output filename> — Saves clearance results as a comma-delimited, quote-enclosed, ASCII text file. By default, the results file is named Results.txt. To give a results file a different name, type the name after the -f parameter at the command prompt.

**ElementType**
The clearance element type. Valid options are:

```plaintext
Part
Component
EndItem
EndItemComponent
```

The default element type is **Part**.

**RecognizeCADComponents**
Specifies whether or not CAD components are recognized during analysis.

Valid values are **0** (off) or **1** (on). The default value is **0**.

**NumberOfProcesses**
The number of CPU processes to use for clearance analysis. This option requires a machine with multiple processors or multiple cores. The default value is **1**.

**Note**
The maximum size a process can have may be limited by the operating system, the system administrator, or otherwise.

**NumberOfThreads**
The number of threads to use for clearance analysis. The default value is the number of cores on the machine. Whenever the number of threads is greater than one, the number of processes is automatically set to one, regardless of the value in the configuration file.

**Note**
The maximum size a process can have may be limited by the operating system, the system administrator, or otherwise.

**AbsMemLimit**
The amount of physical RAM in megabytes to use for batch mode analysis. When the memory limit is exceeded, data is unloaded until memory usage falls below the specified amount. If both AbsMemLimit and RelMemLimit options are used, the lesser effective value is chosen.

**RelMemLimit**
The percentage of physical RAM to use for Batch Mode analysis. When the memory limit is exceeded, data is unloaded until memory usage falls below the specified percentage. The default percentage is 100 percent. You also can force the application to use the virtual memory available on your machine. You can set a memory limit of up to 200 percent of your physical RAM.
Note
If both AbsMemLimit and RelMemLimit options are used, the lesser effective value is chosen.

RulesObject
How to obtain the clearance requirement. Valid options are:

Universal Clearance Requirement
Requirement Rules from ClearanceDB Server
Variant Analysis

Note
If you want to obtain clearance requirements from your organization's ClearanceDB server, specify Requirement Rules from ClearanceDB Server.

ResultsObject
How to handle the clearance results. Valid options are:

General Clearance Results
ClearanceDB Results

Note
To upload results to the server, specify ClearanceDB Results.
You can list the ResultsObject option twice to specify both General Clearance Results and ClearanceDB Results.

Attribute
Specifications to filter clearance results according to attribute metadata associated with your product structure. Specify each attribute you want to use, followed by the assembly search method. Each attribute specification requires a separate entry in the Clearance.cfgglobal or Clearance.cfgproduct file.

Attribute entries must use the following syntax:

Attribute=[attribute_name], [order_of_search_method]

You can search the assembly according to the following criteria:

Part Only
Part Parent Only
Part and Parents (Part First)
Part and Parents (Root First)
Parents Only (Part Parent First)
Parents Only (Root First)

To work with Teamcenter managed products, specify this attribute:
Attribute=__PLM_ABSOCC_UID,Part and Parents (Part First) |
The following entries are examples:
Attribute=Translation Date,Part Only |
Attribute=End Item,Part Only |
Attribute=End Item,Part Parent Only |
Attribute=End Item,Part and Parents (Part First) |
Attribute=End Item,Part and Parents (Root First) |
Attribute=End Item,Parents Only (Part Parent First) |
Attribute=End Item,Parents Only (Root First) |

EnabledMemAndThroughputLog
Specifies whether to create a log file in the batch clearance directory to log the memory usage and the element pair throughput at set intervals. Valid options are 0 for no log file or 1 to create the log file.

MemAndThroughputLogInterval
Number of seconds to specify the interval between entries if EnabledMemAndThroughputLog is set to 1.

GracefulShutdownTimeout
The time duration the operating system is given to abort the worker process. The specified value is in seconds. The default value is 60.

MaxFailures
The maximum number of failures per analysis run. The default value is 1000000.

MaxRelFailures
The maximum relative number of failures per run in respect to all element pairs in the test bed. The specified value is a percentage. The default value is 100.

ProgressDirectory
The directory where intermediate files are placed, including the log file.

OverallTimeout
The maximum duration of the analysis. The default value is 604800, which equals one week.

PairTimeout
The maximum duration of the analysis of a single element pair. If the analysis exceeds the specified time the analysis fails. The default value is 3600, which equals one hour.

RestartAfterPairTimeout
Specifies whether to restart the clearance analysis if an analysis of a single element pair times out (exceeds the PairTimeout duration). Valid values are 0 for no restart or 1 to restart. If the clearance analysis is not restarted, the pair is included as an undefined result in the log file.

MonitorResolution
The maximum frequency of the progress monitoring to standard output. The default value is 60000, which equals one hour.
MaxPairAttempts
The maximum number of possible attempts to analyze an element pair in the same mode (approximate or precise). 1 equals approximate and 2 equals precise. The default is 2.

MaxFailuresPerPart
The maximum number of failures before a part is excluded from the analysis. The default value is 2.

MaxRelFailuresPerPart
The maximum relative number of failures before a part is excluded from the analysis in respect to all occurrences of the element in the test bed. The specified value is a percentage. The default value is 100.

LoadFailures
To include failed element pairs in the results file. Valid options are 0 or 1 for false and true, respectively.

MaxLogFileSize
The maximum size of individual log files in megabytes. The minimum value is 32 MB and the maximum value is 2048 MB (minus 1 byte subtracted). The default value is 2048.

IdentifyPartsWithDatastores
If part occurrences are identified with file names or with part names in the log file. Valid options are 0 or 1 for false and true, respectively. The default value is 1.

IdentifyPartOccurrencesWithMetadata
If part occurrences are identified with CADIDs or with specified metadata. No values or blank values indicate the choice of CADIDs. By default, part occurrences are identified with CADIDs.

RecordPartPairAdditionEvents
If test bed (element pair) creation events are recorded in the log file. Valid options are 0 or 1 for false and true, respectively. The default value is 1.

RecordClearanceResultAdditionEvents
If clearance results creation events are recorded in the log file. Valid options are 0 or 1 for false and true, respectively. The default value is 1.

OffsetInFaceNormalDirection
(For thick sheet clearance analysis) Which side of the surface the thickness is applied to. Valid options are 0 or 1 for false and true, respectively. The default value is the current registry value.

ThicknessAttribute
(For thick sheet clearance analysis) Find the thickness value for the surface parts. This can be used to override the default CAD_PROP_MATERIAL_THICKNESS property name or the currently defined value in the registry. The default value is CAD_PROP_MATERIAL_THICKNESS or the current registry value.

DefaultMaterialThickness
(For thick sheet clearance analysis) The default thickness value for surface parts that are missing the thickness attribute (CAD_PROP_MATERIAL_THICKNESS). If the UseDefaultValue option is set to 1 (true), the default value is the current registry value.
UseDefaultValue
(For thick sheet clearance analysis) To enable or disable the DefaultMaterialThickness value for surface parts with no material thickness property (CAD_PROP_MATERIAL_THICKNESS). Valid options are 0 or 1 for false and true, respectively.

SnapshotImageDirectory
(For use with 2D image captures) The directory where 2D image captures are placed.

SnapshotImageSize
(For use with 2D image captures) The size of 2D image captures. In pixels, specify the width and height of the generated images. The default size is 512,512.

SnapshotImagePNG
(For use with 2D image captures) The 2D image capture format. You can generate 2D captures in the PNG or JPEG formats. Use a value of 1 to create PNG images; use a value of 0 to create JPEG images. By default, JPEG files are created.

SnapshotImageBackground
(For use with 2D image captures) The color to use for the background shown behind the two elements involved in the clearance issue. This setting requires separate red, green, and blue values. By default, the background is set to 190,210,225.

SnapshotImagePartColor1
(For use with 2D image captures) The color to use for the first part involved in the clearance issue. This setting requires separate red, green, and blue values. If this setting is used, by default the part color is set to 0,0,255. If this setting is not turned on the original part color is used.

SnapshotImagePartTransp1
(For use with 2D image captures) The transparency setting for the first part involved in the clearance issue. You can use any value from 0.0 (invisible) to 1.0 (opaque). If this setting is used, the default value is 0.5.

SnapshotImagePartColor2
(For use with 2D image captures) The color to use for the second part involved in the clearance issue. This setting requires separate red, green, and blue values. If this setting is used, by default the part color is set to 0,255,0. If this setting is not turned on the original part color is used.

SnapshotImagePartTransp2
(For use with 2D image captures) The transparency setting for the second part involved in the clearance issue. You can use any value from 0.0 (invisible) to 1.0 (opaque). If this setting is used, the default value is 0.5.

SnapshotImageAdvTransparency
(For use with 2D image captures) To render part transparency with fewer artifacts. Use a value of 1 to turn on advanced transparency.

SnapshotImageShowCPPoints
(For use with 2D image captures) To show the points of contact or penetration. Use a value of 1 to include the contact and penetration points in 2D image captures.
SnapshotImageCPPPointsColor
(For use with 2D image captures) The color to use for points of contact or penetration. This setting requires separate red, green, and blue values. If this setting is used, by default the contact and penetration point color is set to 255,255,0.

SnapshotImageCPPPointsSize
(For use with 2D image captures) The size, in pixels, to use for points of contact or penetration. If this setting is used, the default value is 3.

SnapshotImageZoom
(For use with 2D image captures) The zoom factor for clearance issues. You can use any value from 0 to 100. A value of 0 shows the entire element pair. A value of 100 zooms to the contact or penetration points, or distance lines. If this setting is used, the default value is 40.

SnapshotImageShowDistanceLines
(For use with 2D image captures) Display distance lines in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, distance lines are not shown.

SnapshotImageShowResultVector
(For use with 2D image captures) Display the result vector in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, the result vector is not shown.

SnapshotImageShowBBox
(For use with 2D image captures) Display the bounding box in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, the bounding box is not shown.

Configure the global DBC file
You must modify the global Database Connection (DBC) file with the system name and port number of the machine running your ClearanceDB Proxy. You must also specify a name for the proxy connection, which must be the same as the connect_name value specified in the CONNECT_DATA string in the Clearance.cfgglobal file.

Note
By default, the global.dbc file is used by ClearanceDB components to connect with the specified database. You can also create additional DBC files to connect with databases on a product by product basis. Product-specific DBC files are created when you run the create_product.pl script to create products. Product-specific DBC files override the global DBC file.

1. Navigate to the ClearanceDB Work Area.

2. Using a text editor, open global.dbc and specify the following settings:
For this parameter | Type
---|---
DATASOURCE | The ClearanceDB Proxy system name and port number.

**Note**
If the ClearanceDB Proxy you want to connect to requires a password, include the password at the end of the DATASOURCE string like this:

```
<machine>:<port>/<password>
```

CONNECT_DATA | A name to identify the connection.

3. Save the file.

**Configure the global .vvi file**

If you want to work with Teamcenter managed data, you must specify the web server protocol used by your Teamcenter installation in the global.vvi file.

1. Navigate to the ClearanceDB Work Area.

2. Using a text editor, open the global.vvi file.

3. For the 0_Protocol parameter, type your web server protocol.

```
Example
0_Protocol = http
```

4. Save the file.
Chapter 6: Configuring and running ClearanceDB Proxy

ClearanceDB Proxy overview

ClearanceDB Proxy is essentially the “bridge” that connects the Clearance Calculator and the viewer with the database. It consists of two applications, the ClearanceDbProxyServer and the ClearanceDbProxyClient. Only the ClearanceDbProxyServer is started by the administrator; the ClearanceDbProxyClient is invoked automatically.

ClearanceDB Proxy uses the Oracle Instant Client to connect to the ClearanceDB database on the Oracle server. You can download the Instant Client from the downloads section of http://www.oracle.com.
Install the Oracle Instant Client (Windows)

ClearanceDB Proxy uses the Oracle Instant Client to connect to the ClearanceDB database on the Oracle server.

1. Open the Oracle Instant Client downloads page:
   

   **Tip**
   
   If the location of the downloads page changes and the above link no longer works, type *Instant Client Downloads* in the search box in the upper right corner of the Oracle home page.

2. From the list of available versions, choose the Instant Client for Windows 32-bit or Windows 64-bit (x64), as appropriate to your system.

3. After accepting the Oracle license agreement, download the Basic (not Basic Lite) package for your platform. Choose version 11.2.0.2 (or newer 11.x version, if 11.2.0.2 is not available).

4. When the download is complete, extract the installation files to a directory accessible to the ClearanceDB software.

5. Confirm that a new directory containing the Instant Client files is created in the specified location.

6. Within the Instant Client directory, confirm that the file *orاسql11.dll* exists. If there is a file with a newer version (such as *orاسql12.dll*), copy the newer file, and rename the copy to *orاسql11.dll*.

7. In the ClearanceDB Work Area, open *Clearance.cfgglobal* with a text editor.

8. In the **Section 2: ClearanceDB Proxy Configuration** area of the file, for the *OracleClientDirectory* setting, type the full path of the Instant Client subdirectory.

   **Example**
   
   OracleClientDirectory=d:\apps\instantclient_11_2

9. Save the file.

Install the Oracle Instant Client (UNIX and Linux)

ClearanceDB Proxy uses the Oracle Instant Client to connect to the ClearanceDB database on the Oracle server.

1. Open the Oracle Instant Client downloads page:
   
Tip
If the location of the downloads page changes and the above link no longer works, type **Instant Client Downloads** in the search box in the upper right corner of the Oracle home page.

2. From the list of available versions, choose the Instant Client for Linux x86-64 or Instant Client for Solaris Operating System (SPARC) (64-bit), as appropriate for your system.

3. After accepting the Oracle license agreement, download the Basic (not Basic Lite) package for your platform. Choose version 11.2.0.3 (or newer 11.x version, if 11.2.0.3 is not available).

4. When the download is complete, extract the installation files to a directory accessible to the ClearanceDB software.

5. Within the Instant Client directory, confirm the existence of the file **libclntsh.so.11.1** with these exact version numbers. Note at the time of this writing the library shows **11.1** even for **11.2** libraries.

   **Example**
   
   $ cd /opt/instantclient_11_2
   $ ls libclntsh.so.*
   libclntsh.so.11.1*

6. If your version of **libclntsh** reflects a different (newer) version, for example, **libclntsh.so.11.2**, create a symlink for **libclntsh.so.11.1**, as shown below. The first argument is the name of the library in your installation.

   **Example**
   
   ln –fs libclntsh.so.11.2 libclntsh.so.11.1

7. In the ClearanceDB Work Area, open **Clearance.cfgglobal** with a text editor.

8. In the **Section 2: ClearanceDB Proxy Configuration** area of the file, for the **OracleClientDirectory** setting, type the full path of the Instant Client subdirectory.

   **Example**
   
   OracleClientDirectory=/opt/instantclient_11_2

9. Save the file.
Configure the default ClearanceDB Proxy settings

Configure the global settings for the ClearanceDB Proxy in the Clearance.cfgglobal file. These settings are used as defaults when you use the create_proxy.pl script to create a new instance of the proxy.

1. Install the ClearanceDB Proxy.

2. In the ClearanceDB Work Area, open Clearance.cfgglobal with a text editor.

3. Specify the following settings in the Section 2: ClearanceDB Proxy Configuration section of the file to configure the proxy:

   - **OracleClientDirectory**: Specifies the full directory path containing the Oracle Instant Client files. For example, `OracleClientDirectory=c:\apps\instantclient_11_2` or `OracleClientDirectory=/opt/instantclient_11_2`.

   - **LogFile**: Specifies the name of the file where ClearanceDB Proxy messages are logged.

   - **Compress**: Specifies whether or not the communications between the ClearanceDB client applications and the ClearanceDB Proxy are compressed. Compression results in higher performance. Choose one of the following:

     - **True** — Compress ClearanceDB Proxy messages.
     - **False** — Do not compress ClearanceDB Proxy messages.

   - **Port**: Specifies the port number for ClearanceDB Proxy to use, along with the number of threads that you want to listen to the port. For example, `7206,8`.

     *Note*

     You can specify multiple port numbers like this:

     ```
     port1,number of threads
     port2,number of threads
     ```

     - **ProxyClientPassword**: Specifies an optional password for ClearanceDB Proxy.
Note
If you require a password for clients to access the database, it must be specified in the DBC file like this:

```
DATA SOURCE = <proxy_name>,<port>/<password>
```

**CONNECT_DATA**
Specifies the Oracle connection information. Type the information according to the following syntax:

```
<connect_name>,<user>/<password>@<service_name>
```

**Note**
The value used for `<connect_name>` must match the value specified in the DBC file.

If you are using an encrypted password in an external file, type the information according to the following syntax:

```
<connect_name>,<user>@<service_name>
```

Use the `ClearanceDBUserPasswordFile` setting to specify the full path and file name for the file containing the encrypted Oracle user password.

**ClearanceDBUserPasswordFile**
Specifies the path to a file containing the Oracle user password.

**AdminProxyPassword**
Specifies an optional password used to stop a ClearanceDB Proxy via ClearanceDB Client. This password should be restricted to the proxy administrator only.

4. Save the `Clearance.cfg.global` file.

**Encrypt the Oracle user password**
You can use the Teamcenter `install` utility to encrypt the Oracle user password. The encrypted password is stored in an external file.

1. In an environment configured to run Teamcenter utilities, open a command prompt.
   For information about configuring an environment to run Teamcenter utilities, see *Manually configuring your environment for Teamcenter utilities* in the Utilities Reference.

2. At the command prompt, type:
   ```
   install -encryptpwo -f=<path>
   ```
   where
   ```
   path
   ```
   is the full path and file name for the password file that you want to generate.
The `install` utility displays the following message:

```
Please enter password:
```

3. Type the password and press Enter.

The utility displays the following message:

```
Please re-enter the password:
```

4. Type the password again and press Enter.

The password is encrypted and saved to the specified location.

5. Navigate to the ClearanceDB Work Area.

6. Open the `Clearance.cfgglobal` file or a `Clearance.cfgproxy` file in a text editor.

7. To specify for ClearanceDB to obtain the Oracle user password from the encrypted password file, locate the `ClearanceDBUserPasswordFile` setting, and type the full path and file name for the password file.

```
Example

ClearanceDBUserPasswordFile=C:\ClearanceDB_Work_Area\oracle_key
```

8. Save the file.

**Create a ClearanceDB Proxy**

Use the `create_proxy.pl` script to create a ClearanceDB Proxy.

1. Open a command prompt and navigate to the ClearanceDB Work Area.

2. From the command prompt, type:

```
create_proxy.pl <proxy_name>
```

A folder named after the new proxy is created in the Proxies directory. This directory contains a file called `Clearance.cfgproxy`, which includes the default proxy settings from the `Clearance.cfgglobal` file.

**Configure a ClearanceDB Proxy**

When you use the `create_proxy.pl` script to create a ClearanceDB Proxy, a directory for the proxy is created within the ClearanceDB Work Area. This directory contains a file called `Clearance.cfgproxy`, which is populated with the default proxy settings from the `Clearance.cfgglobal` file.
If necessary, you can adjust the settings in `Clearance.cfgproxy` to override the global defaults. You can also add additional entries from the `Clearance.cfgglobal` file if you want to modify them for your proxy.

1. In the **Proxies** directory within the ClearanceDB Work Area, navigate to the proxy folder, and then open the `Clearance.cfgproxy` file in a text editor.

2. (Windows) Modify the **StartupType** setting to control how the proxy starts. Choose from the following options:

   - **Automatic**
     - Specifies to start the proxy using a Windows service that automatically starts when the system boots.
   
   - **Manual**
     - Specifies to start the proxy using a Windows service that you can manually start from the proxy directory or from the Windows Service monitor.

   Warning
   
   If you want to run the proxy as a Windows service, do not modify the **LogFile** setting, which requires an absolute path for the service to function correctly.

3. If you want to verify the software environment when starting the proxy, set the **VerifyEnvironment** parameter to **True**.

   Note
   
   On Windows, the versions used for the operating system and the ClearanceDB Proxy should all have the same bit architecture (32-bit or 64-bit).

4. If you want to adjust the maximum size of the dynamic buffer used for the compression of data sent between ClearanceDB Proxy and ClearanceDB clients, set the **MaximumLargeMessageBufferSize** parameter to any value from 1 to 256.

   The value used to specify the buffer size is in mebibytes, The default value is 128 mebibytes, which is approximately 134 megabytes (1 meibyte equals approximately 1.048 megabytes).

5. If you want to include additional sections in the ClearanceDB Proxy log file, add the **LogFlags** entry.

   Warning
   
   Adding these sections may significantly affect system performance, especially if you add the **SOCKETS** section.
You can add these sections:

- **SECTIONS** – Obtaining and releasing of Oracle sessions. If LogFlags is not included in the Clearance.cfgproxy file, the SECTIONS section is included in the log file by default. To exclude this section, the LogFlags entry must be included in the Clearance.cfgproxy file without SECTIONS being listed.

- **SQL** – Calls from the proxy to the Oracle RDBMS.

- **PACKETS** – Socket packet receive and send events.

- **SOCKETS** – Events related to socket acceptances and closures by proxy worker threads.

**Note**
Due to the size of this section, it is recommended that you only include it if SECTIONS, SQL, and PACKETS are not included.

To include one section, specify it as follows:

**Example**

LogFlags=SESSIONS

To include multiple sections, separate them with a +, as follows:

**Note**

LogFlags=SESSIONS+SQL+PACKETS


**Run a ClearanceDB Proxy as a Windows service**

On Windows, you can run the ClearanceDB Proxy as a Windows service.

1. In the Proxies directory within the ClearanceDB Work Area, navigate to the folder of the proxy that you want to manage with a service, and then open the Clearance.cfgproxy file in a text editor.

**Warning**

Do not modify the LogFile setting, which requires an absolute path for the service to function correctly.

2. Modify the StartupType setting to specify how the service starts. Choose one of the following options:
Automatically specifies to start the proxy using a Windows service that automatically starts when the system boots.

**Manual**
Specifies to start the proxy using a Windows service that you can manually start from the proxy directory or from the Windows Service monitor.

3. Save the `Clearance.cfgproxy` file.

4. Right-click the `clearance.cfgproxy` file and choose Start Proxy.
   When the proxy starts for the first time, the service is created.

5. Reboot the system.

   **Note**
   If the StartupType option in the `Clearance.cfgproxy` file is set to Automatic, the service starts automatically.

6. Do any of the following:

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start the service from the proxy directory</td>
<td>Right-click the <code>Clearance.cfgproxy</code> file and choose Start Proxy.</td>
</tr>
<tr>
<td></td>
<td>Type:</td>
</tr>
<tr>
<td></td>
<td><code>sc start &lt;service name&gt;</code></td>
</tr>
<tr>
<td>Start the service from the command line</td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>You must provide the service name not the display name of the service.</td>
</tr>
<tr>
<td></td>
<td>To display a list of the services on your machine, type: `sc query</td>
</tr>
<tr>
<td></td>
<td>state= all. Scroll through the list until you find the entry for the</td>
</tr>
<tr>
<td></td>
<td>ClearanceDB Proxy service, which includes the service name.</td>
</tr>
<tr>
<td></td>
<td>Tip</td>
</tr>
<tr>
<td></td>
<td>For descriptions of all of the commands supported by the SC program,</td>
</tr>
<tr>
<td></td>
<td>type sc.</td>
</tr>
</tbody>
</table>
Start a ClearanceDB Proxy

You can start a ClearanceDB Proxy from the command prompt. On Windows, you can also start the proxy with a right-click shortcut associated with the Clearance.cfgproxy file.

1. In the Proxies directory within the ClearanceDB Work Area, navigate to the folder of the proxy that you want to start, and then open the Clearance.cfgproxy file in a text editor.

2. Ensure the StartupType option is set to Console.

3. Save the Clearance.cfgproxy file.

4. Do either of the following:

<table>
<thead>
<tr>
<th>To start the proxy</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>Right-click the clearance.cfgproxy file and choose Start Proxy.</td>
</tr>
</tbody>
</table>
| From the command prompt | Type \n start_proxy.pl <path_to_proxy_directory>\ clearance.cfgproxy \n
**Example**

```
start_proxy.pl C:\ClearanceDB_Work_Area\Proxies\Proxy1\ clearance.cfgproxy
```
If the proxy is able to connect to the specified database, the command prompt message will resemble the following:

Successfully connected to Oracle service [CLDB] using connect_data [CLDB].

**Ping a ClearanceDB Proxy**

You can ping a ClearanceDB Proxy to determine if the proxy is functioning normally.

- Do any of the following:

<table>
<thead>
<tr>
<th>To ping the proxy</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>Right-click a DBC file that references the proxy and choose <strong>Ping Proxy</strong>.</td>
</tr>
<tr>
<td></td>
<td>Type ping_proxy.pl &lt;path_to_DBC_file&gt;\DBC_file</td>
</tr>
<tr>
<td>From the command prompt</td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td>ping_proxy.pl</td>
</tr>
<tr>
<td></td>
<td>C:\ClearanceDB_Work_Area\global.dbc</td>
</tr>
<tr>
<td></td>
<td>ping_proxy.pl</td>
</tr>
<tr>
<td></td>
<td>C:\ClearanceDB_Work_Area\Proxies\Proxy1\proxy1.dbc</td>
</tr>
</tbody>
</table>

**Ping the database**

You can ping the ClearanceDB database to test the database connectivity.

- From the ClearanceDB Work Area, do any of the following:

<table>
<thead>
<tr>
<th>To ping the database</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>Right-click a DBC file and choose <strong>Ping Database</strong>.</td>
</tr>
<tr>
<td>To ping the database</td>
<td>Do this</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>From the command prompt</td>
<td>Type ping_database.pl &lt;path_to_DBC_file&gt;\DBC_file</td>
</tr>
<tr>
<td></td>
<td>Example ping_database.pl C:\ClearanceDB_Work_Area\global.dbc</td>
</tr>
<tr>
<td></td>
<td>ping_database.pl C:\ClearanceDB_Work_Area\Proxies\Proxy1\proxy1 dbc</td>
</tr>
</tbody>
</table>

**List products in the database**

You can display a list of the ClearanceDB products that exist in the database.

- From the ClearanceDB Work Area, do any of the following:

<table>
<thead>
<tr>
<th>To list products in the database</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>Right-click a DBC file and choose List All Products.</td>
</tr>
<tr>
<td>From the command prompt</td>
<td>Type list_all_products.pl &lt;path_to_DBC_file&gt;\DBC_file</td>
</tr>
<tr>
<td></td>
<td>Example list_all_products.pl C:\ClearanceDB_Work_Area\global.dbc</td>
</tr>
<tr>
<td></td>
<td>list_all_products.pl C:\ClearanceDB_Work_Area\Proxies\Proxy1\proxy1 dbc</td>
</tr>
</tbody>
</table>

**Stop a ClearanceDB Proxy**

You can stop a ClearanceDB Proxy from the command prompt. On Windows, you can also stop the proxy with a right-click shortcut associated with the Clearance.cfgproxy file.

1. From the ClearanceDB Work Area, do any of the following:
To stop the proxy | Do this
---|---
(Windows) From the shortcut menu | Right-click the clearance.cfgproxy file and choose Stop Proxy.

<table>
<thead>
<tr>
<th>From the command prompt</th>
<th>Type stop_proxy.pl &lt;path_to_proxy_directory&gt;&lt;proxy_configuration_file&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example stop_proxy.pl \C:\ClearanceDB_Work_Area\Proxies\Proxy1\clearance.cfgproxy</td>
</tr>
</tbody>
</table>

The command line output displays the following message:

```
This command will stop the proxy. Are you sure (y/n)?
```

2. Type y and press Enter.

The proxy stops.
Chapter 7: Setting up products for analysis

Setting up products for analysis

ClearanceDB products consist of the specific metadata configurations, rules, conditions, and zones associated with the 3D model that you want to analyze.

Creating ClearanceDB products for analysis consists of the following tasks:

- **Creating the ClearanceDB default product** — Before you can populate the database with new product information, create the ClearanceDB default product. The default product is essentially a global configuration that you can customize to match your unique analysis requirements and automatically apply to new products that you create.

- **Creating a new ClearanceDB product** — Create a ClearanceDB product for each model you want to analyze.
• **Creating the ALL product for issue inheritance** — More than one instance of the same violating part pairs may exist in your clearance results. Inheritance relationships can be defined in your database so that these issues are not always duplicated but instead are managed based on their level of similarity.

• **Using the product-specific configuration file** — Each ClearanceDB product has its own configuration file, which is used to specify the location of the data to be analyzed by the Clearance Calculator. You can also use product-specific configuration files to override any of the parameters from the `Clearance.cfgglobal` file.

• **Creating ClearanceDB product configurations** — Create a process configuration file for each of your products to control the analysis.

• **Specifying rules and conditions** — Create unique rules and conditions to define clearance requirements or exclusions and to control the scope of the analysis.

• **Defining clearance zones** — Create clearance zones to perform analysis based upon the location of parts and assemblies within the model.

### Configuring the ClearanceDB default product

#### Understanding the ClearanceDB default product

The ClearanceDB default product is essentially a global configuration that you can customize to match your unique analysis requirements and automatically apply to new products that you create.

To create the default product, do any of the following:

- Run the `create_ClearanceDB_working_dir.pl` script with the `-df` option.
- Run the `create_default_product.pl` script.
- Run the `create_product.pl` script with no default product (the default product must exist in the database to create new products so in this situation it will be created and merged with the database automatically).

### Create the ClearanceDB default product

1. Install the ClearanceDB Client or Clearance Calculator.

2. Open a command prompt and type

   ```
   create_default_product.pl <path>
   ```

   where

<table>
<thead>
<tr>
<th><strong>For this parameter</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>path</code></td>
<td>The path to the location on your machine where you want to create the default product.</td>
</tr>
</tbody>
</table>
Example

create_default_product.pl c:\

A new directory named **DEFAULT** is created at the specified location. The **DEFAULT** directory contains the following files:

- conditions.csvcildb
- configuration.csvcildb
- rules.csvcildb
- zones.csvcildb

3. Using a text editor, modify each configuration file to create the product configuration that you want to use in the creation of new products.

4. Update the database in one of the following ways:

<table>
<thead>
<tr>
<th>To merge configuration files</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the command prompt</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td>update_product.pl [&lt;product&gt;]</td>
</tr>
<tr>
<td></td>
<td>-uc:-ur:-ud:-uz:-ua [-df]</td>
</tr>
<tr>
<td></td>
<td>where</td>
</tr>
<tr>
<td></td>
<td>-uc — The option to upload the</td>
</tr>
<tr>
<td></td>
<td>configuration.csvcildb file.</td>
</tr>
<tr>
<td></td>
<td>-ur — The option to upload the</td>
</tr>
<tr>
<td></td>
<td>rules.csvcildb and conditions.csvcildb</td>
</tr>
<tr>
<td></td>
<td>files.</td>
</tr>
<tr>
<td></td>
<td>-uz — The option to upload the</td>
</tr>
<tr>
<td></td>
<td>zones.csvcildb.</td>
</tr>
<tr>
<td></td>
<td>-ua — The option to upload all of</td>
</tr>
<tr>
<td></td>
<td>the configuration files.</td>
</tr>
</tbody>
</table>

Example

update_product.pl fishing_reel -ua

(Windows) From the shortcut menu

In the product directory, right-click any configuration file and choose **Upload to ClearanceDB**.
Create a new ClearanceDB product

Create a new ClearanceDB product, both locally in your ClearanceDB Work Area and in the database.

1. Open a command prompt and navigate to the ClearanceDB Work Area.

   **Note**
   You must create new products from the ClearanceDB Work Area.

2. From the command prompt, type

   ```
   create_product.pl <name> [-uc:-ur:-uz:-ua] [-ds <datasource> -cd <connection name>]
   ```

   where

<table>
<thead>
<tr>
<th>This parameter</th>
<th>Does this</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the name of the ClearanceDB product. The ClearanceDB product name must consist of the name of the top level assembly node from your product data. If the name contains spaces, enclose it in quotation marks. For example, “garage door opener”.</td>
</tr>
</tbody>
</table>

   If you are working with local, unmanaged data, open the 3D model in Mockup to obtain the name of the top level assembly node.

   ![Top level assembly node in Mockup](image-url)

   **Top level assembly node in Mockup**

   If you are working with data from Teamcenter, send the item revision to Structure Manager to obtain the name of the top level assembly node.
### This parameter
<table>
<thead>
<tr>
<th>Does this</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>BOM Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>331-61965 1/A;1-Roof Assembly (View)</td>
</tr>
<tr>
<td>331-61872 1/A;1 (View)</td>
</tr>
<tr>
<td>331-30429 1/A;1</td>
</tr>
<tr>
<td>331-32801 2/A;1 (View)</td>
</tr>
<tr>
<td>826-10982 1/A;1</td>
</tr>
<tr>
<td>826-10982 1/A;1</td>
</tr>
<tr>
<td>826-10982 1/A;1</td>
</tr>
<tr>
<td>826-10982 1/A;1</td>
</tr>
<tr>
<td>826-01706 1/A;1</td>
</tr>
<tr>
<td>826-01706 1/A;1</td>
</tr>
<tr>
<td>826-01708 1/A;1</td>
</tr>
</tbody>
</table>

**Top level assembly node in Structure Manager**

**Note**

A bomline name from Teamcenter has a space before (View), and ClearanceDB product names with spaces must be enclosed in quotation marks, as shown below.

```
“331–61965 1/A;1-Roof Assembly (View)"
```

If you are working within a Teamcenter multifield key environment, see Configuring ClearanceDB for multifield key data.

### -uc:-ur:-uz:-ua

Creates copies of the default product's configuration, rules, conditions, and zones information for you to customize for the product.

Choose from the following:

- **-uc** — Create a `configuration.csvclldb` file.
- **-ur** — Create `rules.csvclldb` and `conditions.csvclldb` files.
- **-uz** — Create a `zones.csvclldb` file.
- **-ua** — Create copies of all of the product configuration files.
### This parameter | Does this
--- | ---
**ds datasource** | Creates a DBC file for the product. For **datasource**, type the name and port of the system running the ClearanceDB Proxy.
**cd connection name** | Creates a DBC file for the product. For **connection name**, type the name used to identify the proxy connection.

#### Example
- Local product: `create_product.pl cottonpicker -ua -ds sfdmachine:7206 -cd CLDB`
- Teamcenter product: `create_product.pl "000092/A;1-cottonpicker (View)" -ua -ds sfdmachine:7206 -cd CLDB`

A new directory with the name you specified is created in the ClearanceDB Work Area. At a minimum, the directory contains the **Clearance.cfgproduct** file. The directory may also contain copies of the default product's configuration files or a DBC file, depending upon how you ran the script.

#### 3. (If you created copies of the default product's configuration files) Modify the configuration files as necessary for your product, and then update the database with your changes in one of the following ways:

<table>
<thead>
<tr>
<th>To merge configuration files</th>
<th>Do this</th>
</tr>
</thead>
</table>
| From the command prompt | Type \`update_product.pl [<product>] [-uc:-ur:-uz:-ua] [-df]\`

where
- **-uc** — The option to upload the `configuration.csvcldb` file.
- **-ur** — The option to upload the `rules.csvcldb` and `conditions.csvcldb` files.
- **-uz** — The option to upload the `zones.csvcldb`.
- **-ua** — The option to upload all of the configuration files.

**Example**
`update_product.pl cottonpicker -ua`
Configuring the ALL product for issue inheritance

Set up the ALL product for issue inheritance

More than one instance of the same violating part pair may exist in your clearance results. Inheritance relationships can be defined in your database so that these issues are not duplicated but instead are occurrences that reference the same shared object.

To configure issue inheritance, you define the ALL product. The configuration settings in this product apply to all products in the database.

1. Follow the steps to create a new ClearanceDB product, typing this in the command prompt:
   create_product.pl ALL

2. Manually create configuration.csvcldb in the ALL product directory and add the appropriate configuration attributes.
   - EAICL_IDENTICAL_MATCHING_ATTRIBUTE
   - EAICL_SIMILAR_MATCHING_ATTRIBUTE
   - EAICL_RESEMBLING_MATCHING_ATTRIBUTE
   - EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION

3. (Optional) Define groups for issue inheritance.

Define groups for the ALL product

When you configure issue inheritance you can define criteria groups, so that a match of all the attributes of any of the groups cause the result to be linked to the common clearance issue.

Additional clearance groups include a digit following the matching type. Group 1 has no digit, group 2 requires 1, group 3 requires 2, and so on.

In this example, if all parameters of group 1 match, or if all parameters of group 2 match, then the issues match.

```
# Identical matching attributes - Group 1
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Number
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,eaiclRESULT

# Identical matching attributes - Group 2
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,Part Name
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,Part Number
```
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,eaiclRELXFORM

Example configuration file for the ALL product

An example of a configuration.csvcldb file for the ALL product is shown here.

ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Number
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,eaiclRESULT
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,Part Name
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,Part Number
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE1,eaiclRELXFORM
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,eaiclRESULT
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,eaiclRELXFORM
ALL,EAICL_RESEMBLING_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_RESEMBLING_MATCHING_ATTRIBUTE,eaiclRESULT
ALL,EAICL_RESEMBLING_MATCHING_ATTRIBUTE,eaiclRELXFORM
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Owner
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Priority
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Status
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Comment
ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Owner
ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Priority
ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Status
ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Owner
ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Priority
ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Status
ALL,EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION,6

Deploying the issue inheritance mechanism

Enable the issue inheritance mechanism

1. From the ClearanceDB Work Area, create the ALL product:
   
   ```
   create_product.pl ALL
   ```

2. In the ALL product directory set up configuration.csvcldb with the appropriate configuration attributes. Take note which metadata attributes, such as Part Name, are referenced as matching attributes in configuration.csvcldb.

3. Modify all configuration.csvcldb files (except the one configuring the ALL product) so they configure all metadata matching attributes such as Part Name as EAICL_RELEVANT_PART_ATTRIBUTE.

4. Update all configuration.csvcldb files modified in step 3 in the database. From the respective product directories, type this:
   
   ```
   update_product.pl -uc
   ```
5. For all products with clearance issues, if they exist, most likely all products except ALL and DEFAULT, re-analyze products in the absolute mode and update the database with recalculated results.

Example
From SQL*Plus:
```sql
exec eaicl_p_reset_gc_attr (eaicl_p_product.product_id ('<product name>'));
```
From a command line:
```bash
analyze_product.pl
update_product.pl -ud
```

6. Upload `configuration.csvcldb` of the ALL product to the database. From the ALL product directory, type this:
```
update_product.pl -uc
```

**Disable the issue inheritance mechanism**

You can disable the existent issue inheritance mechanism by deleting the ALL product. Disabling the issue inheritance mechanism does not change any issue dispositions persisted in the database.

- From the ClearanceDB Work Area, type this:
  ```bash
  delete_product.pl ALL
  ```

**Modifying the issue inheritance mechanism**

While it is possible to modify the configuration of the issue inheritance mechanism, this may require a re-analysis of all the products with clearance and may result in substantial change to clearance issue dispositions.

If matching attributes are modified, it is recommended that you disable the issue inheritance mechanism and subsequently re-enable it following the steps to enable the issue inheritance mechanism (skipping the first step to create the ALL product). If matching attributes are not modified, you need only to upload the modified `configuration.csvcldb` of the ALL product to the database.

**Using the product-specific configuration file**

For the Clearance Calculator to analyze your product, you must specify the product in the `Clearance.cfgproduct` file. For data located outside of Teamcenter, you must specify the file name and full path to the product. For Teamcenter managed data, you must specify the item ID and revision of the top level assembly of your product. If your Teamcenter data is configured with multifield key data, you must specify the item key.
<table>
<thead>
<tr>
<th>For this option</th>
<th>Type this</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductInputFile</td>
<td>The file name and full path to the product.</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td></td>
<td>ProductInputFile = c:\fishing_reel.jt</td>
</tr>
<tr>
<td>ItemKey</td>
<td>The Teamcenter values (strings) of the multifield key properties for the item containing your product.</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td></td>
<td>ItemKey = item_id = 000092, MFK_strkey1 = val1</td>
</tr>
</tbody>
</table>

**Note**
This setting is applicable only to Teamcenter products configured with multifield key data. It corresponds to the \(--key\) argument for the Teamcenter **bomwriter** utility.

You can obtain the necessary multifield key strings using the Teamcenter **get_key_string** utility.
<table>
<thead>
<tr>
<th>For this option</th>
<th>Type this</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItemId</td>
<td>The Teamcenter item ID of the top level assembly of your product.</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td>ItemId=000092</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>This setting corresponds to the –item argument for the Teamcenter bomwriter utility.</td>
</tr>
<tr>
<td>ItemRevision</td>
<td>The Teamcenter revision of the top level assembly of your product.</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td>ItemRevision=A</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>This setting corresponds to the –rev argument for the Teamcenter bomwriter utility.</td>
</tr>
</tbody>
</table>

3. Save the file.

**ClearanceDB product configuration attributes**

**Overview of ClearanceDB product attributes**

ClearanceDB enables you to analyze multiple product configurations, including variants, control models, or revisions of a single product or product family. A product is identified by the name of the highest level node in the assembly (the root node). A product configuration is identified by configuration metadata at this top assembly node. If configuration metadata is not present at this node, the DEFAULT configuration values are used.

The ClearanceDB process configuration (defined in the configuration.csvcldb file) and rule model (defined in the rules.csvcldb and conditions.csvcldb files) are shared among product configurations. The ClearanceDB zone model is specific to a single configuration, defined in the "Configuration Name" field of the zones.csvcldb file.

The Clearance Calculator must be run for all product configurations separately. Clearance results are stored individually for each configuration. They are queried only for the configuration of the currently
opened top level assembly node as defined by the configuration metadata of that node. However, a user disposition assigned in one configuration will be applied to all remaining configurations where results are present. The match is based on the CADIDs of the parts associated with the results. Because of this, the uniqueness of the CADIDs in the product structure is essential.

The clearance management process and the clearance database behavior can be customized via a number of attributes. These attributes are defined by product by pairs of attribute name and attribute value. Some of the configuration attributes can have a single value, some can have multiple values. These configuration attributes are defined in the configuration.csvcldb file.

**ClearanceDB product attributes reference**

You can configure the ClearanceDB process for each of your products. A product configuration is defined in the configuration.csvcldb file, and then merged with the database.

Use the following attributes to define ClearanceDB product configurations:

<table>
<thead>
<tr>
<th>Use this attribute</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAICL_RELEVANT_PART_ATTRIBUTE</td>
<td>Store part metadata essential to the clearance management process in the database.</td>
</tr>
<tr>
<td>EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE</td>
<td>Use part metadata to indicate model design changes.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT</td>
<td>Turn on the initial clearance owner assignment.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_OWNER_CHOICE</td>
<td>Specify the metadata used for the initial clearance owner assignment.</td>
</tr>
<tr>
<td>INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION</td>
<td>Specify the metadata used to determine the initial clearance owner assignment.</td>
</tr>
<tr>
<td>INITIAL_CLEARANCE_OWNER_SEARCH_METHOD</td>
<td>Specify how product structure is searched for the metadata used for the initial clearance owner assignment.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_PRIORITY</td>
<td>Specify an initial clearance issue priority value.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_STATUS</td>
<td>Specify an initial clearance issue status value.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_COMMENT</td>
<td>Specify an initial clearance issue comment value.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_IMPORTANT</td>
<td>Specify an initial clearance issue important value.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_CHANGE_REQUIRED</td>
<td>Specify an initial clearance issue change required value.</td>
</tr>
<tr>
<td>EAICL_INITIAL_CLEARANCE_CHANGE_REFERENCE</td>
<td>Specify an initial clearance issue change reference value.</td>
</tr>
<tr>
<td>EAICL_CLEARANCE_CHANGE_DETECTION</td>
<td>Specify the change detection method.</td>
</tr>
<tr>
<td>EAICL_CLEARANCE_DISPOSITION_PERSISTENCE</td>
<td>Specify how you want to persist user dispositions of clearance owner, status, priority, comment and zone.</td>
</tr>
<tr>
<td>EAICL_PART_MATCHING_METHOD</td>
<td>Specify the part reconciliation method.</td>
</tr>
<tr>
<td>EAICL_PART_NAME_PROPER</td>
<td>Specify the part property which identifies metadata uniquely.</td>
</tr>
<tr>
<td>EAICL_RULES_PART_ATTRIBUTE</td>
<td>Turn on incremental clearance analysis.</td>
</tr>
<tr>
<td>Use this attribute</td>
<td>To do this</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EAICL_OCCURRENCE_KEY</td>
<td>Turn on part occurrence persistence by specified metadata.</td>
</tr>
<tr>
<td>EAICL_CHANGE_HISTORY_COLUMN</td>
<td>Specify the information to include in Issue Change History reports.</td>
</tr>
<tr>
<td>EAICL_PREVERSE_RESOLVED_ISSUES</td>
<td>Specify to track a history of ClearanceDB issues.</td>
</tr>
<tr>
<td>EAICL_PRODUCT_HISTORY_MAX_ENTRIES</td>
<td>Specify the maximum number of ClearanceDB history issues.</td>
</tr>
<tr>
<td>EAICL_PRODUCT_HISTORY_RETENTION_DAYS</td>
<td>Specify how long to store database events.</td>
</tr>
<tr>
<td>EAICL_PRODUCT_STRUCTURE_CHANGE_TEST</td>
<td>Track changes in the product structure size.</td>
</tr>
<tr>
<td>EAICL_PRODUCT_STRUCTURE_CHANGE_REJECTION_PERCENT</td>
<td>Specify the relative change in the product structure, between the old and new product snapshot, above which the clearance result upload will be rejected.</td>
</tr>
<tr>
<td>EAICL_RULEBASE_CHANGE_TEST</td>
<td>Specify to track changes in the size of the clearance requirement rulebase.</td>
</tr>
<tr>
<td>EAICL_RULEBASE_CHANGE_REJECTION_PERCENT</td>
<td>Define the relative change in the clearance requirement rulebase, between the old and new product snapshot, above which the clearance requirement rulebase upload will be rejected.</td>
</tr>
<tr>
<td>EAICL_TMP_OBJECT_CLEANUP_AGE_IN_DAYS</td>
<td>Clean up session-specific database objects older than a specified duration.</td>
</tr>
<tr>
<td>EAICL_IDENTICAL_MATCHING_ATTRIBUTE</td>
<td>When configuring issue inheritance, specify the criteria that reports issues as identical.</td>
</tr>
<tr>
<td>EAICL_SIMILAR_MATCHING_ATTRIBUTE</td>
<td>When configuring issue inheritance, specify the criteria that reports issues as similar.</td>
</tr>
<tr>
<td>EAICL_RESEMBLING_MATCHING_ATTRIBUTE</td>
<td>When configuring issue inheritance, specify the criteria that reports issues as resembling.</td>
</tr>
<tr>
<td>EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION</td>
<td>When configuring issue inheritance, specify the tolerance for numeric criteria in determining whether values are matching.</td>
</tr>
</tbody>
</table>

**EAICL_RELEVANT_PART_ATTRIBUTE**

Use the `EAICL_RELEVANT_PART_ATTRIBUTE` attribute to define the set of model metadata item names at the assembly or part level that are pertinent to the interference/clearance management process. You can have multiple instances of the `EAICL_RELEVANT_PART_ATTRIBUTE` attribute in a configuration.csvcldb file, each with a different value.
Chapter 7: Setting up products for analysis

Example

fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,Translation Date
fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,eaiclRULES
fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,LastModified
fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,OWNER
fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,LastModified
fishing_reel,EAICL_RELEVANT_PART_ATTRIBUTE,SafetyClass

Metadata typically stored in the database falls into three categories:

• Metadata indicating a possible change in part geometry.

Example

The value of the EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE attribute. For example, Translation Date.

• Metadata referenced by clearance requirement rules.

Example

For example, SafetyClass.

• Metadata used to determine the owner of an issue.

Example

The value of the EAICL_INITIAL_CLEARANCE_OWNER_CHOICE and EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION attributes. For example, OWNER for EAICL_INITIAL_CLEARANCE_OWNER_CHOICE and LastModified for EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION.

Note

For space and performance considerations, only the metadata defined by the EAICL_RELEVANT_PART_ATTRIBUTE attribute is saved in the database.

EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE

Use the EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE attribute to define the metadata the change of which indicates a possible change in part geometry. For example, Translation Date. This configuration attribute requires a unique value.

EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT

Use the EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT attribute to turn on the initial clearance owner assignment. Enable the option by including the
The `EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT` attribute in the `configuration.csvcldb` file and setting it to `ON`. The default value of `EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT` is `OFF`.

**Note**
When using the initial clearance owner assignment option, you also need to include the `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE`, `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION`, and `EAICL_INITIAL_CLEARANCE_OWNER_SEARCH_METHOD` attributes, along with appropriate settings, in the `configuration.csvcldb` file.

**EAICL_INITIAL_CLEARANCE_OWNER_CHOICE**

Use the `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE` attribute to specify the metadata you want to you for the initial clearance owner assignment. For example, `OWNER`.

**EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION**

Use the `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION` attribute to specify how the initial clearance owner is assigned. The `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE` value of the part with the larger `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION` value is assigned as the owner. For example, `LastModified`.

**EAICL_INITIAL_CLEARANCE_OWNER_SEARCH_METHOD**

Use the `EAICL_INITIAL_CLEARANCE_OWNER_SEARCH_METHOD` attribute to specify how the product structure is searched for the `EAICL_INITIAL_CLEARANCE_OWNER_CHOICE` metadata.

Valid settings for this attribute include the following:

- 1 - Only the parts are searched.
- 2 - The parts and possibly the (parent) assemblies are searched.
- 3 - Only the (parent) assemblies are searched.

If the specified metadata is not found it is assigned a value of "Unknown".

**EAICL_INITIAL_CLEARANCE_PRIORITY**

Define the clearance issue priority value initially assigned by the database. This value should match the configuration of Mockup installations.

**EAICL_INITIAL_CLEARANCE_STATUS**

Define the clearance issue status value initially assigned by the database. This value should match the configuration of Mockup installations.
**EAICL_INITIAL_CLEARANCECOMMENT**

Define the clearance issue Comment value initially assigned by the database. This value should match the configuration of Mockup installations.

**EAICL_INITIALCLEARANCEIMPORTANT**

Define the clearance issue Important value initially assigned by the database. This value should match the configuration of Mockup installations.

**EAICL_INITIALCLEARANCECHANGEREQUIRED**

Define the clearance issue Change Required value initially assigned by the database. This value should match the configuration of Mockup installations.

**EAICL_INITIALCLEARANCECHANGEREERENCE**

Define the clearance issue Change Reference value initially assigned by the database. This value should match the configuration of Mockup installations.

**EAICLCLEARANCECHANGEDTECTION**

Use the **EAICLCLEARANCECHANGEDTECTION** attribute to specify the change detection method. The supported values are below:

<table>
<thead>
<tr>
<th>Use this</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Inputs</td>
<td>Determines change on the basis of model inputs. This is the default value. These are the available model inputs:</td>
</tr>
<tr>
<td></td>
<td>1. Geometry: The geometry for one or both parts has changed.</td>
</tr>
<tr>
<td></td>
<td>2. Transform: A transform has been applied to one or both parts.</td>
</tr>
<tr>
<td></td>
<td>3. Requirement: The requirement for this pair has changed (possibly due to a rule change).</td>
</tr>
<tr>
<td></td>
<td>4. Variant condition. A variant condition affecting this pair has changed.</td>
</tr>
</tbody>
</table>
**Use this**  
*Analysis Outputs*

**To do this**
Determines change on the basis of outputs.

These are the available analysis outputs:

1. Result: The distance between the element pair (negative if they are penetrating).
2. Type: Contact, penetration, distance, thickness, or unknown.
3. Location: The geometric center of the penetrating region(s).
4. Orientation: The extraction vector for the result.
5. Intersection volume: The intersection volume.
6. Requirement: The requirement for this pair (as calculated from the rules).
7. Violation tolerance: The difference between the requirement and result.

When using `EAICL_CLEARANCE_CHANGE_DETECTION="Analysis Outputs"`, the following parameters are used to determine tolerances for detecting a change. Omitting a given tolerance, or specifying a value of -1, means that no test on this value will be performed, so any change of any size for a missing item will be completely ignored. Specifying a value of 0.0 for numeric tolerances means that there is no tolerance, so any deviation in a subsequent analysis will be detected as a change.

**Note**

You must specify at least one tolerance value or `eaicl_result_clearance_type_change` when using the *Analysis Outputs* option.

<table>
<thead>
<tr>
<th><strong>EAICL_RESULT_CHANGE_TOLERANCE</strong></th>
<th>Clearance result. Positive number. Expressed in model units.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EAICL_RESULT_CLEARANCE_TYPE_CHANGE</strong></td>
<td>Clearance type. YES or NO. YES means change in type is not detected as a change. NO means change in type IS detected as a change.</td>
</tr>
<tr>
<td><strong>EAICL_RESULT_POSITION_CHANGE_TOLERANCE</strong></td>
<td>Either part has moved. Positive number. Expressed in model units.</td>
</tr>
<tr>
<td><strong>EAICL_RESULT_ORIENTATION_CHANGE_TOLERANCE</strong></td>
<td>Either part has been rotated. Positive number. Expressed in degrees.</td>
</tr>
<tr>
<td><strong>EAICL_RESULT_INTERSECTION_VOLUME_CHANGE_TOLERANCE</strong></td>
<td>Volume of the intersection between parts. Positive number. Expressed in model units cubed.</td>
</tr>
<tr>
<td><strong>EAICL_RESULT_REQUIREMENT_CHANGE_TOLERANCE</strong></td>
<td>Change in requirement. Positive number. Expressed in model units.</td>
</tr>
</tbody>
</table>
### Use this

<table>
<thead>
<tr>
<th>EAICL_RESULT_VIOLATION_CHANGE_TOLERANCE</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difference between requirement and result. Positive number. Expressed in model units.</td>
</tr>
</tbody>
</table>

### Note

The default for all of these new tolerance parameters is “OFF”.

### EAICL_CLEARANCE_DISPOSITION_PERSISTENCE

Use the `EAICL_CLEARANCE_DISPOSITION_PERSISTENCE` attribute to control how the database persists user dispositions of clearance owner, status, priority, comment, and zone. Options are `eaiclCADID` and `eaiclCHANGE`.

If the value is set to `eaiclCADID` then the user dispositions are persisted as long as the issue defined by a relationship between two part instances, in turn defined by their CADIDs, exists. In other words, the changes in shape or location of the parts causing a clearance issue do not reset user dispositions to their initial values. However, a possible reassignment of parts to different subassemblies or disappearance and subsequent reemergence of the issue does cause the re-initialization of clearance owner, status, priority, comment, and zone.

If its value is set to `eaiclCHANGE` then the values of clearance owner, status, priority, comment, and zone are initialized each time there are changes in either shape or location of the parts causing a clearance issue.

### EAICL_PART_MATCHING_METHOD

Use the `EAICL_PART_MATCHING_METHOD` attribute to control the method of part reconciliation. Options are `eaiclCADID` and `eaiclSPATIAL`.

The parts of the old and new product snapshot are reconciled based strictly on CADIDs (part names and instance numbers) when `EAICL_PART_MATCHING_METHOD` is set to `eaiclCADID`.

The parts of the old and new product snapshot are reconciled based on proper part names and spatial considerations when `EAICL_PART_MATCHING_METHOD` is set to `eaiclSPATIAL`. This method provides better matches when the product structure changes. However, it may result in
mis-matches and thus in new clearance issues when same objects occupy the same space in the model.

**Note**

You also can use NGID data for part reconciliation. If `EAICL_OCCURRENCE_KEY` is set to `eaiclNGID`, ClearanceDB always uses NGIDs to reconcile product structure. The `EAICL_PART_MATCHING_METHOD` attribute is disregarded.

**EAICL_PART_NAME_PROPER**

Define the part property which identifies metadata uniquely. The eaiclPART value should be specified when only one version of a part is referenced by the product, the eaiclDATASTORE should be specified otherwise.

**EAICL_RULES_PART_ATTRIBUTE**

Enable incremental clearance analysis. This must be set to eaiclRULES to enable this functionality. Additionally, eaiclRULES must be configured as a relevant part attribute.

**EAICL_OCCURRENCE_KEY**

Use the `EAICL_OCCURRENCE_KEY` attribute to define how ClearanceDB stores references to product structure occurrences involved in clearance issues. Options are `eaiclCADID`, `eaiclNGID`, or any metadata that consists of unique identifiers. The default value of `EAICL_OCCURRENCE_KEY` is `eaiclCADID`.

<table>
<thead>
<tr>
<th>Use this</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eaiclCADID</code></td>
<td>Use CADIDs associated with your model to keep track of part occurrences in the ClearanceDB database.</td>
</tr>
<tr>
<td><code>eaiclNGID</code></td>
<td>Use NGIDs associated with your model to keep track of part occurrences in the ClearanceDB database.</td>
</tr>
</tbody>
</table>

**Note**

- If you set `EAICL_OCCURRENCE_KEY` to `eaiclNGID`, ClearanceDB always uses NGIDs to reconcile product structure, regardless of how the `EAICL_PART_MATCHING_METHOD` attribute is set.

- When adopting NGID strings for a product, if the product structure is already stored in the database the CADID strings of the part occurrences must be unique.

- For product structure that already exists in the database, change the `EAICL_OCCURRENCE_KEY` to `eaiclNGID` and then perform analysis. You can then upload results with NGID data to the database.
### Use this

| Metadata that consists of unique identifiers | Use metadata associated with your model to keep track of part occurrences in the ClearanceDB database. |

#### Example

__APPR_PATH_NODE or __PLM_ABSOCC_UID are used for models that originate from a Teamcenter repository.

### EAI_CL_CHANGE_HISTORY_COLUMN

Specify the information to include in Issue Change History reports. You can include any of the following information:

- Comment
- Important
- Owner
- Priority
- Requirement
- Result
- Db Rule Descriptions
- Db Rule Names

Note

These values are case and space-sensitive.
• Status
• Type
• Zone
• Change Reference
• Change Required

**EAICL_PRESERVE_RESOLVED_ISSUES**

Use the `EAICL_PRESERVE_RESOLVED_ISSUES` attribute to for greater control over the existing history behavior.

If set to No, resolved issues are deleted. If an issue recurs, it is treated as new, with a new clearance issue ID.

If set to Yes, resolved issues disappear, but are not deleted. If an issue recurs, it is included in reports with the prior issue ID.

Valid values are listed below:

<table>
<thead>
<tr>
<th>Use this</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Resolved issues are preserved.</td>
</tr>
<tr>
<td>No</td>
<td>Resolved issues are not preserved. This setting is the default behavior.</td>
</tr>
</tbody>
</table>

`EAICL_PRESERVE_RESOLVED_ISSUES` can be changed after the product is created, with the following results:

• Changing from No to Yes: Resolved issues will be preserved from the point in time the configuration is applied to the product in the database. Issues resolved prior to the change, along with any issue history, will not be available; thus, a newly recurring issue will be seen as new the first time it is loaded after the change in parameter. The issue history for unresolved issues will still be maintained according to specified `EAICL_CHANGE_HISTORY_COLUMN` values.

• Changing from Yes to No: Upon uploading of this parameter to the database, existing resolved issues and any history will be deleted. Future resolved issues and history will no longer be maintained.

The existing configuration parameter `EAICL_PRODUCT_HISTORY_RETENTION` (off by default) can be used to specify the maximum age, in days, for issue history (but not the issue itself) to be maintained, for both active and resolved issues. It can be omitted or set to -1, in which case no purging on basis of age will be performed. When changed and uploaded, it will purge issue history as needed.

**EAICL_PRODUCT_HISTORY_MAX_ENTRIES**

Use this attribute to specify the maximum number of product history entries to maintain for both active and resolved issues. If not included in the product configuration, or set to a value of -1, no deletion of history entries is performed. This attribute is used in conjunction with `EAICL_PRODUCT_HISTORY_RETENTION_DAYS`. 
Issue history for currently resolved issues is not available in Mockup reports.

A formerly resolved issue that recurs is viewed as an active issue, with its history available for reporting within Mockup just as with any other issue.

Previously-resolved issues that recur with EAICL_PRESERVE_RESOLVED_ISSUES set to TRUE generate a history entry indicating the restoration of the issue to the list of active issues. The history entries indicating this restoration are visible in Mockup history reports.

**EAICL_PRODUCT_HISTORY_RETENTION_DAYS**

Define the period of time the database events are persisted, in days. For example, a value of 365 equals a one year retention period.

This attribute is used in conjunction with **EAICL_PRODUCT_HISTORY_MAX_ENTRIES**.

**EAICL_PRODUCT_STRUCTURE_CHANGE_TEST**

Specify whether to track changes in the size of the product structure. Valid settings for this attribute are **ON** or **OFF**.

**EAICL_PRODUCT_STRUCTURE_CHANGE_REJECTION_PERCENT**

Define the relative change in the product structure, between the old and new product snapshot, above which the clearance result upload will be rejected. Use any value from 1-100.

**EAICL_RULEBASE_CHANGE_TEST**

Specify whether to track changes in the size of the clearance requirement rulebase. Valid settings for this attribute are **ON** or **OFF**.

**EAICL_RULEBASE_CHANGE_REJECTION_PERCENT**

Define the relative change in the product structure, between the old and new product snapshot, above which the clearance result upload will be rejected. Use any value from 1-100.

**EAICL_TMP_OBJECT_CLEANUP_AGE_IN_DAYS**

Define the amount of time before obsolete objects are automatically deleted from the database. ClearanceDB cleans up session-specific database objects older than this value to prevent objects that are “orphaned” by disconnected sessions from cluttering the database. You can use this parameter for a particular product or for the DEFAULT product.

The unit of time is **days**, and the default value is 0.01 (0.01 days = 14.4 minutes). In normal usage, the default value of 0.01 is sufficient. However, in situations involving high network latency and retrieval of large numbers of clearance results, an ORA-00954 error may result from the premature removal of a session’s objects. Increasing this value will prevent this error from occurring. Note that the specified value cannot be smaller than 0.01.

**EAICL_IDENTICAL_MATCHING_ATTRIBUTE**

Use the **EAICL_IDENTICAL_MATCHING_ATTRIBUTE** attribute in the **ALL** product to specify the criteria that reports issues as **identical**. Identical issues are linked to common clearance issues when
the inheritance process is executed in the database. The master issue (the one that the others are linked to) is the one that was most recently changed.

**Example**

```
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,Part Number
ALL,EAICL_IDENTICAL_MATCHING_ATTRIBUTE,eaiclRESULT
```

Use the **EAICL_SIMILAR_MATCHING_ATTRIBUTE** and **EAICL_RESEMBLING_MATCHING_ATTRIBUTE** attributes to define similar and resembling issues. Attributes in similar and resembling issues are copied.

Including any of these three attributes in the **ALL** product configuration, activates the issue inheritance mechanism that analyzes issues in the entire database for inheritance relationships.

You can configure which attributes are shared between identical issues, defining which attributes are included in the issue and which are included in the issue occurrence. To do this, add the **EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE** attribute to the configuration file.

**Example**

```
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Owner
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Priority
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Status
ALL,EAICL_IDENTICAL_ISSUE_ATTRIBUTE_SHARE,Comment
```

Use the **EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION** attribute to specify the tolerance for numeric criteria in determining whether values are matching.

**EAICL_SIMILAR_MATCHING_ATTRIBUTE**

Use the **EAICL_SIMILAR_MATCHING_ATTRIBUTE** attribute in the **ALL** product to specify the criteria that reports issues as similar. Similar issues share a common baseline issue, which is the one that was most recently changed. When the baseline issue is dispositioned, the information in the baseline issue is copied to the other issues in the same group. This copy operation happens only once, and future changes affect only that issue.

**Example**

```
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,Part Name
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,eaiclRESULT
ALL,EAICL_SIMILAR_MATCHING_ATTRIBUTE,eaiclRELXFORM
```

Use the **EAICL_IDENTICAL_MATCHING_ATTRIBUTE** and **EAICL_RESEMBLING_MATCHING_ATTRIBUTE** attributes to define identical and resembling issues.
Including any of these three attributes in the ALL product configuration, activates the issue inheritance mechanism that analyzes issues in the entire database for inheritance relationships.

You can configure which attributes are copied between similar issues. To do this, add the \texttt{EAICL\_SIMILAR\_ISSUE\_ATTRIBUTE\_COPY} attribute to the configuration file.

\textbf{Example}

\begin{verbatim}
ALL,EAICL\_SIMILAR\_ISSUE\_ATTRIBUTE\_COPY,Owner
ALL,EAICL\_SIMILAR\_ISSUE\_ATTRIBUTE\_COPY,Priority
ALL,EAICL\_SIMILAR\_ISSUE\_ATTRIBUTE\_COPY,Status
\end{verbatim}

Use the \texttt{EAICL\_MATCHING\_CRITERIA\_NUMERIC\_PRECISION} attribute to specify the tolerance for numeric criteria in determining whether values are matching.

\textbf{EAICL\_RESEMBLING\_MATCHING\_ATTRIBUTE}

Use the \texttt{EAICL\_RESEMBLING\_MATCHING\_ATTRIBUTE} attribute in the ALL product to specify the criteria that reports issues as \textit{resembling}. Resembling issues share a common baseline issue, which is the one that was most recently changed. When the baseline issue is dispositioned, the information in the baseline issue is copied to the other issues in the same group. This copy operation happens only once, and future changes affect only that issue.

Resembling issues are marked as not evaluated in the database. This typically means that you must investigate and evaluate the issue.

\textbf{Example}

\begin{verbatim}
ALL,EAICL\_RESEMBLING\_MATCHING\_ATTRIBUTE,Part Name
ALL,EAICL\_RESEMBLING\_MATCHING\_ATTRIBUTE,eaiclRESULT
\end{verbatim}

Use the \texttt{EAICL\_IDENTICAL\_MATCHING\_ATTRIBUTE} and \texttt{EAICL\_SIMILAR\_MATCHING\_ATTRIBUTE} attributes to define \textit{identical} and \textit{similar} issues.

Including any of these three attributes in the \texttt{ALL} product configuration, activates the issue inheritance mechanism that analyzes issues in the entire database for inheritance relationships.

You can configure which attributes are copied between resembling issues. To do this, add the \texttt{EAICL\_RESEMBLING\_ISSUE\_ATTRIBUTE\_COPY} attribute to the configuration file.

\textbf{Example}

\begin{verbatim}
ALL,EAICL\_RESEMBLING\_ISSUE\_ATTRIBUTE\_COPY,Owner
ALL,EAICL\_RESEMBLING\_ISSUE\_ATTRIBUTE\_COPY,Priority
ALL,EAICL\_RESEMBLING\_ISSUE\_ATTRIBUTE\_COPY,Status
\end{verbatim}

Use the \texttt{EAICL\_MATCHING\_CRITERIA\_NUMERIC\_PRECISION} attribute to specify the tolerance for numeric criteria in determining whether values are matching.
**EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION**

Use the `EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION` attribute to specify the tolerance for numeric criteria in determining whether values are matching.

For numeric criteria, such as a clearance result or clearance volume, a tolerance can be specified so that the values do not need to match *exactly* in order to be considered a match. To accomplish this goal, the use of *numeric precision*, rather than *numeric tolerance* is adopted. The precision is expressed in the number of significant digits. For example, for `ALL,EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION,3`, number 123,456,789 is treated as 1.23E8. If a tolerance value is not specified for a numerical attribute, the tolerance is 0, meaning an exact match is required.

The default value of `EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION` is the full (double) precision: 15.

**Merging product attributes with the database**

In order to create a ClearanceDB product and associate product attributes with it, you must merge the `configuration.csvcldb` file with the database.

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.

2. Do any of the following:

   - **From the command prompt**
     - Type `update_product.pl [<product>] [-uc] [-df]`  
     - **Example** 
       - `update_product.pl fishing_reel -uc`

   - **(Windows) From the shortcut menu**
     - In the product directory, right-click the `configuration.csvcldb` file and choose *Upload to ClearanceDB*.

**Example configuration file**

An example of a `configuration.csvcldb` file is shown below.

- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,Sequence`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,OWNER`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,LastModified`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,DesignationEnglish`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,FullPath`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,PartNumber`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,PartsType`
- `A-728AIRCRAFT,EAICL_RELEVANT_PART_ATTRIBUTE,eaiclRULES`
- `A-728AIRCRAFT,EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE,Sequence`
- `A-728AIRCRAFT,EAICL_CLEARANCE_DISPOSITION_PERSISTENCE,eaiclCADID`
- `A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_OWNER_CHOICE,OWNER`
Specifying rules and conditions

ClearanceDB rules and conditions enable you to limit analysis to specified data. Focusing the analysis on specific parts minimizes extraneous results, which in turn helps to optimize the clearance analysis process, especially when working with large sets of data.

Rules and conditions can help you control analysis in situations such as the following:

- To prevent a part from being analyzed because of its repetition within the model
- To prevent a subassembly from being analyzed because it is provided by a supplier
- To focus analysis on a part or assembly because of its position in the overall model
- To focus analysis upon parts or assemblies that match a set of conditions that you define

ClearanceDB rules

Use ClearanceDB rules to specify whether to include or exclude parts and assemblies from analysis, and to specify a clearance distance requirement.

For example, you can create a rule that limits analysis to parts from a set of data named `fishing_reel`, with a clearance distance requirement of 1 mm. Under this rule analysis occurs only for parts from the `fishing_reel` product, and results include all part pairs 1 mm apart or closer. To change this rule so that parts from the `fishing_reel` set of data are excluded from analysis, you would change the clearance distance requirement from 1 mm to `NONE`.

You can create and apply multiple rules to conduct analysis according to more complex criteria. For part pairs that fall under the scope of multiple rules, ClearanceDB conducts analysis according to the following:

- The largest clearance distance requirement is always used.
- The largest requirement along the path from the root node to the part is always used.
- If any of the rules specify to exclude either of the parts from analysis, the pair is not analyzed.
• Rules that exclude parts always take precedence over rules that specify to include particular parts sets.

• If no rule applies to the part pair, it is not analyzed.

For any occurrence pair, the largest clearance distance requirement of the two is always used.

For each occurrence in a pair, the largest requirement along the path from the root node to the part is always used.
If either occurrence of a pair is excluded, both occurrences are excluded.

Rules definition

ClearanceDB rules consist of the following components:

- **Rule Number** - A unique number used to associate the rule with conditions.

- **Product Name** - The name of the model to which you want the rule to apply. This is the name of the top-level node in the model assembly.

- **Rule Description** - Your description of the rule.

- **Requirement** - The clearance requirement value. For actual clearance distance requirements, this must be a numerical value. The unit of measurement is the model units value. For exclusion rules, use a value of **NONE**.

**Example**

```
Rule Number,Product Name,Rule Description,Requirement
100,fishing_reel,General Non-Interference,0
```

ClearanceDB conditions

Use ClearanceDB conditions to further define the scope of data to which ClearanceDB rules apply. You can use any part metadata associated with your geometry to define rule conditions.

A rule can have multiple conditions. Generally, as more conditions are applied to a rule, the scope of the rule becomes smaller, as will the results. However, rules are not required to have conditions. Rules without conditions apply to all part pairs in the model.

Conditions define two sets of parts for analysis according to the clearance distance requirement (or its absence) of the rule. You can define part sets in the following ways:

- Specify a single set of parts only. Under this approach, the other set of parts is implicit, and considered the entire model.
• Specify two unique sets of parts. Under this approach, there is a *left* set of parts, and a *right* set of parts. A rule applies to a part pair only if its conditions are met by the parts within both the left and right sets.

If a condition with Left node and a condition with Right node are used together on the same rule, the effect is similar to a logical AND operator.

**Conditions definition**

ClearanceDB conditions consist of the following components:

• **Rule Number** - The numerical identifier of the rule that you want to associate with the condition.

• **Condition Description** - Your description of the condition.

• **Condition Domain** - The part sets to which you want the condition to apply. Use any of the following values:
  
  o **Single Node** - Use this value when you want to specify a single set of parts only. The second set of parts is considered the entire model.

  o **Left Node** - Use this value to specify the *left* set of parts.

  o **Right Node** - Use this value to specify the *right* set of parts.
• **Search Type** - Where to search within the product assembly. Use any of the following values:
  - Node Only - Use this value to look for a condition match on part nodes only (individual parts within an assembly or subassembly, not the root nodes).
  - Node Parents - Use this value to look for a condition match on part nodes and also the parents of part nodes (individual parts within an assembly or subassembly, as well as root nodes).
  - Parents Only - Use this value to look for a condition match on root nodes only (the root of an assembly or subassembly, not the child parts).

The search type specifies where to search within the product assembly.

• **Match Type** - The applicability of the condition in the event of a match or a mismatch. Use either of the following values:
  - Y - Use this value if you want the condition to apply for matches.
  - N - Use this value if you want the condition to apply for mismatches.

• **Attribute Name** - Any metadata associated with the parts or assemblies of the product. Any value that you specify must be embedded in your model and usually originates from your CAD package. The specified value functions as a category, and is examined together with the Attribute Pattern, which functions as the category value.

  **Note**
  You can use the pseudo-attribute name `eaiclNAME` to refer to the node name itself.

• **Attribute Pattern** - A pattern to be matched with the wild card character %. This value is searched for in relation to the Attribute Name. For example, if you enter `DesignationEnglish` as the Attribute Name, and `%FLOOR%` as the Attribute Pattern, your results will include any parts with a metadata category named `DesignationEnglish` and a string value that contains the word `FLOOR`. 
**Example**

<table>
<thead>
<tr>
<th>Rule Number,Condition Description,Condition Domain,Search Type, Match Type,Attribute Name,Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,Match Part Name,Left Node,Node parents,Y,eaiclNAME,spool_assembly%</td>
</tr>
</tbody>
</table>

**Merging rules and conditions with the database**

In order to apply rules and conditions to your products, you must upload and merge the `rules.csvclldb` and `conditions.csvclldb` files with your database.

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.

2. Do either of the following:

<table>
<thead>
<tr>
<th>To merge rules and conditions</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>Right-click the <code>rules.csvclldb</code> file and choose <strong>Upload to ClearanceDB</strong>.</td>
</tr>
<tr>
<td>From the command prompt</td>
<td>Type <code>update_product.pl -ur</code></td>
</tr>
</tbody>
</table>

**Examples of rules and conditions**

**Examples of rules and conditions**

The following examples of rules and conditions are included as .csvclldb files with Teamcenter lifecycle visualization mockup example data.

<table>
<thead>
<tr>
<th>Example rule</th>
<th>Description</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>General non-interference</td>
<td>There is no associated condition for this rule, so it gets applied to the entire model.</td>
<td>General non-interference analysis on all the parts.</td>
</tr>
<tr>
<td>Single inclusion</td>
<td>Define a single condition on the general non-interference rule.</td>
<td>Analyze the <code>rotor_assembly</code> only.</td>
</tr>
<tr>
<td>Multiple inclusion (requirement override)</td>
<td>Define additional requirements for one or more subsets.</td>
<td>Analyze all parts with general non-interference requirement, while using alternate requirements for the <code>spool assembly</code> and the part <code>handle2</code>.</td>
</tr>
<tr>
<td>Single exclusion</td>
<td>Define an rule and condition to exclude a subset.</td>
<td>Analyze all parts with general non-interference requirement, but exclude the part <code>body</code>.</td>
</tr>
<tr>
<td>Example rule</td>
<td>Description</td>
<td>Use case</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Multiple exclusion</td>
<td>Define one or more rules and conditions to exclude one or more subsets.</td>
<td>Analyze all parts with general non-interference requirement, but exclude all parts and parts of parents that contain the word handle. Also exclude all parts whose parent's name is bail_wire_assembly.</td>
</tr>
<tr>
<td>Multiple inclusion with exclusion</td>
<td>Define additional requirements for one or more subsets, while excluding some subsets from that requirement.</td>
<td>Analyze all parts with general non-interference requirement, except for part rotor and any parts in the bail_wire_assembly. These parts will use a different requirement. Also exclude parts Part1 and Part5 (which are part of the bail wire assembly) from using this new requirement.</td>
</tr>
<tr>
<td>End item exclusion (no self-check)</td>
<td>Define multiple conditions to an exclusion rule to treat subassemblies like end items. Parts in the subassemblies will not be analyzed against each other.</td>
<td>Analyze all parts with general non-interference requirement. The subassemblies handle_assembly and bail_wire_assembly will each be considered an end item.</td>
</tr>
<tr>
<td>Multiple AND conditions</td>
<td>Define multiple conditions to a single rule to isolate the subset to be excluded.</td>
<td>Exclude all parts that are supplied by Company R AND contain the word Part.</td>
</tr>
</tbody>
</table>

### Loading the ClearanceDB example results

Each of the ClearanceDB examples includes ClearanceDB results saved in the .txt file format. You can examine the results in the viewer using the fishing_reel.jt model.

1. If necessary, start the Mockup installer, and then choose the following options to install the necessary example data:
   - Help and Miscellaneous Features→Example Files→3D Example Files
   - Help and Miscellaneous Features→Example Files→ClearanceDB Example Files

   **Note**
   
   The examples are installed in <Mockup installation directory>/Examples

2. Start Mockup and load the fishing_reel.jt model.

3. Choose **Analysis** tab→**Clearance** group→**Load Results**→**Load Results**.

4. In the **Load Clearance Results** dialog box, browse to the .txt results that you want to view, and then click **Open**.

   The results are displayed in the **Results Window**.
General non-interference

Rules: 1

Conditions: 0

Description: Set a single global distance requirement for all part occurrences in a product.

Example case: General non-interference analysis on all parts.
Working example

Set a single global requirement on the root node of the fishing reel.

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Product Name</th>
<th>Rule Description</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fishing_reel</td>
<td>&quot;Description: General non-interference. There is no associated condition (scope) for this rule, so it gets applied to the entire model.&quot;</td>
<td>0</td>
</tr>
</tbody>
</table>
Single inclusion

Rules: 1

Conditions: 1

Description: Set a single global distance requirement for a subset of the product.

Example case: General non-interference analysis on a subassembly.

![Diagram](image)
Working example

Set a single non-interference requirement.

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Product Name</th>
<th>Rule Description</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fishing_reel</td>
<td>&quot;Description: General non-Interference requirement&quot;</td>
<td>0</td>
</tr>
</tbody>
</table>

Set a condition on the rule to only match parts in the `rotor_assembly`.

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Condition Description</th>
<th>Condition Domain</th>
<th>Search Type</th>
<th>Match Type</th>
<th>Attribute Name</th>
<th>Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;Description: Only include the parts from the rotor assembly&quot;</td>
<td>Y</td>
<td>eaiclNAME</td>
<td>rotor_assembly%</td>
<td>Single Node</td>
<td>Parents Only</td>
</tr>
</tbody>
</table>
Multiple inclusion (requirement override)

**Rules:** 2 or more

**Conditions:** 1 or more

**Description:** Define one or more subsets in which the global requirement is overridden with additional requirements.

**Example case:** Analyze all parts with the general non-interference requirement, while using alternate requirements for the *spool* assembly and the part *handle2*.

![Diagram of multiple inclusion rule with subsets and overridden requirements.](image-url)
Working example

Set a single global requirement on the root node of the fishing reel.

Rule Number, Product Name, Rule Description, Requirement
1, fishing_reel, "Description: General non-interference (global scope, no condition)", 0

Set an override requirement of 0.75 for part handle_2.

Rule Number, Product Name, Rule Description, Requirement
2, fishing_reel, "Description: Requirement Override #1", 0.75

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
2, "Description: Apply the requirement from rule #2 to the part handle_2", Single Node, Node Only, Y, eaiiclNAME, handle_2%

Set an override requirement of 0.3 for subassembly spool_assembly.

Rule Number, Product Name, Rule Description, Requirement
3, fishing_reel, "Description: Requirement Override #2", 0.3

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
3, "Description: Apply the requirement from rule #3 to all parts in the spool_assembly", Single Node, Parents Only, Y, eaiiclNAME, spool_assembly%
Single exclusion

Rules: 2

Conditions: 1 or more

Description: Define a subset that will be excluded from analysis.

Example case: Analyze all parts with the general non-interference requirement, but exclude the part body.
Working example

Set a single global requirement on the root node of the fishing reel.

Rule Number,Product Name,Rule Description,Requirement
1,fishing_reel,"Description: General non-interference requirement (global scope, no condition)",0

Set an exclusion rule.

Rule Number,Product Name,Rule Description,Requirement
12,fishing_reel,"Description: Use NONE for exclusion instead of a number",NONE

Set a condition on this exclusion rule that specifies the part called body.

Rule Number,Condition Description,Condition Domain,Search Type,Match Type,Attribute Name,Attribute Pattern
12,"Description: Exclude the body part from analysis",Single Node,Node Only,Y,eaic1NAME,body%
**Multiple exclusion**

**Rules:** 2 or more

**Conditions:** 1 or more

**Description:** Define multiple subsets that will be excluded from analysis.

**Example case 1:** Analyze all parts with the general non-interference requirement, but exclude all parts and parts of parents that contain the word *handle*. Also exclude all parts with a parent name of *bail_wire_assembly*.

**Example case 2:** Analyze all parts with the general non-interference requirement, but exclude all part pairs where neither part is supplied by *Company H*. In other words, only include the part pairs in which *Company H* supplied one or both of the parts.
Working example 1

Set a single global requirement on the root node of the fishing reel.

Rule Number, Product Name, Rule Description, Requirement
1, fishing_reel, "Description: General non-interference (global scope, no condition)", 0

Set an exclusion rule and a condition on that rule to exclude all parts and parts of parents that contain the word handle.

Rule Number, Product Name, Rule Description, Requirement
12, fishing_reel, "Description: Use NONE for exclusion instead of a number.", NONE

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
12, "Description: Exclude parts & parents of parts where name=handle* from analysis" , Single Node, Node Parents, Y, eaiclNAME, handle%

Set an exclusion rule and a condition on that rule to exclude all parts with a parent name of bail_wire_assembly.

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
13, fishing_reel, "Description: Use NONE for exclusion instead of a number.", NONE

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
13, "Description: Exclude all children of bail_wire_assembly from analysis" , Single Node, Parents only, Y, eaiclNAME, bail_wire_assembly%

Exhibit 7-42
Working example 2

Set a single global requirement on the root node of the fishing reel.

Rule Number, Product Name, Rule Description, Requirement
1, fishing_reel, "Description: General non-interference requirement (global scope, no condition)", 0

Set an exclusion rule and two conditions on that rule to exclude all parts where neither supplier is Company H.

Rule Number, Product Name, Rule Description, Requirement
12, fishing_reel, "Description: Use NONE for exclusion instead of a number", NONE

Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern
12, "Description: Exclude parts on the left where Supplier is not equal to Company H", Left Node, Parents Only, N, Supplier, Company H
12, "Description: Exclude parts on the right where Supplier is not equal to Company H", Right Node, Parents Only, N, Supplier, Company H

Global:
R = 0

Part supplied by "Company H"
Multiple inclusion with comparison exclusion

**Rules:** 2 or more

**Conditions:** 2 or more

**Description:** Define one or more subsets in which the global requirement is overridden with additional requirements and exclude subsets from those requirements.

**Example case:** Analyze all parts with the general non-interference requirement, except for part rotor and any parts in the bail wire assembly. These parts will use a different requirement. Also exclude parts Part1 and Part5 (which are part of the bail wire assembly) from using this new requirement, because they are in contact with the rotor and are better handled by the general non-interference requirement.
Working example

Set a single global requirement on the root node of the fishing reel.

<table>
<thead>
<tr>
<th>Rule Number, Product Name, Rule Description, Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, fishing_reel, &quot;Description: General non-Interference requirement (global scope, no condition)&quot;, 0</td>
</tr>
</tbody>
</table>

Set an override requirement of 6.0 for part rotor and parts with a parent name of bail_wire_assembly.

<table>
<thead>
<tr>
<th>Rule Number, Product Name, Rule Description, Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, fishing_reel, &quot;Description: Requirement over-ride&quot;, 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, &quot;Description: Match parts on the left who's Name is rotor&quot;, Left Node, Node Only, Y, eaiclNAME, rotor%</td>
</tr>
<tr>
<td>10, &quot;Description: Match parts on the right who's parent Name is bail_wire_assembly&quot;, Right Node, Parents Only, Y, eaiclNAME, bail_wire_assembly%</td>
</tr>
</tbody>
</table>

Exclude Part1 and Part5 from analysis with the new requirement of 6.0.

<table>
<thead>
<tr>
<th>Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, &quot;Description: Don't match Part1 on the right&quot;, Right Node, Node Only, N, eaiclNAME, Part1%</td>
</tr>
<tr>
<td>10, &quot;Description: Don't match Part5 on the right&quot;, Right Node, Node Only, N, eaiclNAME, Part5%</td>
</tr>
</tbody>
</table>
End item exclusion (self check)

**Rules:** 2 or more

**Conditions:** 2 or more

**Description:** Define one or more subsets to be evaluated as an end item. Parts in the subsets will not be analyzed against each other.

**Example case:** Analyze all parts with the general non-interference requirement. The subassemblies *handle_assembly* and *bail_wire_assembly* will each be considered an end item.
Working example

Set a single global requirement on the root node of the fishing reel.

<table>
<thead>
<tr>
<th>Rule Number,Product Name,Rule Description,Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,fishing_reel,&quot;Description: General non-Interference (global scope, no condition)&quot;&quot;,0</td>
</tr>
</tbody>
</table>

Set an exclusion rule and a condition on that rule to exclude all part pairs where the parent name of both parts is `handle_assembly`.

<table>
<thead>
<tr>
<th>Rule Number,Product Name,Rule Description,Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>21,fishing_reel,&quot;Description: Use NONE for exclusion instead of a number&quot;,NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Number,Condition Description,Condition Domain,Search Type,Match Type,Attribute Name,Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>21,&quot;Description: Exclude children of handle_assembly (left)&quot;,Left Node,Parents only,Y,eaiclNAME,handle_assembly%</td>
</tr>
<tr>
<td>21,&quot;Description: Exclude children of handle_assembly (right)&quot;,Right Node,Parents only,Y,eaiclNAME,handle_assembly%</td>
</tr>
</tbody>
</table>

Set an exclusion rule and a condition on that rule to exclude all part pairs where the parent name of both parts is `bail_wire_assembly`.

<table>
<thead>
<tr>
<th>Rule Number,Product Name,Rule Description,Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>22,fishing_reel,&quot;Description: Use NONE for exclusion instead of a number&quot;,NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Number,Condition Description,Condition Domain,Search Type,Match Type,Attribute Name,Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>22,&quot;Description: Exclude children of bail_wire_assembly (left)&quot;,Left Node,Parents only,Y,eaiclNAME,bail_wire_assembly%</td>
</tr>
<tr>
<td>22,&quot;Description: Exclude children of bail_wire_assembly (right)&quot;,Right Node,Parents only,Y,eaiclNAME,bail_wire_assembly%</td>
</tr>
</tbody>
</table>
Chapter 7: Setting up products for analysis

Global:
\[ R = 0 \]

Exclude analysis between these parts

Exclude analysis between these parts
Multiple AND conditions

Rules: 1 or more

Conditions: 2 or more

Description: Use multiple conditions to define one or more subsets that will be excluded from analysis. All conditions have to be true in order for the exclusion to take place.

Example case: Analyze all parts with the general non-interference requirement.Exclude all parts that are supplied by Company R AND contain the word Part.
Working example

Set a single global requirement on the root node of the fishing reel.

<table>
<thead>
<tr>
<th>Rule Number, Product Name, Rule Description, Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, fishing_reel, &quot;Description: General non-interference (global scope, no condition)&quot;, 0</td>
</tr>
</tbody>
</table>

Set an exclusion rule and two conditions on that rule to exclude all parts supplied by Company R AND that contain the word Part in the eaiclNAME.

<table>
<thead>
<tr>
<th>Rule Number, Product Name, Rule Description, Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>21, fishing_reel, &quot;Description: Use NONE for exclusion instead of a number&quot;, NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Number, Condition Description, Condition Domain, Search Type, Match Type, Attribute Name, Attribute Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>21, &quot;Description: Include parts where supplier is Company R&quot;, Single Node, Parents Only, Y, Supplier, Company R</td>
</tr>
<tr>
<td>21, &quot;Description: Include parts who's name starts with Part^*&quot;, Single Node, Node Only, Y, eaiclNAME, Part%</td>
</tr>
</tbody>
</table>

Defining clearance zones

Defining clearance zones

ClearanceDB enables you to create interference/clearance zones. Clearance zones allow you to perform analysis based upon the location of parts and assemblies within the model, rather than by attribute information.
Understanding clearance zones

A zone is defined by a bounding box that consists of two 3D points, X1, Y1, Z1, and X2, Y2, Z2. An interference/clearance issue belongs to a zone under the following conditions:

- In the case of penetration, if the point that corresponds to the largest penetration is within the zone bounding box.
- In the case of contact, if the point that represents the average of all the points of contact is within the zone bounding box.
- In the case of distance, if the minimum distance midpoint between the two closest points is within the zone bounding box.

In the case of overlapping zones, the zone sequence determines the search order. The first zone found that contains an issue is assigned the issue.

If no named zone can be assigned to a clearance issue because of the absence of a definition or because of geometric concerns, the value of "Unknown" is assigned to the clearance issue.

Zones definition

ClearanceDB zones consist of the following content:

- Product name - The name of the model to which you want the rule to apply. This is the name of the top-level node in an assembly.
- Configuration name - A name that you provide that describes the product configuration.
- Seq - The position in the zone sequence. The zone sequence determines the search order.
- Zone name - A name that you provide for the zone.
- X1 - The X axis coordinate that defines the beginning of the zone.
- Y1 - The Y axis coordinate that defines the beginning of the zone.
- Z1 - The Z axis coordinate that defines the beginning of the zone.
- X2 - The X axis coordinate that defines the end of the zone.
- Y2 - The Y axis coordinate that defines the end of the zone.
- Z2 - The Z axis coordinate that defines the end of the zone.

Merging zones with the database

In order to apply zones to your products, you must upload and merge the zones.csvcldb file with your database.

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.
2. Do any of the following:
To merge product attributes | Do this
--- | ---
From the command prompt | Type
update_product.pl [<product>] [-uz] [-df]

(Windows) From the shortcut menu | In the product directory, right-click the zones.csvclidb file and choose Upload to ClearanceDB.

**Example zones file**

An example of a zones file is shown below.

```
A-728AIRCRAFT,DEFAULT,1,CTR_SECTION_UNPRESSURIZED,9000,-1630,1500,18500,1630,3400
A-728AIRCRAFT,DEFAULT,2,AFT_FUSELAGE_UNPRESSURIZED,21819,-1500,2750,28900,1500,5700
A-728AIRCRAFT,DEFAULT,3,PYLON_LHS,10000,-3500,2000,16900,-4500,3300
A-728AIRCRAFT,DEFAULT,4,PYLON_RHS,10000,3500,2000,16900,4500,3300
A-728AIRCRAFT,DEFAULT,5,WING_LHS,11000,0,2000,18500,13500,4500
A-728AIRCRAFT,DEFAULT,6,ATA531_FWD_COCKPIT,2480,-1735,2265,5642.8,1735,5735
A-728AIRCRAFT,DEFAULT,7,ATA532_FWD_FUSELAGE,5642.8,-1735,2265,10570.4,1735,5735
A-728AIRCRAFT,DEFAULT,8,ATA533_CENTRE_FUSELAGE,10570.4,-1735,2265,16299,1735,5735
A-728AIRCRAFT,DEFAULT,9,ATA534_REAR_FUSELAGE,16299,-1735,2265,22056.4,1735,5735
A-728AIRCRAFT,DEFAULT,10,ATA535_REAR_FUSELAGE,22056.4,-1735,2265,24514,1735,5735
A-728AIRCRAFT,DEFAULT,11,UNPOS_PARTS,500,500,500,-500,-500,-500
```
Chapter 8: Performing analysis

Accuracy of clearance calculations

Clearance analysis is the process of considering a model in terms of the spatial relationships between its parts or part groups. The location of each part or group within the model is compared to the location of every other part or group within the model. Any part or group pairs that are closer to one another than the specified clearance distance is identified as a clearance issue.

You can define inheritance relationships in your database so that issues that are instances of the same violating part pairs are not always duplicated in the clearance results but instead are managed based on their level of similarity.

During clearance analysis, first the bounding boxes of each part or group pair are tested. If the bounding boxes do not intersect and the minimum distance between them is greater than the specified clearance distance value, the part or group pairs are not identified as clearance issues and do not appear in the Results list.

For each part or group pair with bounding boxes that do intersect, or where the minimum distance between them is less than the clearance distance, the tessellated geometry of the two parts is then tested. If the geometry does not intersect and the minimum distance between the parts or groups is greater than the contact tolerance value, the two parts or groups are not identified as clearance issues and do not appear in the Results list.

If the tessellated geometry intersects or if the minimum distance between the parts or groups is less than the contact tolerance value, the NURBS data (if it is available in your model) of the two parts or groups is then tested. The NURBS test determines the exact points of contact between the two parts or groups.

Once points of contact are identified, the tessellated geometry and the contact tolerance value you have specified are used to determine whether the parts or groups are in contact only or whether one part or group penetrates the other. If one part penetrates the other to a depth greater than the contact tolerance value, the penetration depth is calculated for the part pair. The penetration depth is calculated as the distance the two penetrating parts need to be moved away from each other to avoid the penetration.

Example

Because of the way clearance is calculated, some clearance checks may not yield the results you expect. For example, if a clearance check results in a part pair that consists of a cylinder and the hole with which it is designed to mate, you may expect the penetration depth to describe the difference in the diameters of the cylinder and the hole when in actuality the penetration depth describes the depth along the length of the cylinder.

Penetration depth calculations are always an approximation, even when using precise data. A precise depth calculation requires a manually specified extraction vector, which is the direction used to separate a penetrating part pair. Without an extraction vector, an infinite number of solutions are
possible for each penetration. Since user input is not possible during batch calculations, an algorithm estimates the extraction vector for each penetration. This estimation is the direction in which the parts move the shortest distance in order to eliminate the penetration.

**Note**

Penetration depth estimations may vary slightly on different platforms due to factors such as different CPUs, different operating systems, and 32-bit versus 64-bit architectures. The penetration depths provided by Teamcenter lifecycle visualization are intended to provide a reasonable approximation of the penetrations. Only the designers of the geometry know how to best resolve each penetration, so they need to use the penetration depth as a guide to help them quickly locate and resolve clearance issues.

### Enabling rules and conditions

For the Clearance Calculator to access rules and conditions stored in the ClearanceDB database, you must enable the ClearanceDB rules and results settings in the `Clearance.cfgglobal` file.

To edit the `Clearance.cfgglobal` file so that the Clearance Calculator uses rules and conditions stored in the database, remove the number character (++) before the `RulesObject` and `ResultsObject` parameters as shown below:

```plaintext
#RulesObject = Universal Clearance Requirement |
RulesObject = Requirement Rules from ClearanceDB Server |
#ResultsObject = General Clearance Results |
ResultsObject = ClearanceDB Results |
```

### Clearance elements and CAD components

Elements and CAD components are the parts and branches of product structure recognized for clearance analysis in Teamcenter lifecycle visualization. They are used to control the granularity of the analysis.

There are three element types: parts, leaf components, and end items. Any combination of these elements can exist in the product structure for use during a clearance analysis. However, the part element type is always turned on.

If an element contains other elements within it, no clearance issues between those elements are reported in the clearance results list. Also, any two leaf components or end items can have many clashes between them, but only the single, most severe result between any two elements is reported.

The items that appear in the clearance results list are always elements. A CAD component is essentially an end item that allows any number of clearance results between itself and other nodes. For this reason, a CAD component is not considered an element.
<table>
<thead>
<tr>
<th>Element type</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td>The lowest level elements in the clearance analysis are parts. Analysis is always conducted between part elements, unless the parts are overridden by other element types (leaf components and end items) or CAD components.</td>
</tr>
<tr>
<td>Results</td>
<td>Only issues between parts are displayed in the clearance results, even if leaf components, end items, or CAD components exist in the structure.</td>
</tr>
<tr>
<td>Leaf component</td>
<td></td>
</tr>
</tbody>
</table>
### Element type

<table>
<thead>
<tr>
<th>Element type</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>The lowest level elements in the clearance analysis are parts, unless they are overridden by leaf components. In that case, all of the parts underneath the leaf component node are considered to be part of the same, monolithic element.</td>
</tr>
<tr>
<td>Results</td>
<td>Only the single, most severe result between each leaf component and other elements is displayed. That result appears in the clearance results list as the leaf component itself (not the parts within the leaf component).</td>
</tr>
<tr>
<td>End item</td>
<td>Only the single, most severe result between each end item and other elements is displayed. That result appears in the list as the end item itself (not the parts within the end-item).</td>
</tr>
</tbody>
</table>

### Note

End items are branches of product structure where the root node contains the metadata __PLM_END_ITEM=True. |
<table>
<thead>
<tr>
<th>Element type</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf components and end items</td>
<td></td>
</tr>
</tbody>
</table>

**Analysis**

The lowest level elements in the clearance analysis are parts, unless they are overridden by a leaf component or end item. In that case, all of the parts underneath the “parent” node are considered to be part of the same, monolithic element.

**Results**

No issues are reported between elements within the leaf component or end item. Only the single, most severe result between each leaf component or end item and other elements is displayed. That result appears in the list as the root item (not the parts within the root item).
Element type | Behavior
--- | ---
CAD components | Parts

### Analysis

The lowest level elements in the clearance analysis are parts, unless they are overridden by a CAD component. In that case, all of the parts underneath the CAD component node are considered to be part of the same monolithic component.

### Results

No issues are reported between elements within the CAD component. However, any number of issues may be reported between elements within the CAD component and other elements in the model. Those results appear in the list as the elements themselves (not the CAD component).

### Note

CAD components are branches of product structure where the root node contains the metadata `__PLM_CADCOMPONENT=True`.

---

**Configure the Clearance Calculator**

Use the *Section 4: Clearance Analysis configuration* section of the `Clearance.cfgglobal` file to specify options for clearance analysis. The settings from this file are applied to all of the clearance products in the ClearanceDB Work Area, unless the settings are manually copied into the `Clearance.cfgproduct` for a specific product. Any settings specified in the `Clearance.cfgproduct` take precedence over the `Clearance.cfgglobal` file.
Note

- All values for all time-based options are in seconds, with the exception of `MonitorResolution`, which is in milliseconds.

- If `RecordPartPairAdditionEvents` and `RecordClearanceResultAdditionEvents` are set to 0 in the `Clearance.cfgglobal` file, only major events such as process restarts or analysis failures are reported in the log file.

1. Navigate to the ClearanceDB Work Area:

2. Using a text editor, open the `Clearance.cfgglobal` file.

3. Change any of the following settings in the Section 4: Clearance Analysis configuration section of the file to configure the analysis:
ClearanceExeOptions
- **d <value>** — Specifies the clearance distance, in model units.
- **n** — Checks clearance using NURBS data, if present in your model.
- **c** — Checks for points of contact and penetration.
- **p** — (Valid with -c option only) Calculates the depth of penetrations.
- **t <value>** — (Valid with -c and -p options only) Specifies the contact tolerance.
- **e** — Automatically sends e-mail messages to owners, as specified in your Clearance Manager preferences.
- **s** — Shows only error messages as the Clearance Calculator runs.
- **r** — Specifies the maximum distance requirement, in model units.
- **q <dbc>** — Specifies to perform analysis against the ClearanceDB database. You must specify a DBC file to use this option.

**Note**
You only need to use this option if manually running the Clearance Calculator from the command line, without using the `analyze_product.pl` or `analyze_managed_product.pl` Perl script.

- **Q <dbc>** — Specifies to perform a ClearanceDB query. A DBC file is optional.

**Note**
You only need to use this option if manually running the Clearance Calculator from the command line, without using the `analyze_product.pl` or `analyze_managed_product.pl` Perl script.

- **b <value>** — Specify the directory to use for fault-tolerant restarts.
- **m <crl>** — Includes material thickness in the clearance check. Specify the `crl` option for more accurate results.
- **l** — Applies the default layer filter when checking clearance.
- **i** — Generates 2D images of the element pairs involved in clearance violations.

**Note**
To generate 2D images of clearance issues, you must set the `ResultsObject` option to **General Clearance Results** in the `Clearance.cfgglobal` file.

- **f <output filename>** — Saves clearance results as a comma-delimited, quote-enclosed, ASCII text file. By default, the results file is named `Results.txt`. To give a results file a different name, type the name after the `-f` parameter at the command prompt.

**ElementType**
The clearance element type. Valid options are:

- **Part**
- **Component**
- **EndItem**
EndItemComponent
The default element type is Part.

RecognizeCADComponents
Specifies whether or not CAD components are recognized during analysis. Valid values are 0 (off) or 1 (on). The default value is 0.

NumberOfProcesses
The number of CPU processes to use for clearance analysis. This option requires a machine with multiple processors or multiple cores. The default value is 1.

**Note**
The maximum size a process can have may be limited by the operating system, the system administrator, or otherwise.

NumberOfThreads
The number of threads to use for clearance analysis. The default value is the number of cores on the machine. Whenever the number of threads is greater than one, the number of processes is automatically set to one, regardless of the value in the configuration file.

**Note**
The maximum size a process can have may be limited by the operating system, the system administrator, or otherwise.

AbsMemLimit
The amount of physical RAM in megabytes to use for batch mode analysis. When the memory limit is exceeded, data is unloaded until memory usage falls below the specified amount. If both AbsMemLimit and RelMemLimit options are used, the lesser effective value is chosen.

RelMemLimit
The percentage of physical RAM to use for Batch Mode analysis. When the memory limit is exceeded, data is unloaded until memory usage falls below the specified percentage. The default percentage is 100 percent. You also can force the application to use the virtual memory available on your machine. You can set a memory limit of up to 200 percent of your physical RAM.

**Note**
If both AbsMemLimit and RelMemLimit options are used, the lesser effective value is chosen.

RulesObject
How to obtain the clearance requirement. Valid options are:

Universal Clearance Requirement
Requirement Rules from ClearanceDB Server
Variant Analysis

Note
If you want to obtain clearance requirements from your organization's ClearanceDB server, specify Requirement Rules from ClearanceDB Server.

ResultsObject
How to handle the clearance results. Valid options are:

General Clearance Results
ClearanceDB Results

Note
To upload results to the server, specify ClearanceDB Results.
You can list the ResultsObject option twice to specify both General Clearance Results and ClearanceDB Results.

Attribute
Specifications to filter clearance results according to attribute metadata associated with your product structure. Specify each attribute you want to use, followed by the assembly search method. Each attribute specification requires a separate entry in the Clearance.cfgglobal or Clearance.cfgproduct file.

Attribute entries must use the following syntax:
Attribute=[attribute_name], [order_of_search_method]
You can search the assembly according to the following criteria:

Part Only
Part Parent Only
Part and Parents (Part First)
Part and Parents (Root First)
Parents Only (Part Parent First)
Parents Only (Root First)

To work with Teamcenter managed products, specify this attribute:
Attribute=__PLM_ABSOCC_UID,Part and Parents (Part First)

The following entries are examples:
Attribute=Translation Date,Part Only
Attribute=End Item,Part Only
Attribute=End Item,Part Parent Only
Attribute=End Item,Part and Parents (Part First)
Attribute=End Item,Part and Parents (Root First)
Attribute=End Item,Parents Only (Part Parent First) | Attribute=End Item,Parents Only (Root First) |

EnabledMemAndThroughputLog
Specifies whether to create a log file in the batch clearance directory to log the memory usage and the element pair throughput at set intervals. Valid options are 0 for no log file or 1 to create the log file.

MemAndThroughputLogInterval
Number of seconds to specify the interval between entries if EnabledMemAndThroughputLog is set to 1.

GracefulShutdownTimeout
The time duration the operating system is given to abort the worker process. The specified value is in seconds. The default value is 60.

MaxFailures
The maximum number of failures per analysis run. The default value is 1000000.

MaxRelFailures
The maximum relative number of failures per run in respect to all element pairs in the test bed. The specified value is a percentage. The default value is 100.

ProgressDirectory
The directory where intermediate files are placed, including the log file.

OverallTimeout
The maximum duration of the analysis. The default value is 604800, which equals one week.

PairTimeout
The maximum duration of the analysis of a single element pair. If the analysis exceeds the specified time the analysis fails. The default value is 3600, which equals one hour.

RestartAfterPairTimeout
Specifies whether to restart the clearance analysis if an analysis of a single element pair times out (exceeds the PairTimeout duration). Valid values are 0 for no restart or 1 to restart. If the clearance analysis is not restarted, the pair is included as an undefined result in the log file.

MonitorResolution
The maximum frequency of the progress monitoring to standard output. The default value is 60000, which equals one hour.

MaxPairAttempts
The maximum number of possible attempts to analyze an element pair in the same mode (approximate or precise). 1 equals approximate and 2 equals precise. The default is 2.

MaxFailuresPerPart
The maximum number of failures before a part is excluded from the analysis. The default value is 2.
**MaxRelFailuresPerPart**

The maximum relative number of failures before a part is excluded from the analysis in respect to all occurrences of the element in the test bed. The specified value is a percentage. The default value is **100**.

**LoadFailures**

To include failed element pairs in the results file. Valid options are **0** or **1** for false and true, respectively.

**MaxLogFileSize**

The maximum size of individual log files in megabytes. The minimum value is **32 MB** and the maximum value is **2048 MB** (minus 1 byte subtracted). The default value is **2048**.

**IdentifyPartsWithDatastores**

If part occurrences are identified with file names or with part names in the log file. Valid options are **0** or **1** for false and true, respectively. The default value is **1**.

**IdentifyPartOccurrencesWithMetadata**

If part occurrences are identified with CADIDs or with specified metadata. No values or blank values indicate the choice of CADIDs. By default, part occurrences are identified with CADIDs.

**RecordPartPairAdditionEvents**

If test bed (element pair) creation events are recorded in the log file. Valid options are **0** or **1** for false and true, respectively. The default value is **1**.

**RecordClearanceResultAdditionEvents**

If clearance results creation events are recorded in the log file. Valid options are **0** or **1** for false and true, respectively. The default value is **1**.

**OffsetInFaceNormalDirection**

(For thick sheet clearance analysis) Which side of the surface the thickness is applied to. Valid options are **0** or **1** for false and true, respectively. The default value is the current registry value.

**ThicknessAttribute**

(For thick sheet clearance analysis) Find the thickness value for the surface parts. This can be used to override the default **CAD_PROP_MATERIAL_THICKNESS** property name or the currently defined value in the registry. The default value is **CAD_PROP_MATERIAL_THICKNESS** or the current registry value.

**DefaultMaterialThickness**

(For thick sheet clearance analysis) The default thickness value for surface parts that are missing the thickness attribute (**CAD_PROP_MATERIAL_THICKNESS**). If the **UseDefaultValue** option is set to **1** (true), the default value is the current registry value.

**UseDefaultValue**

(For thick sheet clearance analysis) To enable or disable the **DefaultMaterialThickness** value for surface parts with no material thickness property (**CAD_PROP_MATERIAL_THICKNESS**). Valid options are **0** or **1** for false and true, respectively.
SnapshotImageDirectory
(For use with 2D image captures) The directory where 2D image captures are placed.

SnapshotImageSize
(For use with 2D image captures) The size of 2D image captures. In pixels, specify the width and height of the generated images. The default size is 512,512.

SnapshotImagePNG
(For use with 2D image captures) The 2D image capture format. You can generate 2D captures in the PNG or JPEG formats. Use a value of 1 to create PNG images; use a value of 0 to create JPEG images. By default, JPEG files are created.

SnapshotImageBackground
(For use with 2D image captures) The color to use for the background shown behind the two elements involved in the clearance issue. This setting requires separate red, green, and blue values. By default, the background is set to 190,210,225.

SnapshotImagePartColor1
(For use with 2D image captures) The color to use for the first part involved in the clearance issue. This setting requires separate red, green, and blue values. If this setting is used, by default the part color is set to 0,0,255. If this setting is not turned on the original part color is used.

SnapshotImagePartTransp1
(For use with 2D image captures) The transparency setting for the first part involved in the clearance issue. You can use any value from 0.0 (invisible) to 1.0 (opaque). If this setting is used, the default value is 0.5.

SnapshotImagePartColor2
(For use with 2D image captures) The color to use for the second part involved in the clearance issue. This setting requires separate red, green, and blue values. If this setting is used, by default the part color is set to 0,255,0. If this setting is not turned on the original part color is used.

SnapshotImagePartTransp2
(For use with 2D image captures) The transparency setting for the second part involved in the clearance issue. You can use any value from 0.0 (invisible) to 1.0 (opaque). If this setting is used, the default value is 0.5.

SnapshotImageAdvTransparency
(For use with 2D image captures) To render part transparency with fewer artifacts. Use a value of 1 to turn on advanced transparency.

SnapshotImageShowCPPPoints
(For use with 2D image captures) To show the points of contact or penetration. Use a value of 1 to include the contact and penetration points in 2D image captures.

SnapshotImageCPPPointsColor
(For use with 2D image captures) The color to use for points of contact or penetration. This setting requires separate red, green, and blue values. If this setting is used, by default the contact and penetration point color is set to 255,255,0.
SnapshotImageCPPointsSize
(For use with 2D image captures) The size, in pixels, to use for points of contact or penetration. If this setting is used, the default value is 3.

SnapshotImageZoom
(For use with 2D image captures) The zoom factor for clearance issues. You can use any value from 0 to 100. A value of 0 shows the entire element pair. A value of 100 zooms to the contact or penetration points, or distance lines. If this setting is used, the default value is 40.

SnapshotImageShowDistanceLines
(For use with 2D image captures) Display distance lines in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, distance lines are not shown.

SnapshotImageShowResultVector
(For use with 2D image captures) Display the result vector in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, the result vector is not shown.

SnapshotImageShowBBox
(For use with 2D image captures) Display the bounding box in the image capture. Valid options are 0 or 1 for off and on, respectively. By default, the bounding box is not shown.

4. Save the file.

Run the Clearance Calculator

Take the following steps to perform ClearanceDB analysis using the Clearance Calculator:

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.

2. Using a text editor, open the Clearance.cfgproduct file and ensure that the ProductInputFile setting points to the location of your product data.

3. Do any of the following:

<table>
<thead>
<tr>
<th>To run the Clearance Calculator</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the command prompt</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td>analyze_product.pl [&lt;product&gt;]</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td>analyze_product.pl fishing_reel</td>
</tr>
</tbody>
</table>

(Windows) From the shortcut menu

In the products directory, right-click the Clearance.cfgproduct file and choose Analyze Product.

The analysis runs, and when it is complete the clearance results file, ClearanceResultsDbUpload.csvcldb, appears in the product directory.
Merging results with the database

For visualization clients to access analysis results, you must upload and merge the `ClearanceResultsDbUpload.csvclldb` file with your database.

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.

2. Do any of the following:

<table>
<thead>
<tr>
<th>To merge product attributes</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the command prompt</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td><code>update_product.pl [&lt;product&gt;] [ud] [df]</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td></td>
<td><code>update_product.pl fishing_reel -ud</code></td>
</tr>
<tr>
<td>(Windows) From the shortcut menu</td>
<td>In the ClearanceDB_Work_Area\Products\product_name directory, right-click the ClearanceResultsDbUpload.csvclldb file and choose <strong>Upload to ClearanceDB</strong>.</td>
</tr>
</tbody>
</table>

Understanding ClearanceDB log files

Running the Clearance Calculator generates a number of log files, most of which are automatically deleted unless an error occurs. This topic explains the contents of these log files.

The primary ClearanceDB log file is called `Log.txt`. The `Log.txt` file is an UTF-16 comma delimited file, which contains the event log of the analysis. This file is generated each time you run the Clearance Calculator.

The `Log.txt` file contains the following fields:

```
"Timestamp","Event","Name1","Datastore1","Name2","Datastore2"
"Timestamp","Event","CADID1","Datastore1","CADID2","Datastore2"
"Timestamp","Event","Name1","Datastore1","Name2","Datastore2"
"Timestamp","Event","CADID1","Part Number1","CADID2","Part Number2"
```

where

<table>
<thead>
<tr>
<th>This field</th>
<th>Records this information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>The date and time of the operation.</td>
</tr>
<tr>
<td>This field</td>
<td>Records this information</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Event</td>
<td>The name of the event. The following events are recognized:</td>
</tr>
<tr>
<td></td>
<td>• Pair added — A part pair was added to the test bed <em>(TmpPairs.txt)</em>.</td>
</tr>
<tr>
<td></td>
<td>• Loading pairs — The test bed is being loaded for analysis.</td>
</tr>
<tr>
<td></td>
<td>• Pair loading complete — All of the test bed is loaded for analysis.</td>
</tr>
<tr>
<td></td>
<td>• Result added — A clearance result (incident) was added to the temporary result set <em>(TmpResults.txt)</em>.</td>
</tr>
<tr>
<td></td>
<td>• Loading results — The temporary result set is being translated to the ClearanceResultsDbUpload.csvcldb format.</td>
</tr>
<tr>
<td></td>
<td>• Results loading complete — All of the temporary result set was translated to the ClearanceResultsDbUpload.csvcldb format.</td>
</tr>
<tr>
<td></td>
<td>• Failures loading — The set of failures is being added to the result set when requested.</td>
</tr>
<tr>
<td></td>
<td>• Failure loading complete — All of the failures were processed.</td>
</tr>
<tr>
<td></td>
<td>• Failure added — A failure to arrive at a clearance result due to failure to calculate within a time limit or at all is recognized.</td>
</tr>
<tr>
<td></td>
<td>• All processing is complete — Analysis is complete.</td>
</tr>
<tr>
<td></td>
<td>• Pair to be analyzed exactly the second time — The second attempt at NURBS analysis.</td>
</tr>
<tr>
<td></td>
<td>• Pair to be analyzed approximately the second time — The second attempt at facet based analysis.</td>
</tr>
</tbody>
</table>
### This field

**Records this information**

- Pair to be analyzed approximately only — The first attempt at facet based analysis after the NURBS analysis failed.

<table>
<thead>
<tr>
<th>Name1</th>
<th>Determined by the IdentifyPartOccurrencesWithMetadata setting in the Clearance.cfgglobal file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore1</td>
<td>Determined by the IdentifyPartsWithDatastores setting in the Clearance.cfgglobal file.</td>
</tr>
<tr>
<td>Name2</td>
<td>Determined by the IdentifyPartOccurrencesWithMetadata setting in the Clearance.cfgglobal file.</td>
</tr>
<tr>
<td>Datastore2</td>
<td>Determined by the IdentifyPartsWithDatastores setting in the Clearance.cfgglobal file.</td>
</tr>
</tbody>
</table>

**Note**

The Timestamp and Event data is always present in the file. The remaining fields are populated only if the appropriate settings in the Clearance.cfgglobal file are defined.

The **Clearance.cfgglobal** contains a number of other parameters which affect the contents of the Log.txt file. You can adjust the following settings:

<table>
<thead>
<tr>
<th>This setting</th>
<th>Does this</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProgressDirectory</td>
<td>Specifies the location of all log files, including the Log.txt file. By default, these files are placed in a directory called FTC, which is located in the same directory as the Clearance Calculator executable.</td>
</tr>
<tr>
<td>IdentifyPartsWithDatastores</td>
<td>Indicates whether part occurrences are identified with file names or with part names in the Log.txt file (1 or 0). The default is 1.</td>
</tr>
<tr>
<td>MaxLogFileSize</td>
<td>Defines the size of individual log files (Log.txt, Log_2.txt, etc.) in megabytes. If the log file size limit is reached, the application automatically creates multiple log files. Multiple log files follow the naming convention, Log.txt, Log_2.txt, Log_3.txt, etc. The minimum log file size is 32 MB. The maximum log file size is 2048 MB (minus 1 byte subtracted). The default size is 2048 MB.</td>
</tr>
<tr>
<td>RecordPartPairAdditionEvents</td>
<td>Controls whether the test bed (part pair) creation events are recorded in the log file (1 or 0). The default is 1.</td>
</tr>
<tr>
<td>RecordClearanceResultAdditionEvents</td>
<td>Controls whether the clearance results creation events are recorded in the log file (1 or 0). The default is 1.</td>
</tr>
</tbody>
</table>
In addition to the Log.txt file, the following temporary files are created and deleted when you run the Clearance Calculator:

<table>
<thead>
<tr>
<th>This file</th>
<th>Has this purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TmpLock.txt</td>
<td>Indicates that the Clearance Calculator is utilizing a particular directory (where the file is created, specified by the ProgressDirectory parameter in the Clearance.cfgglobal file). Only one ClearanceExe application can use a given working directory at a time. This is a binary file, and it is always 0 bytes in size.</td>
</tr>
<tr>
<td>TmpFail.txt</td>
<td>Counts failures to arrive at the result of clearance analysis. This is a binary file.</td>
</tr>
<tr>
<td>TmpPairs.txt</td>
<td>Stores the list of part pairs to be analyzed (the clearance test bed). This is a binary file. It is 12 * (number of part pairs) bytes in size.</td>
</tr>
<tr>
<td>TmpResults.txt</td>
<td>Stores the temporary results of clearance calculations (violations only). The records of the file are keyed off the internal IDs of part occurrences, and are not meaningful except to the Clearance Calculator application. This is an ASCII comma delimited file.</td>
</tr>
<tr>
<td>TmpHeader.txt</td>
<td>Monitors the progress of the Clearance Calculator worker process. This is an ASCII comma delimited file. It consists of a single record of the following seven fields:</td>
</tr>
<tr>
<td></td>
<td>• Field 1 — The name of the file relative to the directory the Clearance Calculator is started from.</td>
</tr>
<tr>
<td></td>
<td>• Field 2 — The timestamp of the start of the Clearance Calculator.</td>
</tr>
<tr>
<td></td>
<td>• Field 3 — The timestamp of the start of the Clearance Calculator, or the last re-start, whichever is later.</td>
</tr>
<tr>
<td></td>
<td>• Field 4 — The cardinality of the clearance test bed as it is discovered, starts at 0, then grows to some value, and remains unchanged during the actual part pair clearance calculations. This correlates to the TmpPairs.txt size with the factor of 12.</td>
</tr>
<tr>
<td></td>
<td>• Field 5 — The currently analyzed part pair, starts with 0, then grows to the value of Field 4, and is incremented by 1 when all of the calculations complete.</td>
</tr>
</tbody>
</table>
The file has the following purpose:

| Field 6  | The number of failed attempts to arrive at the result of the current part pair. |
| Field 7  | The initialization state: 1 after the test bed is completely generated. |

Example

"FTC/TmpHeader.txt", "2006/05/26 15:09:34", "2006/05/31 09:24:28", "7126488", "268848", "0", "1"

Understanding ClearanceDB advanced log files

**EAI_CL_ADVANCED_TRACE_FILE** is a log file for clearance batch analysis that reports all major events.

**Warning**

Enabling this log file may significantly affect system performance, especially if running in a multi-process or multi-thread mode.

The log file is enabled by setting the environment variable **EAI_CL_ADVANCED_TRACE_FILE**.

**Example**

EAI_CL_ADVANCED_TRACE_FILE=mylog.log

Unless a full path is specified for the log file, the file is created in the directory from which **analyze_product.pl** or **analyze_managed_product.pl** is executed.

Because a large amount of data can be written to the log file, the environment variable, **EAI_CL_ADVANCED_TRACE_FLAGS**, is used to control the data that is saved, allowing you to include additional sections. You can add these sections:

- **LOADING** – 3D data loading events.
- **EXCLUSIONS** – Exclusions from the clearance analysis due to ClearanceDB rulebase and variant conditions.
- **ANALYSIS** – Committed memory and available physical memory reported as calculations progress (for multi-threaded analysis only).
- **URESULTS** – Undefined results, including pair analysis timeouts.

To include these sections, specify them with the environment variable as follows:
**Example**

To include one section: `EAI_CL_ADVANCED_TRACE_FLAGS=LOADING`

To include multiple sections, separate them with a `+`

`EAI_CL_ADVANCED_TRACE_FLAGS=LOADING+EXCLUSIONS+ANALYSIS+RESULTS`

The **EAI_CL_ADVANCED_TRACE_FILE** log file contains the following columns:

- Timestamp
- Process Number
- Thread Number
- Authoring module
- Event
- Event Outcome
- Number (relevant to the owning event row)
- Text Field 1 (relevant to the owning event row)
- Text Field 2 (relevant to the owning event row)
- (Optional) Event Duration (In seconds, usually associated with the *End* events in respect to *Begin* events.)

**Understanding the results file format**

ClearanceDB results are saved as the **ClearanceResultsDbUpload.csvcldb** file. Each record in this file consists of four fields:

- **1** — Record type
- **2** — Entity ID
- **3** — Entity attribute name or referenced entity ID
- **4** — Entity attribute value or referenced entity ID

The following four record types are supported:

**Record type 0 — general timing information**

- **1** — Record type = 0
- **2** — Timestamp
- **3** — Event name
- **4** — Formatted timestamp
Performing analysis

Clearance checks are performed using the viewer (stand-alone Mockup, the Lifecycle Viewer, Structure Manager, or DesignContext) or the Clearance Calculator. Analysis can refer to either the standard single requirement defined in the viewer or on rules and conditions stored in the ClearanceDB database. The latter method allows for incremental calculations in respect to the database content.
Performing clearance analysis on a product generates a clearance results file called `ClearanceResultsDbUpload.csvcldb`. This file contains the product definition (names of assemblies and parts, their CADIDs, locations, and metadata) and the clearance issues with their geometric attributes. To view clearance results in the viewer, this file must be merged with the database.

To perform analysis on a product and update the database with the results, you must take the following steps:

- Enable the ClearanceDB rules and results components
- Configure the batch analysis options for the Clearance Calculator
- Use the Clearance Calculator to perform analysis on your product
- Merge the results generated by the Clearance Calculator with the ClearanceDB database

You should understand the log files generated when you run the Clearance Calculator in case errors occur. It also may be useful to understand the ClearanceDB results file format.
Chapter 9: Updating products in the database

Updating products in the database

Product data, including configurations, rules, conditions, and zones, must be merged with the ClearanceDB database to be available for the analysis. Results of the analysis also must be uploaded to the database to be available for evaluation in the viewer.

There are two methods available for merging product data with the database, uploading via the ClearanceDB Proxy and the Oracle Instant Client, or directly uploading via a full Oracle Client installation.
Configuring ClearanceDB for product updates

You must update the Clearance.cfgglobal file to update the ClearanceDB database with your product data.

1. Navigate to the ClearanceDB Work Area.

2. Using a text editor, open the Clearance.cfgglobal file.

3. Change any of the following settings in the Section 3: ClearanceDB update/upload settings section of the file to configure product updates:

   **ClearanceDBResultsUploadFile**
   The name and, optionally, the path of the results file generated by the Clearance Calculator. By default, the name of this file is ClearanceResultsDbUpload.csvcldb.

   **UploadMethod**
   The option specifying how to merge data with the ClearanceDB database. You can specify one of the following:

   - **Proxy** — Perform updates using the ClearanceDB Proxy.
   - **NoProxy** — Perform updates using the Oracle Client only.

   **Note**
   To use this option, the full Oracle Client must be installed on the machine from which you want to make database updates.

   **OracleSqlLoaderExecutable**
   The name of the Oracle SQL*Loader executable.

   **Note**
   This is required only if UploadMethod is set to NoProxy.

   **OracleSqlPlusExecutable**
   The name of the Oracle SQL*Plus executable.

   **Note**
   This is required only if UploadMethod is set to NoProxy.

   **ClearanceDBOracleUserCredentials**
   The Oracle user account information. Type this according to the following syntax:

   `username/password`
Updating products in the database

**Note**
This is required only if `UploadMethod` is set to `NoProxy`.

**ClearanceDBOracleNetServiceName**
The Oracle Net Service Name.

**Note**
This is required only if `UploadMethod` is set to `NoProxy`.

4. Save the file.

**Merge product data with the database**

You must merge your product data, including configurations, rules, conditions, zones, and results, with the database.

1. Navigate to the appropriate product directory in your ClearanceDB Work Area.

2. Do any of the following:

<table>
<thead>
<tr>
<th>To merge product attributes</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the command prompt</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td><code>update_product.pl</code> [&lt;product&gt;]</td>
</tr>
<tr>
<td></td>
<td><code>-uc:-ur:-ud:-uz:-ua [-df]</code></td>
</tr>
<tr>
<td></td>
<td>where</td>
</tr>
<tr>
<td></td>
<td><code>product</code> — is the product name.</td>
</tr>
</tbody>
</table>

**Note**
If you run the `update_product.pl` script from within a product directory, you do not need to specify the product name that you want to update. If you run `update_product.pl` from the root of the ClearanceDB Work Area, you must specify the product name.

`-uc` — The argument to upload the configuration.
### To merge product attributes

<table>
<thead>
<tr>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>−ur — The argument to upload the rules and conditions.</td>
</tr>
<tr>
<td>−ud — The argument to upload the results.</td>
</tr>
<tr>
<td>−uz — The argument to upload the zones.</td>
</tr>
<tr>
<td>−ua — The argument to upload all of the product data, including the configuration, rules and conditions, results, and zones.</td>
</tr>
<tr>
<td>−df — The argument to upload the default configuration of the product.</td>
</tr>
</tbody>
</table>

#### Example

```
update_product.pl -ua
```

---

<table>
<thead>
<tr>
<th>(Windows) From the shortcut menu</th>
<th>In the directory, right-click any of the following files and choose <strong>Upload to ClearanceDB</strong>:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <code>configuration.csvcldb</code> — ClearanceDB process configuration file</td>
</tr>
<tr>
<td></td>
<td>• <code>rules.csvcldb</code> — ClearanceDB rules definition file</td>
</tr>
<tr>
<td></td>
<td>• <code>conditions.csvcldb</code> — ClearanceDB conditions definition file</td>
</tr>
<tr>
<td></td>
<td>• <code>zones.csvcldb</code> — ClearanceDB zones definition file</td>
</tr>
<tr>
<td></td>
<td>• <code>ClearanceResultsDbUpload.csvcldb</code> — ClearanceDB results file</td>
</tr>
</tbody>
</table>

Your data is merged with the database and log files are created and saved to the product directory.
Chapter 10: Viewing and managing results

Clearance results tasks in the viewer

ClearanceDB results evaluation and issue management occurs within the viewer. Supported visualization clients include stand-alone Mockup, the Lifecycle Viewer, and Structure Manager.

Viewing clearance issues and changing their status within the viewer consists of the following activities:

- Use the clearance preferences within the viewer to enable access to the ClearanceDB database
- Load and view results from the database
- Use optional server-side filters to obtain an additional level of control over which results are loaded into the viewer
• Understand the different types of potential clearance violations
• Display the clearance issue status
• Analyze clearance issues with multiple occurrences
• Update the clearance issue status in the database

Enable ClearanceDB within the viewer

You must enable ClearanceDB functionality within the viewer to evaluate database results.

1. In the viewer, choose **Analysis tab**→**Clearance group**→**Enable Clearance Analysis**.

2. In the **Clearance** group, choose **Clearance Preferences**→**Requirement Components**.

3. In the **Requirement Rule Components** dialog box, select **Requirement Rules from ClearanceDB Server**.

4. In the **Clearance** group, choose **Clearance Preferences**→**Result Components**.

5. In the **Result Components** dialog box, select one of these options:

<table>
<thead>
<tr>
<th>Use</th>
<th>To enable this</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Clearance Results</td>
<td>Visual representation of results within the application. Select this option if you only want to view and manage ClearanceDB results from the database.</td>
</tr>
<tr>
<td>ClearanceDB Results</td>
<td>Results upload to the Oracle database. Select this option if you want to perform analysis locally within Mockup, and then upload the results to the database.</td>
</tr>
</tbody>
</table>

**Note**

To upload results to the database from the viewer, the following environment variables must be set:

- **EAI_CL_CONNECT_DATA** — The proxy connection name
- **EAI_CL_DATASOURCE** — The ClearanceDB Proxy system name and port number
- **EAI_CL_UPLOAD_FILE** — A name for the ClearanceDB results file

6. Click **OK**.
Load ClearanceDB results in the viewer

1. In the viewer, open the 3D model that has product data and associated ClearanceDB results in the database.

2. Choose Analysis tab→Clearance group→Load Results→Load Results 🟢. The Load Clearance Results dialog box appears.

3. In the Load Clearance Results dialog box, select ClearanceDB DataBase Connection (*.DBC) from the Files of type list.

4. Select your Database Connection (DBC) file, and click Open. If there are ClearanceDB results associated with the model in the database, they appear in the Results list.

Using server-side filters

Server-side filtering extends the use of the viewer clearance filters to the ClearanceDB database server. When applied to ClearanceDB results already loaded in the viewer, a filter specifies which results to display. When applied to ClearanceDB database queries, a filter controls which results to load into the client. Server-side filters provide smaller and faster clearance result loading times, especially when working with large sets of data.

Note

Server-side filters do not affect the display of clearance results already displayed in the Clearance Results window. They affect the set of clearance results that are loaded from the database to the viewer. If you want to refine clearance results already loaded from the database, you can create and apply an additional filter from within the viewer.

While the performance of server-side filters varies according to the filter definition, these are some general guidelines:

- The more selective the filter, the smaller the clearance results set and the faster the response from the database.

- Filters with the "=" operator result in faster queries than those with the "!=" operator.

- Definitions without wild cards ("*"*) are more efficient than those containing them.

Filter modes

There are two modes of server-side filtering available, Direct and Symmetric. The default filter mode is Direct.

In Direct mode, the filter is interpreted by the database as is. In particular, the references to Part Number1 and Part Number2 are applied only to parts identified in the database as Part Number1 and to Part Number2. In Symmetric mode, the references to Part Number1 and Part Number2 are augmented symmetrically with corresponding references to Part Number2 and Part Number1.
For example, the filter Part Number1 = Name1 is augmented with the Part Number2 = Name1 clause in the Symmetric mode. In addition, Part Number1 = Name1 and Part Number2 = Name2 is augmented by Part Number1 = Name2 and Part Number2 = Name1, and so on.

Referencing part attributes

Server-side filtering supports part attributes in both the Direct and Symmetric modes. Part attributes to be filtered on can be either direct or inferred.

Direct attributes are directly attached to parts. In the viewer, you can view these by right-clicking a part, and selecting Properties. Inferred attributes are not necessarily attached to parts. They can be associated with any assembly the part belongs to. Inferred attributes can be discovered using any of the following search methods:

- Part only
- Part parent only
- Part and parents (part first)
- Part and parents (root first)
- Parents only (part parent first)
- Parents only (root first)

For server-side filters to reference inferred attributes, you must do the following:

- Include attribute configuration specifications in the Clearance.cfgglobal or Clearance.cfgproduct file.
- Perform the clearance batch analysis, and merge the results with the ClearanceDB database.
- Define the server-side filter in the viewer, choosing the appropriate columns from the Filter Values dialog box.

Create server-side filters

To create server-side filters, do the following:

1. In the viewer, define a clearance filter with your search criteria. Save the filter to the registry.

   Note
   Refer to Filtering clearance results in the viewer help for detailed information on creating clearance filters and saving them to the registry.

2. Create or modify a DBC file in the following way:
<table>
<thead>
<tr>
<th>To</th>
<th>Add this line</th>
</tr>
</thead>
</table>
| Enable a filter | `FILTER_NAME=[FILTERNAME]`  
  where  
  `FILTERNAME` is the name of the filter saved to the registry. |
| Specify to use the filter designated by the user to be loaded on startup | `FILTER_NAME=eaiclINITIAL` |
| Specify that no filter should be applied | `FILTER_NAME=eaiclNONE` |
| Specify the filter mode | `FILTER_MODE=[FILTERMODE]`  
  where  
  `FILTERMODE` is one of these values:  
  • DIRECT  
  • SYMMETRIC |

**Example**

For example, a DBC file that enables a server-side filter might look like this:

```
DATASOURCE=proxymachine:7206
CONNECT_DATA=SECONDDB
FILTER_NAME=myCDBFilter
FILTER_MODE=DIRECT
```

3. Within the viewer, load the DBC file.

The registry is searched for the specified filter. If found, the filter definition is submitted to the clearance database. If results match the filter criteria, they are loaded into the viewer. If the filter is not found in the registry, you can load the results without filtering or cancel loading the results.

**Overview of clearance types**

When you perform a clearance check, any two parts or elements that are closer to one another than a specified distance are identified as an issue for further investigation. The **Results** list displays the clearance issues identified by the analysis, along with the distance value between element pairs and other relevant information such as the interference type.

Clearance interferences can be *soft* or *hard*. Soft interferences are distance violations. They occur when the distance between two elements is equal to or closer than the specified minimum distance. Elements involved in a soft interference do not touch or penetrate one another. Hard interferences are contact or penetration violations. They occur when two elements touch or occupy the same geometric space.

The table below describes each clearance type and provides examples to help you understand the criteria used to identify them.
<table>
<thead>
<tr>
<th>This type</th>
<th>Appears in the results list as</th>
<th>Indicating this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>C</td>
<td>A hard interference. The elements are in contact but do not penetrate. The distance result is always ~ 0.0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clearance results are approximate if your elements do not consist of precise geometry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clearance results of geometrically similar parts that overlap may not recognize the contact if the individual polygons in the tessellated versions of the parts exactly correspond to each other.</td>
</tr>
<tr>
<td>Distance</td>
<td>D</td>
<td>A soft interference. The elements are closer than the specified, but do not come into contact with or penetrate one another. The distance result is always a value greater than ~ 0.0.</td>
</tr>
</tbody>
</table>

**Note**

You also can use clearance calculator settings to adjust the Contact Tolerance value. Specify a tolerance for penetrations so that any penetration value that is less than the tolerance is defined as a contact.
<table>
<thead>
<tr>
<th>This type</th>
<th>Appears in the results list as</th>
<th>Indicating this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure</td>
<td>F</td>
<td>The clearance calculation has failed to complete. This may happen if the clearance calculator crashes.</td>
</tr>
<tr>
<td>Penetration</td>
<td>P</td>
<td>A hard interference. One element penetrates or is completely contained by another. The distance result is always a value less than ( \sim 0.0 ).</td>
</tr>
</tbody>
</table>

**Note**

Penetration values are always approximate, even when using precise geometry.

**Partial penetration**

For a clearance issue that consists of a pair of elements where one element is completely contained within another, the distance value is the approximate penetration depth of the contained element. If the minimum distance between elements is \( \sim 0.0 \) (the elements are in contact with one another), and the elements contain NURBS data, the penetration depth is the length of the bounding box diagonal of the inner element.
<table>
<thead>
<tr>
<th>This type</th>
<th>Appears in the results list as</th>
<th>Indicating this</th>
</tr>
</thead>
</table>

**Note**

If the minimum distance between elements is ~ 0.0 and the parts do not contain NURBS data, the issue is classified as a contact.

![Diagram of contact](image1)

**Full penetration (elements in contact)**

If the minimum distance between elements is greater than ~ 0.0 (the elements are not in contact with one another), the penetration depth is the minimum distance plus the distance the minimum distance vector extends to the further side of the inner elements' bounding box.

![Diagram of full penetration](image2)

**Full penetration (elements are apart)**

**Note**

- Completely contained elements are categorized as penetrations only if the outer element is geometrically closed.

- The inner element must consist of tessellated geometry. An element pair with an inner element that consists only of wireframe geometry is not identified as a penetration.
<table>
<thead>
<tr>
<th>This type</th>
<th>Appears in the results list as</th>
<th>Indicating this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>T</td>
<td>A violation where is a factor of the calculation. is required to detect whether an issue involving material thickness is a soft or hard interference. The distance result can be a value less than or greater than ~ 0.0.</td>
</tr>
<tr>
<td>Unknown</td>
<td>U</td>
<td>A hard interference. is required to classify as a contact or penetration.</td>
</tr>
</tbody>
</table>

**Note**

Element pairs with material thickness values that come into contact or penetrate one another are identified as the Unknown clearance type. Contact, penetration, and distance line analysis are not available for clearance issues that contain element pairs with material thickness values.

Display clearance issue status

The database extensions to the viewer allow the end user to load the clearance results from the database via a Database Connection (DBC) file. Once the results are displayed in the viewer, the end user can evaluate clearance issues and then update their status in the database.

The end user can trace the status of the database content and its relationship to the client by examining the following clearance view fields:
• **Db Sync** — Indicates whether there are uncommitted changes to clearance attributes (Yes/No), or whether the clearance issue stored in the database can be matched with currently opened JT document (Yes/<blank>).

• **Db Modified On** — Contains the date of the last change to user dispositions.

• **Db Modified By** — Contains the OS login name of the user who committed changes to user dispositions last.

• **Db Message** — Gives verbose indication of the status of the clearance issue.

• **Db Change** — Indicates changes in the product, issues, and its dispositions.

• **Db Rules** — Displays the numbers of the violated clearance requirement rules.

• **Db Rule Descriptions** — Displays the descriptions of the violated clearance requirement rules.

• **Db Match** — Displays the occurrence type for each issue:
  - **I (Identical)**
    All identical issues in a group are linked when the inheritance process is executed in the database. The master issue (the one that the others are linked to) is the one that was most recently changed.
  - **S (Similar)**
    All similar issues in a group share a common baseline issue, which is the one that was most recently changed. When the baseline issue is dispositioned, the information in the baseline issue is copied to the other issues in the same group. This copy operation happens only once, and future changes affect only that issue.
  - **R (Resembling)**
    Same as similar, except the issues are marked as not evaluated in the database. This typically means that someone must investigate and evaluate the issue.
  - **X**
    The issue is unrelated.

• **Db Base** — Displays the base issue for similar and resembling occurrences. The column is blank for identical occurrences.

• **Db Issue** — Displays the database issue number for each issue. For identical occurrences, this number will be the same.

• **Db Issue Count** — Displays the number of issues in each identical group. For similar, resembling, and unrelated occurrences, this is always 1.

**Note**
All of the above fields are read only.
Analyze multiple instances of part pair violations

More than one instance of the same violating part pairs may exist in your clearance results. Inheritance relationships can be defined in your database so that these issues are not always duplicated but instead are managed based on their level of similarity. Membership in a specific inheritance category determines to what degree they share clearance issue attributes and dispositions. For example, if an issue meets the criteria for being Identical, the issue is shared between occurrences and appears only once in the clearance results. If the issue is later updated or changed, all part occurrences reference the same, updated version of the issue. Also, if these part pair occurrences exist in different products, they can share the same issue. The attributes of Similar and Resembling issues are shared to a lesser degree and are handled differently from Identical issues, as explained below.

1. In the Results list, review the issue occurrences.

   The Db Match, Db Base, Db Issue, Db Issue Count columns provide information about the occurrences.

   Three different levels of similarity may be defined for your database, and they appear in the results lists with different symbols, as follows:

   **Note**
   
   Unrelated issues appear with the symbol.

   • Identical

   Identical issues are linked when the inheritance process is executed in the database. The master issue (the one that the others are linked to) is the one that was most recently changed.

   • Similar

   Similar issues share a common baseline issue, which is the one that was most recently changed. When the baseline issue is dispositioned, the information in the baseline issue is copied to the other issues in the same group. This copy operation happens only once, and future changes affect only that issue.

   • Resembling

   Same as similar, except the issues are marked as not evaluated in the database. This typically means that you must investigate and evaluate the issue.

2. To group issues by element pair, right-click a column heading, and choose Group Tree by Elements.

3. On the Clearance Results toolbar, click Show/Hide Occurrences Window.

   A window appears below the Results window, providing details about the selected occurrence.

4. Select other issues in the Results list to view the details for the occurrences.
Update issue status database from the viewer

1. In the viewer, right-click an issue in the Results list, and change any of the following settings:
   - Set Owner
   - Set Status
   - Set Priority
   - Set Zone
   - Enter Resolution

   If your ClearanceDB configuration is set to Autocommit mode, database updates are triggered by changes to any of the above values.

2. (If not in Autocommit mode) Right-click an issue in the Results list, and select Commit.
Chapter 11: Using ClearanceDB with Teamcenter

Using ClearanceDB with Teamcenter

You can perform ClearanceDB analysis on product data stored in Teamcenter. Analysis of managed product data is performed via the `analyze_managed_product.pl` script, which launches the processes that evaluate the data and then automatically uploads the results to the ClearanceDB database.

ClearanceDB offers a number of options for working with Teamcenter, including:

- Analyze the data associated with BOM lines on a product-by-product basis.

- Perform a single clearance variant analysis that excludes all non-buildable part pairs from a 150% BOM line.

- Work with DesignContext to identify a series of target parts and then quickly find other relevant data within a given proximity to those parts.

- Analyze the data associated with 4GD worksets.

You can evaluate ClearanceDB results in stand-alone Mockup, the Teamcenter Lifecycle Viewer, Structure Manager, and DesignContext. However, ClearanceDB results based on 4GD worksets can be evaluated only in stand-alone Mockup.
Product and system requirements

The ClearanceDB integration with Teamcenter has the following requirements:

• A properly configured and functioning ClearanceDB software environment, including the ClearanceDB Server, Proxy, and Client tiers.

• A Teamcenter 8.x, 9.x, or 10.x server.

• An FMS file client cache (FCC).

• A Teamcenter user account with system administrator privileges.

• A Mockup license in the rich client to view results in the Lifecycle Viewer or Structure Manager, or DesignContext with the Repeatable Digital Validation (RDV) component.

• Product data in the JT or PLM XML format within an item revision on the Teamcenter server.

• Absolute occurrence IDs for all BOM lines to be analyzed.

Steps to analyze managed products

The following process describes the tasks needed to analyze a Teamcenter item revision in ClearanceDB:

1. Ensure that your environment meets the requirements for performing ClearanceDB analysis on data from Teamcenter.

2. Create a ClearanceDB product, following the procedures located in the Setting up products for analysis section of this guide. This includes the following:

   • The product configuration, which is required.

   Note

   In a managed environment, ClearanceDB is configured by Teamcenter revision rules, so ClearanceDB configuration names are Teamcenter revision rule names.

   • Rules, at least one of which is required.

   • Conditions, which are optional.

   • Zones, which are optional.
Note

The name of the product must be the same as the top-level node in the assembly. Send the item revision containing the product data to Structure Manager to obtain the name of the top level assembly node, which is displayed in the BOM Line column.

If you are working within a Teamcenter multifield key environment, see Configuring ClearanceDB for multifield key data.

3. Update the global configuration options for working with Teamcenter data.

4. Specify for ClearanceDB to use absolute occurrence IDs.

5. Specify the item and revision IDs for the top level assembly of your product.


7. If you want to perform analysis using Teamcenter variant conditions, enable variant analysis for clearance.

8. If you want to cache the product data on your local machine to improve the performance of the analysis, adjust the load_fcccache options.

9. Use the analyze_managed_product.pl script to perform analysis on the managed product and merge the results with the database.

Configuring ClearanceDB to work with Teamcenter

Teamcenter global configuration options

You must update the Clearance.cfgglobal file to match your Teamcenter environment.

1. Using a text editor, open the Clearance.cfgglobal file.

2. Change any of the following settings in the Section 1: Teamcenter Configuration section of the file to configure ClearanceDB to work with Teamcenter:

   **TC_ROOT**
   Specifies the Teamcenter root directory.

   **TC_DATA**
   Specifies the Teamcenter TC_DATA directory.

   **ItemRevisionType**
   Specifies to perform analysis of 4GD worksets. If ItemRevisionType=Workset, then item, rev, and rev_rule arguments are interpreted as those of a workset, and the plmxml_export utility will be invoked to extract the product structure from Teamcenter, rather than the bomwriter utility.
TeamcenterUserId
Specifies a Teamcenter user name with Teamcenter system administration privileges. If no user name is provided, ClearanceDB uses the operating system user name.

**Note**
This is the same option as the `bomwriter -u=` command line argument.

TeamcenterUserPassword
This setting is deprecated. The Teamcenter user password is now stored in a separate file, and the path to the file is specified with the `TeamcenterUserPasswordFile` setting. Now if the password is specified with `TeamcenterUserPassword`, it is written to an unencrypted file in the user’s home directory with a randomly generated name beginning with `CLDB`.

TeamcenterUserPasswordFile
Specifies the path to a file containing the Teamcenter user password.

**Note**
This is the same option as the `bomwriter -pf=` command line argument.

TeamcenterUserGroupId
Specifies the Teamcenter group ID. If no group ID is provided, ClearanceDB uses the default group of the specified Teamcenter user.

**Note**
This is the same option as the `bomwriter -g=` command line argument.

RevisionRule
Specifies the revision rule for the product's top level product structure node, or the revision rule of a workset when `ItemRevisionType=Workset`.

**Note**
This is the same option as the `bomwriter -revision_rule=` command line argument, or the `plmxml_export -rev_rule` when `ItemRevisionType=Workset`.

SavedVariantRule
Specifies the saved variant configuration to pass to the `bomwriter`. The setting is not active when `ItemRevisionType=Workset`.

LexicographicalVariantAnalysis
Defines the mode of the evaluation of variant condition relational expressions. Choose one of the following:
**No** — If undeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

**Yes** — If undeterminable variant conditions exist, evaluate them lexicographically.

**Note**
Variant conditions that cannot be evaluated based on textual representations are undeterminable.

**VariantAnalysisClientTraceFile**
Specifies the name for the log file of the variant analysis. By default, this is not set, and a log file is not generated.

**Example**

```
VariantAnalysisClientTraceFile=cldb_variant_analysis.log
```

**VariantAnalysisClientTraceFlags**
Specifies the content of the log file of the variant analysis. The trace flags are additive, concatenated with the plus (+) character.

**Example**

```
VariantAnalysisClientTraceFlags=CONFIGURATION+STATISTICS
```

Add any of the following flags:

- **CONFIGURATION**
  Lists the variant analysis configuration attributes.

- **STATISTICS**
  Provides the basic variant analysis statistics.

- **VARIANTCONDITIONS**
  Provides details of textual variant conditions.

- **DETERMINABILITY**
  Lists the determinability of variant conditions.

- **SATISFIABILITY**
  Lists the satisfiability of variant conditions.

- **EXCLUDEDVCPAIRS**
  Lists excluded variant condition pairs.

- **EXCLUDEDUIDPAIRS**
  Lists excluded clearance element pairs as ABSOCCs.

- **EXCLUDEDNGIDPAIRS**
  Lists excluded clearance element pairs as NGIDs.

- **IMPACT**
  Lists the impact of variant conditions on the product structure.

**Caution**
Enabling the **EXCLUDEDUIDPAIRS** and **EXCLUDEDNGIDPAIRS** trace flags may result in extremely large log files.
**VariantLogicalExpression**
Defines the variant condition UserValue title and BOM line property pair in the PLM XML file. This has the form of:

“Variant Condition”:bl_variant_condition

“Variant Formula”:bl_formula

**Example**

VariantLogicalExpression="VC:bl_variant_condition"

**Note**
This must match the content of the BomWriterUserAttributes setting.

**RunLevel**
Specifies the actions of the analyze_managed_product.pl script. Use a value from 1 to 6.

1 — In the product directory, a.vvi file is created, which is used by the bomwriter to generate a .plmxml file referencing the managed data.
2 — Using the .vvi and .plmxml files in the product directory, the Clearance Calculator performs analysis upon the managed product data and generates a results file.
3 — The results file is uploaded to the ClearanceDB database.
4 — Both the RunLevel 1 and 2 actions are performed.
5 — Both the RunLevel 2 and 3 actions are performed.
6 — All of the RunLevel actions are performed, 1, 2, and 3.

**BomWriterUserAttributes**
Specifies the variant conditions for the bomwriter to include in the generated .plmxml file. Type these according to the following syntax:

target:Instance,key:myAttribute,literal:"My Attribute Value"

**Note**
This is the same option as the bomwriter -ua= command line argument.

**Example**

Use this option to include variant model related BOM line properties in the generated .plmxml file. For example:

BomWriterUserAttributes=target:Instance,key: VC,prop: bl_variant_condition

**TeamcenterWebServerPath**
Specifies the Teamcenter web server path, including the protocol, host name, and port number. Type this according to the following syntax:
TeamcenterWebServerPath=http://machine_name:port

**Example**


**JtDataStagingProcess**

Specifies to use the Teamcenter `load_fcccache` utility to download the model data from the Teamcenter server to the local system for clearance analysis. Enabling this option pre-populates the FMS client cache (FCC), which leads to faster and more reliable analysis.

Valid values are 0 (off) or 1 (on). The default value is 0.

**JtDataStagingProcessErrorLimit**

Specifies when to abort the `analyze_managed_product.pl` script, based on the following custom error codes:

- FILECOPY_FAILED
- INVALID_DAKID_FORMAT
- FCC_OPENFILE_FAILED
- FCC_DOWNLOAD_FAILED
- GET_READ_TICKET_FAILED
- DATASET_READ_FAILED
- PLMXML_MISSING_JT
- CHMOD_FAILED
- COPYOUT_CLEANUP_FAILED

You can specify a numerical value for each error code. By default, the error codes are given values that correspond to the severity of the problem, with the lowest value representing the most severe failure. If a value is not specified for the `JtDataStagingProcessErrorLimit` setting, the script stops whenever an error occurs during the staging process.

The default value is 30.

**Note**

The default values are recommended for the `JtDataStagingProcessErrorLimit` setting and the related error codes. Essentially, with these defaults the clearance analysis will abort when any of these errors occur, and you can use the reported error message to troubleshoot the problem.

**FILECOPY_FAILED**

Specifies the error code that indicates a copy operation to the output directory has failed.

The default value is 20.

**INVALID_DAKID_FORMAT**

Specifies the error code that indicates an invalid DAKID was found.

The default value is 21.
**FCC_OPENFILE_FAILED**
Specifies the error code that indicates the FCC failed to open the file using the ticket.
The default value is 22.

**FCC_DOWNLOAD_FAILED**
Specifies the error code that indicates a failure most likely due to a missing file in the volume.
The default value is 23.

**GET_READ_TICKET_FAILED**
Specifies the error code that indicates a read ticket failed.
The default value is 24.

**DATASET_READ_FAILED**
Specifies the error code that indicates no read access on the dataset.
The default value is 25.

**PLMXML_MISSING_JT**
Specifies the error code that indicates a missing JT file reference in the .plmxml file.
The default value is 26.

**CHMOD_FAILED**
Specifies the error code that indicates a failure to set the access mode during copy out.
The default value is 27.

**COPYOUT_CLEANUP_FAILED**
Specifies the error code that indicates a failure to remove a file during lifetime cleanup.
The default value is 28.

**CopyOutLocation**
Specifies the location for the dataset files downloaded from Teamcenter server.
The default location is the product directory in the ClearanceDB Work Area.

**Note**
Do not use special characters in folder names.

**BucketCount**
Specifies how many directories to use for the cached files. Spreading the files over multiple directories can lead to better performance.
The default value is 30.

**UseAbsoluteLocation**
Specifies to use an absolute value for the location attribute in the .plmxml file generated by the bomwriter. It is usually better to have a relative reference, although there are instances where an absolute reference is required, such as when the .plmxml file is moved to a different location from the referenced files.
The default value is **No**.

**DirAccessMode**
Specifies the access mode setting for the directories created to hold the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**FileAccessMode**
Specifies the access mode setting for the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**BucketPrefix**
Specifies a prefix to add to the names of directories created to hold the cached files.

The default value is **RW**.

**FilenamePrefix**
Specifies a prefix to add to the names of the cached files.

The default value is **fmsr_**.

**CopyOutLifetime**
Specifies the lifetime of the files cached in the **StagingProcessDownloads** directory. The directory is scanned for files older than the specified value, which are removed. The lifetime value is specified in seconds, where one day is equal to 86400 seconds and two weeks is equal to 1209600 seconds.

The default value is **1209600** (two weeks).

**Note**
This option requires the **FilenamePrefix** option to be set since it uses the prefix as validation of ownership to prevent the accidental removal of files.

**LifetimeCheck**
Specifies to scan the **StagingProcessDownloads** directory for files older than the **CopyOutLifetime** value.

Valid values are **0** (off) or **1** (on). The default value is **0**.

**LifetimeCheckInterval**
Specifies how often to scan the **StagingProcessDownloads** directory for files older than the **CopyOutLifetime** value. If the directory holds many files and it is not important to check the lifetime each time the **analyze_managed_product.pl** script is run, you can improve performance by increasing the value so the check is made less frequently. If the specified value is 10, the lifetime check occur once over the course of 10 script executions.

The default value is **10**.
**LifetimeProcessLimit**
Specifies the maximum number of seconds the file lifetime check is allowed to continue. The lifetime check randomly examines cached files. If the StagingProcessDownloads directory consists of many files, this option has the effect of randomly processing a subset of files each time the lifetime check takes place. Over time, all of the files are examined.

The default value is **300**.

**LogTypes**
Specifies the type of logging to be reported. The following are valid log types:

- NONE
- ERROR
- WARNING
- INFORMATION
- DEBUG
- PERFORMANCE
- ALL

**Note**
Use the + sign to use multiple log types. For example, ERROR+WARNING.

**RulesObject = Requirement Rules from ClearanceDB Server**
Specify to perform analysis using the product's rules and conditions from the ClearanceDB database. To enable this option, remove the number sign symbol (#) that precedes the RulesObject = Requirement Rules from ClearanceDB Server line.

**RulesObject = Variant Analysis**
Specify to perform analysis using Teamcenter variant conditions. To enable this option, remove the number sign symbol (#) that precedes the RulesObject = Variant Analysis line.

**ResultsObject = ClearanceDB Results**
Specify to generate a ClearanceDB results file (ClearanceResultsDbUpload.csvcldb). To enable this option, remove the number sign symbol (#) that precedes the ResultsObject = ClearanceDB Results line.

**Attribute=**__PLM_ABSOCC_UID, Part and Parents (Part First)**
Specify to use product structure absolute occurrences. To enable this option, remove the number sign symbol (#) that precedes the Attribute=__PLM_ABSOCC_UID,Part and Parents (Part First) line.

3. Save the file.
Specify to use absolute occurrence IDs

For ClearanceDB to work with data from Teamcenter, an absolute occurrence ID is required for each product structure node of the BOM line to be analyzed.

1. Enable absolute occurrence IDs in the global configuration options (specified in the Clearance.cfgglobal file) by removing the number sign symbol (#) that precedes the Attribute=__PLM_ABSOCC_UID,Part and Parents (Part First) line.

2. Add the following entry to the process configuration (specified in the configuration.csvcldb file) for each product:

   "000092/A;1-cottonpicker (View)",EAICL_RELEVANT_PART_ATTRIBUTE,__PLM_ABSOCC_UID

Configure the Clearance.cfgproduct file

For the Clearance Calculator to analyze Teamcenter managed data, you must specify the item ID and revision of the top level assembly of your product. If your Teamcenter data is configured with multifield key data, you must specify the item key.

1. Use a text editor, open the Clearance.cfgproduct file for the product and provide the following information:

   **ItemKey**
   Specifies the Teamcenter values of the multifield key properties for the item containing your product.

   **Note**
   This setting is applicable only to Teamcenter products configured with multifield key data. It corresponds to the -key argument for the Teamcenter bomwriter utility.

   You can obtain the necessary multifield key strings with the Teamcenter get_key_string utility.

   **ItemId**
   Specifies the Teamcenter item ID of the top level assembly of your product.

   **ItemRevision**
   Specifies the Teamcenter revision of the top level assembly of your product.

2. Save the file.

Encrypt the Teamcenter user password

You can use the Teamcenter install utility to encrypt the Teamcenter user password. The encrypted password is stored in an external file.

1. In an environment configured to run Teamcenter utilities, open a command prompt.

   For information about configuring an environment to run Teamcenter utilities, see Manually configuring your environment for Teamcenter utilities in the Utilities Reference.
2. At the command prompt, type:
   
   ```
   install –encryptpwf –f=<path>
   ```
   
   where
   
   **path** is the full path and file name for the password file that you want to generate.

   **Example**
   
   ```
   install –encryptpwf –f=C:\ClearanceDB_Work_Area\account_key
   ```
   
   The `install` utility displays the following message:
   
   ```
   Please enter password:
   ```
   
   3. Type the password and press Enter.
   
   The utility displays the following message:
   
   ```
   Please re-enter the password:
   ```
   
   4. Type the password again and press Enter.
   
   The password is encrypted and saved to the specified location.

   5. Navigate to the ClearanceDB Work Area.

   6. Open the **Clearance.cfgglobal** file or a **Clearance.cfgproduct** file in a text editor.

   7. To specify for ClearanceDB to obtain the Teamcenter user password from the encrypted password file, locate the **TeamcenterUserPasswordFile** setting, and type the full path and file name for the password file.

   **Example**
   
   ```
   TeamcenterUserPasswordFile=C:\ClearanceDB_Work_Area\account_key
   ```

   8. Save the file.

**Configuring ClearanceDB for multifield key data**

Multifield keys are identifiers assigned to objects to ensure their uniqueness in the database. For example, if the item business object type is configured to use multifield keys, it is possible to have two item objects with the same item ID.

For ClearanceDB to analyze Teamcenter data configured with multifield key data, you must do the following:

- When running the `create_product.pl` script to **create the product in the ClearanceDB database**, append the `__PLM_ITEMREV_UID` for the item revision to the product name.

  To find the UID for an item revision, open the item revision in Structure Manager, and view the `bl_rev_fnd0objectid` property. Include this value within brackets after the (View) suffix, enclosing the entire product name in quotation marks, as shown below:
Example
“000092/A;1-cottonpicker (View) [4BYKHIYy$IoP]”

- When configuring the clearance.cfgproduct file for the product, populate the ItemKey parameter with the multifield key properties of the item. This setting corresponds to the –key argument for the Teamcenter bomwriter utility.

To obtain the key information, use the Teamcenter get_key_string utility.

Example
ItemKey=item_id=000092,MFK_strkey1=val1

Optionally configure Teamcenter export for 4GD worksets

When you export 4GD data to PLM XML, you can export design element properties that are in addition to the properties that are exported by default. To do this, you must add the properties to the PLM XML transfer mode. You may also customize the titles of the properties for the PLM XML file.

Add the property to the PLM XML transfer mode

To include the additional property, you must edit the transfer mode used for exporting data from Teamcenter by adding the property derived from the 4GD root business object Cpd0DesignElement.

1. In Teamcenter, start the PLM XML/TC XML Export Import Administration application.

   - Note
   More information on the PLM XML/TC XML Export Import Administration application is located in the Teamcenter help collection.

2. Select the 4GDPIEDataExportTcVisClearanceDB transfer mode from the PLMXML Import Export Modes tree.
   - The TransferMode pane appears, with the options for the selected transfer mode.

3. Expand 4GDPIEDataExportTcVisClearanceDB in the left pane, and select 4GDPIEDataExportCLDB_PS.

4. Create the clause that specifies how the data is traversed:
   - a. Click the Add clause button located to the right of the clause table.
   - b. Select CLASS from the Primary Object Class Type list.
   - c. Type Cpd0DesignElement in the Primary Object box.
   - d. Select PROPERTY from the Relation Type list.
   - e. Type object_name in the Related Property Or Object box.
f. Select **DO** from the **Property Action Type** list.

g. Click the **Modify** button.

Customize the property title in the PLM XML file

When the property is added to the property set, it is exported to the PLMXML with its name as `object_name`. You can change this to a more specific name.

1. Open the rule file `4GD_cldb.csv` at this location:

```
TC_DATA\xformer\LLTCXML_PLMXML
```

2. Under the **Cpd0DesignElement** business object, in the **ProductRevisionView** section, add the **ProductRevisionView/UserData** rule as shown in this example, where `DesignElementName` is the new name you want to create for the property.

```plaintext
# Cpd0DesignElement

Cpd0DesignElement|ProductInstance
Cpd0DesignElement.object_name|ProductInstance.name
Cpd0DesignElement<fnd0affectedObject.Mdl0VariantExprBlock.fnd0expressions>
   Cfg0ApplicationVariantExp.fnd0formula|ProductInstance/UserData|
   add_user_value($TARGET_END_OBJECT
Cpd0DesignElement.owning_user>User.user_name|ProductInstance/UserData|
   add_user_value($TARGET_END_OBJECT
Cpd0DesignElement.owning_group>Group.name|ProductInstance/UserData|
   add_user_value($TARGET_END_OBJECT
Cpd0DesignElement|ProductInstance.partRef>ProductRevisionView
Cpd0DesignElement|Occurrence
Cpd0DesignElement.puid|Occurrence/UserData|add_user_value($TARGET_END_OBJECT
Cpd0DesignElement|Occurrence.instanceRefs>ProductInstance
Cpd0DesignElement<cpd0presented_parent.Cpd0DesignElement|
   Occurrence.instanceRefs>Occurrence
Cpd0DesignElement|Occurrence.partRef>ProductRevisionView
Cpd0DesignElement.cpd0UG_ENTITY_HANDLE|Occurrence/UserData|
   add_user_value($TARGET_END_OBJECT
Cpd0DesignElement|ProductRevisionView
Cpd0DesignElement.object_name|ProductRevisionView.name
Cpd0DesignElement.object_name|ProductRevisionView/UserData|add_user_value
   ($TARGET_END_OBJECT, 'DesignElementName', '',$SOURCE_VALUES, false, false)
Cpd0DesignElement.cpd0source_object>ItemRevision.items_tag>Item.fnd0is_monolithic|ProductRevisionView/UserData|add_user_value($TARGET_END_OBJECT
Cpd0DesignElement<cpd0presented_parent.Cpd0DesignElement|ProductRevisionView.
   instanceRefs>ProductInstance
Cpd0DesignElement<mdl0positioned_object.Mdl0DefaultGeometry|
   ProductRevisionView/Representation
Cpd0DesignElement<mdl0positioned_object.Mdl0DefaultGeometry.
   mdl0direct_model_data>DataSet.object_type|ProductRevisionView/
   Representation.format|lookup_table( DSFormatList
```
3. Near the beginning of the file, add the ProductRevisionView rule as shown in this example.

```plaintext
!source_hierarchy Cpd0WorksetLine\BOMLine
!source_hierarchy Cpd0SubsetLine\BOMLine
!source_hierarchy Cpd0WorksetRevision\ItemRevision

!target_graph_definition UserValue.title

!target_graph_definition UserValue.value
!target_id_reference InstanceGraph.rootRefs|*.id
!target_id_reference ProductRevisionView.instanceRefs|*.id
!target_id_reference ProductView.rootRefs|*.id

|ProductView/ApplicationRef.application| 'NX'|
   @ProductView.usage == 'modelView'

|InstanceGraph.$CHILD_ELEMENT|concat(@@InstanceGraph.$CHILD_ELEMENT>$MAPPED
|InstanceGraph.$CHILD_ELEMENT|concat(@@InstanceGraph.$CHILD_ELEMENT>$MAPPED

|ProductView.$CHILD_ELEMENT|concat(@@ProductView.$CHILD_ELEMENT>$MAPPED
|ProductRevisionView/UserData/UserValue| remove_object($SOURCE_VALUE)|
   @UserValue.title == 'object_name'
```

Understanding clearance variant analysis

Clearance variant analysis is based on product variant information stored in Teamcenter. Analysis is performed on a 150% BOM structure, which is the union of all of the structure of a product without respect to buildable combinations. Non-buildable combinations of product structure are detected and excluded from the clearance analysis.

The benefits of removing non-buildable occurrence relationships from ClearanceDB analysis include the following:

- A reduction in clearance calculations
- Overall faster clearance analysis
- Fewer clearance incidents for engineers and designers to evaluate
A simple example with variant conditions (150% BOM with variants)

Using variant analysis, a single 150% BOM can contain all buildable product combinations. This enables you to perform a single clearance analysis, which excludes all non-buildable part pairs.

With no variants, all nodes are analyzed (15 part pairs).
No nodes are excluded.

With the E=V6 variant, nodes 1, 2, 3, and 4 are analyzed (6 part pairs).
Nodes 5 and 6 are excluded (9 part pairs).

With the E=V8 variant, nodes 1, 2, 5, and 6 are analyzed (6 part pairs).
Nodes 3 and 4 are excluded (9 part pairs).
A simple example without variant conditions (separate 100% BOMs)

Without using variant analysis, you must create and manage separate 100% BOMs to represent each buildable product combination. You must evaluate each BOM with a separate clearance analysis.

For the analysis of the V6 Engine, without variant conditions, nodes 1, 2, 3, and 4 are analyzed (6 part pairs).
No nodes are excluded.

For the analysis of the V8 Engine, without variant conditions, nodes 1, 2, 5, and 6 are analyzed (6 part pairs).
No nodes are excluded.

For more information about working with Teamcenter variant conditions, see Getting Started with Product Structure in the Teamcenter help collection.

Enable variant analysis for clearance

You must enable the option to evaluate ClearanceDB products according to Teamcenter variant conditions.

1. Using a text editor, open Clearance.cfgglobal and adjust the following settings:

   **Note**
   You can also specify these options in the Clearance.cfgproduct file for the product.

   **LexicographicalVariantAnalysis**
   Defines the mode of the evaluation of variant condition relational expressions. Choose one of the following:
**No** — If undeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

**Yes** — If undeterminable variant conditions exist, evaluate them lexicographically.

**Note**
Variant conditions that cannot be evaluated based on textual representations are undeterminable.

**VariantAnalysisClientTraceFile**
Specifies the name for the log file of the variant analysis. By default, this is not set, and a log file is not generated.

**Example**
```
VariantAnalysisClientTraceFile=cldb_variant_analysis.log
```

**VariantAnalysisClientTraceFlags**
Specified the content of the log file of the variant analysis. The trace flags are additive, concatenated with the plus (+) character.

**Example**
```
VariantAnalysisClientTraceFlags=CONFIGURATION+STATISTICS
```

Add any of the following flags:

- **CONFIGURATION**
  Lists the variant analysis configuration attributes.

- **STATISTICS**
  Provides the basic variant analysis statistics.

- **VARIANTCONDITIONS**
  Provides details of textual variant conditions.

- **DETERMINABILITY**
  Lists the determinability of variant conditions.

- **SATISFIABILITY**
  Lists the satisfiability of variant conditions.

- **EXCLUDEDVCPAIRS**
  Lists excluded variant condition pairs.

- **EXCLUDEDUIDPAIRS**
  Lists excluded clearance element pairs as ABSOCCs.

- **EXCLUDEDNGIDPAIRS**
  Lists excluded clearance element pairs as NGIDs.

- **IMPACT**
  Lists the impact of variant conditions on the product structure.

**Caution**
Enabling the **EXCLUDEDUIDPAIRS** and **EXCLUDEDNGIDPAIRS** trace flags may result in extremely large log files.
**VariantLogicalExpression**
Define the variant condition UserValue title and BOM line property pair in the PLM XML file. This has the form of:

“Variant Condition”:bl_variant_condition

“Variant Formula”:bl_formula

**Example**

```plaintext
VariantLogicalExpression="VC-V6:bl_variant_condition"
```

**Note**
This must match the content of the **BomWriterUserAttributes** setting.

**BomWriterUserAttributes**
Specify the variant conditions for the **bomwriter** to include in the generated .plmxml file. Type these according to the following syntax:

```plaintext
target:Instance,key:myAttribute,literal:"My Attribute Value"
```

**Note**
This is the same option as the **bomwriter -ua** command line argument.

**Example**

Use this option to include variant model related BOM line properties in the generated .plmxml file. For example:

```plaintext
BomWriterUserAttributes=target:Instance,key: VC-V6,prop: bl_variant_condition
```

**RulesObject = Variant Analysis**
Specify to perform analysis using Teamcenter variant conditions. To enable this option, remove the number sign symbol (#) that precedes the **RulesObject = Variant Analysis** line.

2. Save the file.

**Caching managed data for analysis**

You can configure ClearanceDB to use the **load_fcccach** utility to download and cache model data from the Teamcenter server to your local machine, improving both the reliability and performance of the analysis. Options to control the **load_fcccach** utility are located in the **Clearance.cfgglobal** file.
For most of these settings the default values are recommended and they should not be modified unless you understand how the changes will affect the data staging process. Exceptions include:

- **JtDataStagingProcess** — By default the `load_fcccache` option is disabled and you must change this setting to enable it.

- **CopyOutLocation** — This setting specifies where to copy the files cached from Teamcenter.

- **LogTypes** — This setting controls the content of the log files.

For more information on the Teamcenter File Management System (FMS), see the System Administration Guide in the Teamcenter documentation.

1. Using a text editor, open `Clearance.cffglobal` and adjust the following settings:

   - **JtDataStagingProcess**
     Specifies to use the Teamcenter `load_fcccache` utility to download the model data from the Teamcenter server to the local system for clearance analysis. Enabling this option pre-populates the FMS client cache (FCC), which leads to faster and more reliable analysis.

     Valid values are 0 (off) or 1 (on). The default value is 0.

   - **JtDataStagingProcessErrorLimit**
     Specifies when to abort the `analyze_managed_product.pl` script, based on the following custom error codes:

     ```
     FILECOPY_FAILED
     INVALID_DAKID_FORMAT
     FCC_OPENFILE_FAILED
     FCC_DOWNLOAD_FAILED
     GET_READ_TICKET_FAILED
     DATASET_READ_FAILED
     PLMXML_MISSING_JT
     CHMOD_FAILED
     COPYOUT_CLEANUP_FAILED
     ```

     You can specify a numerical value for each error code. By default, the error codes are given values that correspond to the severity of the problem, with the lowest value representing the most severe failure. If a value is not specified for the `JtDataStagingProcessErrorLimit` setting, the script stops whenever an error occurs during the staging process.

     The default value is 30.
**Note**

The default values are recommended for the `JtDataStagingProcessErrorLimit` setting and the related error codes. Essentially, with these defaults the clearance analysis will abort when any of these errors occur, and you can use the reported error message to troubleshoot the problem.

**FILECOPY_FAILED**
Specifies the error code that indicates a copy operation to the output directory has failed.
The default value is 20.

**INVALID_DAKID_FORMAT**
Specifies the error code that indicates an invalid DAKID was found.
The default value is 21.

**FCC_OPENFILE_FAILED**
Specifies the error code that indicates the FCC failed to open the file using the ticket.
The default value is 22.

**FCC_DOWNLOAD_FAILED**
Specifies the error code that indicates a failure most likely due to a missing file in the volume.
The default value is 23.

**GET_READ_TICKET_FAILED**
Specifies the error code that indicates a read ticket failed.
The default value is 24.

**DATASET_READ_FAILED**
Specifies the error code that indicates no read access on the dataset.
The default value is 25.

**PLMXML_MISSING_JT**
Specifies the error code that indicates a missing JT file reference in the .plmxml file.
The default value is 26.

**CHMOD_FAILED**
Specifies the error code that indicates a failure to set the access mode during copy out.
The default value is 27.

**COPYOUT_CLEANUP_FAILED**
Specifies the error code that indicates a failure to remove a file during lifetime cleanup.
The default value is 28.

**CopyOutLocation**
Specifies the location for the dataset files downloaded from Teamcenter server.
The default location is the product directory in the ClearanceDB Work Area.
Note
Do not use special characters in folder names.

**BucketCount**
Specifies how many directories to use for the cached files. Spreading the files over multiple directories can lead to better performance.

The default value is **30**.

**UseAbsoluteLocation**
Specifies to use an absolute value for the location attribute in the .plmxml file generated by the bomwriter. It is usually better to have a relative reference, although there are instances where an absolute reference is required, such as when the .plmxml file is moved to a different location from the referenced files.

The default value is **No**.

**DirAccessMode**
Specifies the access mode setting for the directories created to hold the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**FileAccessMode**
Specifies the access mode setting for the cached files. Use a chmod octal value. This setting is used only on UNIX and Linux systems.

The default value is **0640**.

**BucketPrefix**
Specifies a prefix to add to the names of directories created to hold the cached files.

The default value is **RW**.

**FilenamePrefix**
Specifies a prefix to add to the names of the cached files.

The default value is **fmsr_**.

**CopyOutLifetime**
Specifies the lifetime of the files cached in the **StagingProcessDownloads** directory. The directory is scanned for files older than the specified value, which are removed. The lifetime value is specified in seconds, where one day is equal to 86400 seconds and two weeks is equal to 1209600 seconds.

The default value is **1209600** (two weeks).

Note
This option requires the **FilenamePrefix** option to be set since it uses the prefix as validation of ownership to prevent the accidental removal of files.
LifetimeCheck
Specifies to scan the StagingProcessDownloads directory for files older than the CopyOutLifetime value.
Valid values are 0 (off) or 1 (on). The default value is 0.

LifetimeCheckInterval
Specifies how often to scan the StagingProcessDownloads directory for files older than the CopyOutLifetime value. If the directory holds many files and it is not important to check the lifetime each time the analyze_managed_product.pl script is run, you can improve performance by increasing the value so the check is made less frequently. If the specified value is 10, the lifetime check occur once over the course of 10 script executions.
The default value is 10.

LifetimeProcessLimit
Specifies the maximum number of seconds the file lifetime check is allowed to continue. The lifetime check randomly examines cached files. If the StagingProcessDownloads directory consists of many files, this option has the effect of randomly processing a subset of files each time the lifetime check takes place. Over time, all of the files are examined.
The default value is 300.

LogTypes
Specifies the type of logging to be reported. The following are valid log types:

   NONE
   ERROR
   WARNING
   INFORMATION
   DEBUG
   PERFORMANCE
   ALL

Note
Use the + sign to use multiple log types. For example, ERROR+WARNING.

2. Save the file.

Analyzing managed ClearanceDB products
Analysis of managed products is performed via the analyze_managed_product.pl script, which triggers the following events in sequence:

1. The bomwriter utility generates a .plmxml file referencing the Teamcenter product data.
2. The load_fcccache utility downloads the product data to your local machine. A second .plmxml file referencing the cached data is generated.

3. The Clearance Calculator analyzes the cached data according to a ClearanceDB product configuration (defined in the configuration.csvclddb file, together with any specified rules, conditions, and zones). ClearanceDB results are generated and uploaded to the ClearanceDB database.

Analyze a managed product

Take the following steps to perform a ClearanceDB analysis upon a Teamcenter managed product.

1. **Configure** the Clearance Calculator.

2. Navigate to the appropriate product directory in your ClearanceDB Work Area.

3. From the command prompt, type

   ```plaintext
   analyze_managed_product.pl <product>
   ```

   **Example**

   ```plaintext
   analyze_managed_product.pl flux_capacitor
   ```

   Depending upon how you set the RunLevel parameter in the Clearance.cfgglobal file, one or more of the following actions are performed:

<table>
<thead>
<tr>
<th>This RunLevel setting</th>
<th>Does this</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the product directory, a.vvi file is created, which is used by the bomwriter to generate a .plmxml file referencing the managed data.</td>
</tr>
<tr>
<td>2</td>
<td>Using the .vvi and .plmxml files in the product directory, the Clearance Calculator performs analysis upon the managed product data and generates a results file.</td>
</tr>
<tr>
<td>3</td>
<td>The results file is uploaded to the ClearanceDB database.</td>
</tr>
<tr>
<td>4</td>
<td>Both the RunLevel 1 and 2 actions are performed.</td>
</tr>
<tr>
<td>5</td>
<td>Both the RunLevel 2 and 3 actions are performed.</td>
</tr>
<tr>
<td>6</td>
<td>All of the RunLevel actions are performed, 1, 2, and 3.</td>
</tr>
</tbody>
</table>

   The analysis runs, and when it is complete the clearance results file, ClearanceResultsDbUpload.csvclldb, appears in the product directory.
Example output from managed analysis

The command line output below is from a successful ClearanceDB analysis of a Teamcenter managed product.

C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\analyze_managed_product.pl

---
+ analyze_managed_product.pl
+ Tue Feb  9 14:01:45 2010
---

---
+ Teamcenter Configuration:
+ TC_ROOT=Y:\
+ TC_DATA=Z:\
+ TeamcenterUserId=tcuser
+ RevisionRule=Latest Working
+ RunLevel=6
+ JtDataStagingProcess=1
---

"141-f10002_A_1-Front_Suspension_view.vvi" file is created in the product directory. Please verify it.

---
+ Step 1a of 3: Run Bomwriter.exe
---

Running:
Y:\bin\bomwriter -u=tcuser -p=xxxxxxxx -g = -i=item=141-f10002" -rev=A -format=plmxml+type=
AbsoluteOccurrences=ua=target:Root key:Configuration, literal:"Latest Working" -revision_rule="Latest Working" -output_file=C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\ 141-f10002_A_1-Front_Suspension_view.plmxml"

platform=MSWin32
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\bomwriter_exec.bat
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\set TC_ROOT=Y:\
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\set TC_DATA=Z:\
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\call Z:\\tc_profilevars

---
+ Bomwriter.exe execution SUCCEEDED
---

Execution time = 00:15:03

---
+ Run exe
---

Staging process uses config file: C:\ClearanceDB_Work_Area\Clearance.cfgglobal
The staging process saved the original plmxml as: C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\orig.plmxml
Warning: Invalid "CopyOutLocation" in the configuration file. Use product dir if by default.

Running:
Y:\bin\load_fcccache -u=tcuser -p=xxxxxxxx -config=C:\ClearanceDB_Work_Area\Clearance.cfgglobal -load -plmXml=C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\141-f10002_A_1-Front_Suspension_view_orig.plmxml -output_plmxml=C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\JtDataStagingProcess.log

platform=MSWin32
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\stage_exec.bat
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\set TC_ROOT=Y:\
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\set TC_DATA=Z:\
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\call Z:\\tc_profilevars

C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view\call Z:\\tc_profilevars

DEBUG 2010-02-09T19:21:16.509Z: Setting BucketCount to: 30
DEBUG 2010-02-09T19:21:16.509Z: Setting UseAbsoluteLocation to: No
Performing Loading Creating Launching 141-f10002_A_1-Front_Suspension_view_.vvi clearance.exe C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/C:\vis\Products\Mockup\C:\WINDOWS;C:\WINDOWS\System32;C:\vis\Products\Mockup\ClearanceDB C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/set CLDB_PROXY_CONNECT C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/set EAI_CLDatosource=Cldbmachine:7206 C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/set EAI_CL_CONNECT_DATA=Cldb_PROXY_CONNECT C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/set PATH=C:\Perl\bin;C:\WINDOWS\System32;C:\vis\Products\Mockup\ClearanceDB C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/C:\vis\Products\Mockup\clearance.exe -s 1 -c C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension_view_/141-f10002_A_1-Front_Suspension_view_.vvi


Running:
C:\vis\Products\Mockup\ClearanceDB\update_product.pl -ud
No local .dbc file in the product directory. Looking for global version(s) in ClearanceDB_Work_Area directory....

Uploading via ClearanceDB Proxy....
2010/02/09-14:31:09: Connecting to Clearance Database...
2010/02/09-14:31:09: Connected to ClearanceDB Proxy on machine cldbmachine, at port 7206.
2010/02/09-14:31:09: Successfully connected to Clearance Database.
2010/02/09-14:31:09: Lock the Clearance Database for results update...
2010/02/09-14:31:11: Starting upload for C:/ClearanceDB_Work_Area/Products/141-f10002_A_1-Front_Suspension_view/ClearanceResultsDbUpload.csvcldb...
2010/02/09-14:31:11: 100% complete.
2010/02/09-14:31:11: Starting merge results.
View results in the Lifecycle Viewer and Structure Manager

View results in the Lifecycle Viewer

1. In My Teamcenter, right-click the item revision containing your product data and choose **Send To→Lifecycle Viewer**.

   The dataset opens in the Lifecycle Viewer.

2. Choose **Clearance→Preferences→Requirement Components**.

3. In the **Requirement Rule Components** dialog box, select the **Requirement Rules from ClearanceDB Server** check box, and then click **OK**.

4. On the 3D Clearance toolbar, click **Load Results**.

5. In the **Load Clearance Results** dialog box, from the **Files of Type** list, choose **ClearanceDB DataBase Connection (*.dbc)**.

6. Select your DataBase Connection (DBC) file, and click **Open**.

   The 3D model and the database content are matched by the name of the top-level assembly node. Results are displayed in the **Results** list.

View results in Structure Manager

1. In My Teamcenter, right-click the item revision containing your product data and choose **Send To→Structure Manager**.

   The Structure Manager opens and displays the product structure.

2. Choose **View→Show/Hide Data Panel**.

3. In the Data Panel, click the **Viewer** tab.

   The visualization components load and the Structure Manager embedded viewer opens.

4. Choose **Graphics→Clearance→Preferences→Clearance→Requirements**.
5. In the Requirement Rule Components dialog box, select the Requirement Rules from ClearanceDB Server check box, and then click OK.

6. Right-click the embedded viewer toolbar area and choose 3D Clearance.
   The 3D Clearance toolbar is displayed.

7. On the 3D Clearance toolbar, click Load Results.

8. In the Load Clearance Results dialog box, from the Files of Type list, choose ClearanceDB DataBase Connection (*.dbc).

9. Select your DataBase Connection (DBC) file, and click Open.
   The 3D model and the database content are matched by the name of the top-level assembly node. Results are displayed in the Results list.

**Working with DesignContext**

You can combine ClearanceDB with Teamcenter's DesignContext application to create an Integrated Clearance Management (ICM) system. DesignContext identifies a series of target parts and then quickly finds other relevant data within a given proximity to those parts. This data can be sent to ClearanceDB to perform clearance analysis in database query (batch) mode or real-time mode. Results are displayed in DesignContext, and also can be sent to the Structure Manager, the Lifecycle Viewer, or standalone Mockup for display and further analysis.

ClearanceDB accepts any pruned or full product structure provided it is represented by a JT file in the Teamcenter database. Any parts for which JT data is not available are not included in the clearance analysis. The product structure also should be searchable in DesignContext.

The ICM system offers two clearance analysis modes:

- **Database query mode**
  In database query mode, the ClearanceDB administrator creates a batch process to perform clearance analysis periodically as a background task, typically every night. The batch process invokes the Clearance Calculator to perform the analysis and uploads the results to the ClearanceDB database. The DesignContext end user can then evaluate the results, identify issues, and assign the issues via a workflow to designers to analyze and resolve within other applications such as the Lifecycle Viewer, Structure Manager, NX, or other CAD package.

- **Real-time mode**
  In real-time mode, the rich client user selects target and background Bill of Materials (BOM) lines in DesignContext, and then invokes a clearance analysis that is performed in real time. Typically, this option is used to make on-the-fly clearance calculations after design changes or prior to releasing a part. Teamcenter runs these calculations in the background, allowing the end user to continue with other work while waiting for the results. Results are displayed in the DesignContext Issues panes.

A SCO (Structure Context Object) containing the clearance results can also be created in the Newstuff folder, provided the RDVCclearanceakesCOForClearance preference is set to true. The
SCO can be opened in a subsequent session to retrieve the selected product structure lines and analysis results.

**Note**
For more information on using DesignContext, see the *DesignContext Guide* in the Teamcenter documentation.

### DesignContext clearance analysis modes

#### Database query mode

The ClearanceDB administrator configures the scope of the analysis and the DesignContext end user can only view the results. The end user can use the DesignContext search engine capabilities to focus upon particular parts.

Clearance issues are displayed in one of the following **Issues** panes:

- **Target-Target Issues**
  Shows clearance issues among all the appearances in the DesignContext target appearances table in the third window. The specific issue between each pair of parts is listed.

- **Target-Background Issues**
  Shows clearance issues among target parts against all the appearances in the DesignContext background appearances table in the third window. The specific issue between each pair of parts is listed.

- **Target-Other Issues**
  Teamcenter displays each pair of parts that violate one of the defined clearance rules. All violations are shown in the context of the selected target part appearance or appearances. For each violation, the following information is listed:
    - **Type**
      The clearance violation type.
    - **Result**
      The calculated violation with respect to the requirement.
    - **Requirement**
      The required clearance.
    - **Location**
      The x,y,z coordinates of the violation.
Real-time mode

Real-time clearance analysis is a two step process. The first step is to create search criteria for the clearance calculation. The second step is to send the search criteria to DesignContext to see the clearance analysis results.

To perform real-time clearance analysis, the end user manually selects BOM lines of interest in DesignContext. BOM lines involved in the analysis process are DesignContext target parts. The user selects these parts from the target and background BOM lines that are configured in the DesignContext third window. Global background parts cannot be included in a real-time clearance analysis. If particular BOM lines are not selected, all BOM lines in the target appearances are considered target parts.

Teamcenter analyses the BOM lines selected as DesignContext target parts against all other BOM lines in the DesignContext third window. It presents the issues in the following categories:

• Target parts against all the appearances in the DesignContext target appearances table in the third window.

• Target parts against all the appearances in the DesignContext background appearances table in the third window.

 Optionally, Teamcenter may create an SCO (Structure Context Object) in the Newstuff folder containing information about the session. The SCO can be opened in a subsequent session to retrieve the selected product structure lines and analysis results.

Specify connection information for DesignContext

For DesignContext to connect with the ClearanceDB database, you must use the Teamcenter RDVClearanceProxyServers preference to specify the ClearanceDB connection information.

1. Log on to the rich client as the administrator.

2. Choose Edit→Options.
3. At the bottom of the Options dialog box, click Search.

4. In the Search On Keywords box, type RDVClearanceProxyServers, and then click the magnifying glass icon to search for the preference.

5. In the Preferences List, select RDVClearanceProxyServers.

6. In the Current Values box, type your connection information according to the following syntax:

   `<machine_name>:<port>:<connection_name>`

   where

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Type this</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine_name</td>
<td>The name of the system running the ClearanceDB Proxy or Oracle Client.</td>
</tr>
<tr>
<td>port</td>
<td>The port number used for the ClearanceDB connection.</td>
</tr>
<tr>
<td>connection_name</td>
<td>The name of the Oracle connection.</td>
</tr>
</tbody>
</table>

7. Click the + sign to the right of the Current Values box.

   **Note**

   The RDVClearanceProxyServers preference can consist of multiple ClearanceDB connections. For each additional connection that you want to establish, type the required connection information as described above and then click the + sign to the right of the Current Values box to update the preference.

8. Click Modify.

   **Specify to create an SCO for clearance results**

   Set the RDVCreateSCOForClearance preference to true if you want to create an SCO (Structure Context Object) after performing real-time analysis.

   **Note**

   You must have Teamcenter administrator rights to modify the RDVCreateSCOForClearance preference.

   1. Log on to the rich client as the administrator.

   2. Choose Edit→Options.

   3. At the bottom of the Options dialog box, click Search.

   4. In the Search On Keywords box, type RDVCreateSCOForClearance, and then click the magnifying glass icon to search for the preference.
5. In the **Preferences List**, select **RDVCreateSCOForClearance**.

6. In the **Current Values** box, type **true**.

7. Click **Modify**.

**Display additional clearance results information**

Modify the **ClearanceDBShownColumns** preference to display additional clearance results information in the DesignContext **Issues** panes.

```
Note
You must have Teamcenter administrator rights to modify the
ClearanceDBShownColumns preference.
```

1. Log on to the rich client as the administrator.

2. Choose **Edit→Options**.

3. At the bottom of the **Options** dialog box, click **Search**.

4. In the **Search On Keywords** box, type **ClearanceDBShownColumns**, and then click the magnifying glass icon to search for the preference.

5. In the **Preferences List**, select **ClearanceDBShownColumns**.

6. In the **Current Values** box, type the names of the ClearanceDB results columns that you want to display.

7. Click **Modify**.

**Load ClearanceDB results in DesignContext**

1. Choose **Tools→Clearance Analysis**.

2. In the **Clearance Analysis** dialog box, select **Query the database for existing results**.

3. Click **Start Analysis**.

   After a delay that depends on the quantity of stored data, Teamcenter displays the retrieved clearance issues in one of the **Issues** panes.

4. To view the associated parts in the embedded viewer, select one or more issues and click the **Communicate to DesignContext** button.

   Teamcenter opens the relevant BOM lines in DesignContext, allowing you to examine the clearance details in the embedded viewer.

   Alternatively, you can also view the parts associated with one or more selected issues in standalone Teamcenter lifecycle visualization mockup if you click the **Start/Open In TcVis** button.
Perform real-time analysis

1. Select the product structure lines of interest in the third DesignContext window. If you do not select any lines, Teamcenter includes all appearances in the third DesignContext window in the analysis.

2. Choose Tools→Clearance Analysis.

3. In the Clearance Analysis dialog box, select Perform Real-Time Clearance Analysis.

4. Click Start Analysis.

   A progress bar shows the percentage completion of the calculation, and you can click Cancel above the progress bar to terminate the process if necessary. After a delay that depends on the scope of the query, Teamcenter displays the clearance issues in one of the Issues panes.

5. Click OK on the form.

   **Note**
   
   Real-time analysis results are not stored in the ClearanceDB database. If you want to preserve them, you must create an SCO (Structure Context Object) object. You can open the SCO in a subsequent session to retrieve the selected product structure lines and analysis results. The RDVCreateSCOForClearance preference must be set to true for an SCO to be created.

Open clearance results in an SCO

1. In your Newstuff folder, right-click the SCO containing your results and choose Send To→Design Context.

   Teamcenter opens the results in the first DesignContext window.

2. Click Finish.

   Teamcenter opens the third DesignContext window.

   **Note**
   
   Click Yes on any confirmation dialog boxes that are displayed.

3. Choose Tools→Clearance Analysis.

   Teamcenter displays the Clearance Analysis dialog box.

4. Select Display Cached Results from SCO and then click Start Analysis.

   After a delay that depends on the quantity of stored data, Teamcenter displays the retrieved clearance issues in one of the Issues panes.
Chapter 12: Maintaining the database

Database maintenance overview

This section provides information on maintaining and optimizing the database.

The administrator may choose to execute stored procedures to delete obsolete objects, delete all data associated with a product, copy all data associated with a product, or rename a product. In addition, ClearanceDB automatically performs certain database maintenance functions to assure the integrity of the data persisted in the ClearanceDB Oracle database.

Understanding the Simple Security Model

The Simple Security Model describes the supported modes of viewing and updating the ClearanceDB database.

The viewing of clearance issues is unrestricted provided that the user can access the product JT data. If the user can open the main assembly file that defines the product, the pertinent clearance results can be loaded and visualized.

The updating of clearance issues in the Mockup client is unrestricted. The user can modify the following clearance issue attributes:

- Resolution (includes Comment, Change Required?, Change Reference, and Item Important?)
- Owner
- Status
- Priority
- Zone

The user can modify these attributes in the Mockup Results list and save them locally in a text file. Unrestricted local persistence provides for individual flexibility of data handling and does not affect other users.

The ability to commit local clearance issue modifications to the database can be restricted and controlled by the database configuration per product and the content of the Owner field per clearance issue.

For any given product, the database checks to see if the EAICL_UPDATE_SECURITY attribute value is set to ON. If it is, then the database checks for the EAICL_PRODUCT_OWNER attribute. If the user's identity matches this value, the user's modifications to the clearance issue status are committed to the database. If EAICL_UPDATE_SECURITY is set to OFF, or not present in the product's configuration file, any user can commit changes to the issue.
Mockup users are identified to the database by logname (UNIX) or user name (Windows). A number of users such as Oracle User accounts should be allowed to update all the issues pertaining to a specific product. Additionally, users identified in the Owner field of a clearance issue (Issue Owners) are always allowed to modify the attributes of that issue.

Oracle User accounts are designated during the product configuration using the EAICL_PRODUCT_OWNER attribute. Issue Owners can be designated in the following ways:

- By the database itself based on part metadata (for new issues only)

  **Note**

  For the database to automatically specify an Issue Owner based upon part metadata, the part attribute that defines the Issue Owner must be specified in the product configuration.

  For any part pairs involved in a clearance issue, you also have the option of specifying that the presence of a part attribute for one of the parts means that part takes precedence over the other part.

  For example, ownership may be based on metadata called "Engineer" with the precedence defined by the "Last Update" metadata. The "Engineer" of the part with the more recent "Last Update" date is designated as the Issue Owner.

- By the Oracle User account
- By the Current Issue Owner

Possible scenarios for these last two options include:

- Completely Manual - No Issue Owners are pre-assigned by the database. All designations are done by Oracle User accounts.

- Partially Automated, Partially Manual - Issue Owners are pre-assigned by the database when the metadata is available. Missing designations are done by Oracle User accounts.

- Mostly Automated - Issue Owners are pre-assigned by the database with the metadata available reliably. Some designations are done by Oracle User accounts or current Issue Owners.
Automatic database maintenance

ClearanceDB automatically performs certain database maintenance functions to assure the integrity of the data persisted in the ClearanceDB Oracle database. These functions include:

- Running the eaicl_p_clean procedure.
- Monitoring both database space and extent allocations, and providing additional resources or reporting deficiencies.
- Monitoring the fragmentation of indexes and rebuilding them as required.

The completeness and integrity of the database objects is verified upon all major transactions, including:

- Upgrading the ClearanceDB Oracle application.
- Uploading product configurations.
- Uploading clearance rules and conditions.
- Uploading zone definitions.
- Uploading clearance results.
- Updating clearance issue dispositions.
- Running stored procedures to delete, rename, or copy product data.

The following tablespaces are checked for their type and size:

- Table
- Index
- Rollback (including the existence of the RHUGE rollback segment - Oracle 8i only)

The existence of the following database objects is checked for:

- Tables
- Indexes
- Views
- Sequences
- Triggers
- Procedures and functions
- Packages and their bodies

The following relationships are checked for integrity:

- Current values of sequences and the values of corresponding primary keys
ClearanceDB also checks for the completeness of the clearance results upload file. In addition, the content of the results is examined for the following:

- Non-unique CADIDs
- The presence of geometry change attributes on all parts
- The presence of attributes referenced by clearance requirements rule conditions
- The timestamp of the clearance analysis

**Execute stored procedures**

To manually perform maintenance on the ClearanceDB database, take the following steps:

1. From the command prompt, type SQLPLUS, and logon as the ClearanceDB user.
2. Type any of the following commands:

<table>
<thead>
<tr>
<th>Type</th>
<th>To do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>exec eaicl_p_clean</td>
<td>Delete obsolete objects from the database.</td>
</tr>
<tr>
<td>exec eaicl_p_reset_gc_attr</td>
<td>Make subsequent applications absolute rather than incremental.</td>
</tr>
<tr>
<td>(eaicl_p_product.product_id ('&lt;product name&gt;')</td>
<td>Note: ClearanceDB automatically runs eaicl_p_clean when it is necessary to do so. Although you can manually execute the procedure, it is better to let ClearanceDB do it for performance reasons.</td>
</tr>
<tr>
<td>exec eaicl_p_product.del_product</td>
<td>Delete all the data associated with a product.</td>
</tr>
<tr>
<td>(eaicl_p_product.product_id ('&lt;product name&gt;'), '&lt;user_name&gt;')</td>
<td></td>
</tr>
<tr>
<td>exec eaicl_p_product.ren_product</td>
<td>Rename a product. Only the name of the product changes.</td>
</tr>
<tr>
<td>('&lt;old_product name&gt;', '&lt;new_product name&gt;', '&lt;user_name&gt;')</td>
<td></td>
</tr>
<tr>
<td>exec eaicl_p_product.cpy_product</td>
<td>Copy a product. All associated data is copied, except the issue modification/deletion history.</td>
</tr>
<tr>
<td>('&lt;old_product name&gt;', '&lt;new_product name&gt;', '&lt;user_name&gt;')</td>
<td></td>
</tr>
</tbody>
</table>

**Enforcing the integrity of input data**

The ClearanceDB database assumes that input data such as the clearance requirement rulebase or a product's JT data reflect modeling intent. However, due to the complexity of both the product and
requirement models, it is quite feasible that the models presented to the ClearanceDB process result in data loss.

To help prevent this, the following input data properties can be detected, upon which subsequent results merges can be rejected:

- The absence of non-trivial clearance requirements, resulting from either a trivial rulebase, the complete coverage of exclusion rules, or from the failure of rule base retrieval
- A substantial (negative) change in the size of the product model or the rulebase, possibly due to the unintended loss of the input data

Enforcement of the input data integrity is configurable. All exceptions are logged. The following configuration attributes are available:

- EAICL_PRODUCT_STRUCTURE_CHANGE_TEST - (ON/OFF)
- EAICL_PRODUCT_STRUCTURE_CHANGE_REJECTION_PERCENT - (1-100)
- EAICL_RULEBASE_CHANGE_TEST - (ON/OFF)
- EAICL_RULEBASE_CHANGE_REJECTION_PERCENT - (1-100)
- EAICL_NON_TRIVIAL_REQUIREMENT_TEST - (ON/OFF)

**Note**

The REJECTION_PERCENT variable defines the threshold of the decrease in size of the product structure or the rulebase above which the product or the rulebase update will be rejected. The size of the product structure is measured in part occurrences. The size of the rulebase is measured in rules and conditions. The greater of the change in the rules and the change the conditions is compared against the threshold.

**Available status reports**

The following database status reports are available:

- Database Installation Summary - The installation success indicator
- Database Installation Detail - Database object after-installation status report, possibly indicating installation problems per individual objects
- Database Status Summary - The database object status summary
- Database Status Detail - Database object status report, possibly indicating post installation problems per individual objects

Three reports are available in regard to the ClearanceDB results upload file:

- Non-unique CADID History - All non-unique CADIDs and their cardinality per product update
• Missing Geometry Change Metadata History - All missing metadata per part occurrence and their cardinality per product update

• Missing Requirement Metadata History - All metadata references in the product requirement rulebase that are not attached to any of the parts per product update

**Note**

The history report retention is controlled by the EAICL_PRODUCT_HISTORY_RETENTION_DAYS configuration variable.

ClearanceDB maintains four enhanced exception logs:

• Application installation log collecting installation events/exceptions

• Database exception log collecting application exceptions

• Application event log collecting application corrective actions

• User data exception log collecting user data exceptions

These logs are configurable and available as the Database Event History report. All of the event logs have the following structure:

• EVENT_ID - unique sequential number

• TIMESTAMP - the (server) date/time of the event

• PRODUCT_ID - product id/product name

• CONFIGURATION_ID - configuration id/ product configuration name

• PRODUCT_EVENT - e.g. product update, change in user clearance issue disposition

• DATABASE_EVENT - e.g. insertion of a record

• DATABASE_OBJECT - e.g. table name

• DESCRIPTION - verbose description of the event

**Note**

The event log retention is controlled by the EAICL_DATABASE_HISTORY_RETENTION_DAYS configuration variable. All reports are sortable and subject to the ClearanceDB Reports Time Period filter.
Chapter 13: Troubleshooting

Loading results in the viewer

In the viewer, if the user cannot connect to the ClearanceDB database to obtain results, it may be because the ClearanceDB results components are not turned on.

Ask the user to turn on the ClearanceDB results components:

1. In the viewer, choose Analysis tab→Clearance group→Enable Clearance Analysis.
2. In the Clearance group, choose Clearance Preferences→Requirement Components.
4. Click OK

Frozen Clearance Results window

In the viewer, if the Clearance Results window locks up, a results upload process may be operating in the background.

Ask the user to wait for a few minutes and try again. If the results window does not become active, there may be a problem with the network or the ClearanceDB database.

Clearance results disappear after user deletes a filter

In the viewer, the user applies a filter to clearance results displayed in the Clearance Results window, and then deletes it. The user expects the original list of results to appear in the window, but the window is blank.

This occurs because the clearance results list does not automatically re-post results after an applied filter is deleted. Ask the user to take the following actions:

1. In the Clearance Results window, right-click, and select Filter.
2. Click Add.
3. Select Column = Number.
5. Select Value = -1.
6. Click OK.
7. For Filter Input, choose All Items.

8. For Filter Action, choose Show Only.

9. Click OK.

The clearance results should appear again.

**Viewer licensing in Teamcenter**

To work with ClearanceDB results in Teamcenter, the rich client, you must have a license for the Mockup product configuration with the Analysis option. Licensing for the Lifecycle Viewer and the Teamcenter embedded viewers is specified in the `client_specific.properties` file, which is located in the ..\rac\plugins\configuration_8000.0.0 directory on client machines.

**Example**

This example shows the visualization licensing parameters in the `client_specific.properties` file set to run the Mockup product configuration, with all available options:

- `PortalViewer_License_Level=Mockup`
- `PortalViewerOptional_Licenses=ECAD,Analysis,Path_Planning,Concept_Desktop,Collaboration,ClearanceDB`

**Viewer preferences and settings**

Multiple users performing real-time analysis on the same product may receive inconsistent results if the following viewer preferences and settings are configured differently:

- **PLMXML Units** — Specifies the unit of measurement for PLM XML files loaded in the viewer. This option is located on the File tab→Preferences→PLM XML menu.

- **Calculator Settings** — Specifies clearance options such as the element type. These settings are located from Analysis tab→Clearance group→Clearance Preferences → Calculator Settings.

**Perl compatibility and configuration**

ClearanceDB requires Perl version 5.03 or later. If Perl is not already installed on your system or you have a version earlier than 5.03, many sources are available for you to manually install the latest distribution, such as from [http://www.activestate.com](http://www.activestate.com). For conflicts that may arise due to the packaging of Perl with other applications, particularly Oracle in a Windows environment, this topic provides an overview of some possible Perl configurations and problem resolutions.
Preliminaries

Before proceeding, do the following to determine if Perl is already installed and specified in the system PATH environment variable:

1. Open a command prompt.

2. Type `perl -v` and then press Enter.
   
   If Perl is installed, the first line of the command line output displays the version, as shown below:
   
   ```
   This is perl, v5.8.0 built for MSWin32-x86-multi-thread
   (with 1 registered patch, see perl -V for more detail)
   
   Copyright 1987-2002, Larry Wall
   
   Binary build 806 provided by ActiveState Corp. http://www.ActiveState.com
   Built 16:19:14 Jun 19 2003
   ```

   Perl may be copied only under the terms of either the Artistic License or the GNU General Public License, which may be found in the Perl 5 source kit.

   Complete documentation for Perl, including FAQ lists, should be found on this system using `man perl` or `perldoc perl`. If you have access to the Internet, point your browser at http://www.perl.com/, the Perl Home Page.

   If Perl cannot be found, it is either not installed or not specified in the system PATH environment variable.

On systems with an Oracle installation, Perl may be installed but not included in the system PATH environment variable. Do the following to determine if Perl is installed in the Oracle product directory:

- From the command prompt, type `set PERL5LIB`.

   If the command line output states that PERL5LIB is not defined, Perl is most likely not installed on the system. If PERL5LIB is set, and its value includes an Oracle product directory, Perl is probably installed for use with the Oracle software but not specified in the system PATH environment variable.

Possible issues and workarounds

Perl is not installed

You need to install Perl if an installation location is not included in the system PATH environment variable and the PERL5LIB environment variable is also not defined. For Windows, you can download the latest ActiveState distribution from http://www.activestate.com. After installing a supported version of Perl, test the installation as described in the Preliminaries section of this topic to ensure that it is recognized by the system.

Perl is installed with Oracle software

If the PERL5LIB environment variable is set, and its value includes an Oracle product directory, Perl is probably installed for use with the Oracle software but not specified in the system PATH environment variable. If this version of Perl is 5.03 or later, you can use it for ClearanceDB. To configure your
system to use the Oracle Perl installation, you need add the location of the Perl executable to the system PATH environment variable and then associate the .pl file extension with it.

Do the following to add the location of the Perl executable to the system PATH environment variable:

1. From the command prompt, type `set PERL5LIB`.
   The command line output displays the value of the PERL5LIB environment variable, including the path to the Oracle Perl installation.

2. Do the following to add the location of the Oracle Perl installation to the system path:
   a. Right-click **My Computer** and select **Properties**.
   b. In the **System Properties** dialog box, click the **Advanced** tab.
   c. On the **Advanced** page, click **Environment Variables**.
   d. On the **Environment Variables** page, in the **System variables** section, find **PATH** and modify it to include the path to the Oracle Perl installation, as shown in the value of the PERL5LIB environment variable.
   e. Click **OK**.

3. Test the installation as described in the **Preliminaries** section of this topic to ensure that it is recognized by the system.

Do the following to associate the .pl file extension with the Perl executable:

1. On the Windows desktop, click **Start** and then choose **Run**.

2. In the **Run** dialog box, type explorer and then click **OK**.
   The Windows Explorer appears.

3. Choose **Tools→Folder Options**.

4. In the **Folder Options** dialog box, click the **File Types** tab.

5. On the **File Types** page, click **New**.

6. In the **Create New Extension** dialog box, for **File Extension**, type PL.

7. Click **Advanced**.

8. From the **Associated File Type** list, select **Text Document**, and then click **OK**.

9. In the **Folder Options** dialog box, in the **Registered file types** section, select PL.

10. Click **Advanced**.

11. In the **Edit File Type** dialog box, type **Perl Script** in the file type box.

12. In the **Actions** section, select open, and then click **Edit**.

13. Click **Browse**, select the Perl executable, and then click **Open**.
14. In the Application used to perform action box, add the following arguments to the end of the path:
   
   `-s -x "%0" %*

   **Example**

   C:\oracle\product\10.2.0.4\perl\5.8.3\bin\MSWin32-x86-multi-thread\perl.exe s -x "%0" %*

15. Ensure the **Use DDE** check box is not selected.

16. In the **Edit File Type** dialog box, if additional entries are listed in the **Actions** section (such as **print** and **printto**), select them and then click **Remove**.

17. Click **OK**.

18. In the **Folder Options** dialog box, click **Close**.

**Perl is installed twice**

If two versions of Perl are installed on a system, including a version installed for use with Oracle software, you must resolve the different versions. Oracle sets the PERL5LIB environment variable to point to its Perl libraries, which may be incompatible with the ones installed with ClearanceDB.

There are two potential workarounds to this problem:

- Do the following to disable the PERL5LIB environment variable and resolve the conflict:
  1. Right-click **My Computer** and select **Properties**.
  2. In the **System Properties** dialog box, click the **Advanced** tab.
  3. On the **Advanced** page, click **Environment Variables**.
  4. On the **Environment Variables** page, in the **System variables** section, find PERL5LIB and change its name to **PERL5LIB_ORA**.
  5. Click **OK**.

   This workaround enables you to change the name of the environment variable back to PERL5LIB in the event you experience difficulties with the Oracle installation. This solution works well if you use the Oracle version of Perl only to patch the Oracle software (using the opatch program).

- Do the following to disable the version of Perl that was not installed with Oracle:
  1. Alter the order of the system PATH environment variable:
     a. Right-click **My Computer** and select **Properties**.
     b. In the **System Properties** dialog box, click the **Advanced** tab.
     c. On the **Advanced** page, click **Environment Variables**.
d. On the Environment Variables page, in the System variables section, find PATH and change the order of its entries so that the path to the Oracle Perl executable comes before the other version.

e. Click OK.

2. Create or change the file association for .pl files, as described in the Perl is installed with Oracle software section of this topic, to use the Oracle version of Perl.

This workaround is better if you prefer to use the Oracle version of Perl.
Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2018 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Geolus, GO PLM, I-deas, Insight, JT, NX, Parasolid, Solid Edge, Teamcenter, Tecnomatix and Velocity Series are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other trademarks, registered trademarks or service marks belong to their respective holders.