TEAMCENTER

ClearanceDB — Administration

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

What is Clearance?

What is Clearance? 1-1
What is ClearanceDB? 1-6

System requirements

System components 2-1
System requirements 2-4
Supported platforms 2-5
Version compatibility 2-7
Supported locales 2-7

Installing and configuring ClearanceDB

ClearanceDB installation and configuration overview 3-1
Typical ClearanceDB environments 3-1
Installing and configuring — quick start 3-4
- Quick start steps 3-4
- Install the ClearanceDB components (condensed version) 3-4
- Configure the ClearanceDB Server (condensed version) 3-5
- Configure the ClearanceDB Proxy (condensed version) 3-5
- Configure the ClearanceDB Client and Clearance Calculator (condensed version) 3-7
- Create and manage the ClearanceDB products (condensed version) 3-8

Installing and configuring — detailed instructions 3-9
- Installing the ClearanceDB components (expanded version) 3-9
- Configuring the Server (expanded version) 3-13
- Configuring the Proxy (expanded version) 3-31
- Configuring the Client and Clearance Calculator (expanded version) 3-52

Performing the analysis

Performing analysis overview 4-1
Setting up products for analysis 4-2
- Perl and SQL script reference 4-2
- Setting up products for analysis 4-4
- Configuring the ClearanceDB default product 4-5
- Create a new ClearanceDB product 4-6
- Using the product-specific configuration file 4-9

Performing analysis 4-11
- Accuracy of clearance calculations 4-11
- Clearance elements and CAD components 4-12
- Configure the Clearance Calculator 4-17
- Run the Clearance Calculator 4-19
- Understanding the results file format 4-20

Performing clearance batch analysis 4-21

Introduction to performing clearance batch analysis 4-21
Creating a Clearance Work Area 4-21
Create a clearance product 4-22
Configure Clearance products for analysis 4-22
Analyze clearance products from the Clearance Work Area 4-31
Set up end item batch calculation for managed products 4-32
Perform clearance analysis using Clearance.exe 4-40
Perl scripts for batch clearance analysis 4-40

Updating products in the database

Updating products in the database 5-1
Configuring ClearanceDB for product updates 5-1
Upload product data to the database 5-3
Upload data using the ClearanceDbClient 5-4
Merging results with the database 5-6

Using Mockup to view and manage Clearance results

Viewing and managing results overview 6-1
Enable ClearanceDB within the viewer 6-2
Load ClearanceDB results in the viewer 6-3
Display clearance issue status 6-3
Update issue status database from the viewer 6-4

Attributes

Overview of ClearanceDB product attributes 7-1
ClearanceDB product attributes reference 7-1
EAICL_RELEVANT_PART_ATTRIBUTE 7-3
EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE 7-4
EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT 7-4
EAICL_INITIAL_CLEARANCE_OWNER_CHOICE 7-5
EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION 7-5
EAICL_INITIAL_CLEARANCE_OWNER_SEARCH_METHOD 7-5
EAICL_INITIAL_CLEARANCE_PRIORITY 7-5
EAICL_INITIAL_CLEARANCE_STATUS 7-5
EAICL_INITIAL_CLEARANCE_COMMENT 7-6
EAICL_INITIAL_CLEARANCE_IMPORTANT 7-6
EAICL_INITIAL_CLEARANCE_CHANGE_REQUIRED 7-6
EAICL_INITIAL_CLEARANCE_CHANGE_REFERENCE 7-6
EAICL_CLEARANCE_CHANGE_DETECTION 7-6
EAICL_CLEARANCE_DISPOSITION_PERSISTENCE 7-8
EAICL_PART_MATCHING_METHOD 7-8
EAICL_PART_NAME_PROPER 7-9
EAICL_RULES_PART_ATTRIBUTE 7-9
EAICL_OCCURRENCE_KEY 7-9
EAICL_CHANGE_HISTORY_COLUMN 7-10

EAICL_PRESERVE_RESOLVED_ISSUES 7-11
EAICL_PRODUCT_HISTORY_MAX_ENTRIES 7-12
EAICL_PRODUCT_HISTORY_RETENTION_DAYS 7-12
EAICL_PRODUCT_STRUCTURE_CHANGE_TEST 7-12
EAICL_PRODUCT_STRUCTURE_CHANGE_REJECTION_PERCENT 7-12
EAICL_RULEBASE_CHANGE_TEST 7-12
EAICL_RULEBASE_CHANGE_REJECTION_PERCENT 7-13
EAICL_TMP_OBJECT_CLEANUP_AGE_IN_DAYS 7-13
EAICL_IDENTICAL_MATCHING_ATTRIBUTE 7-13
EAICL_SIMILAR_MATCHING_ATTRIBUTE 7-14
EAICL_RESEMBLING_MATCHING_ATTRIBUTE 7-15
EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION 7-16
EAICL_RULE_CATEGORY 7-16
Upload product attributes to the database 7-16
Example configuration file 7-17

Issue inheritance

Inheritance overview 8-1
Configuring the ALL product for issue inheritance 8-2
Set up the ALL product for issue inheritance 8-2
Define groups for the ALL product 8-2
Example configuration file for the ALL product 8-3
Deploying the issue inheritance mechanism 8-3
Enable the issue inheritance mechanism 8-3
Disable the issue inheritance mechanism 8-4
Modifying the issue inheritance mechanism 8-5

Log files

Understanding ClearanceDB log files 9-1
Understanding ClearanceDB advanced log files 9-4

Rules

What are clearance rules? 10-1
Using rules in clearance analysis 10-2
ClearanceDB rules and conditions details 10-5
Create ClearanceDB rule categories 10-7
Upload rules and conditions to the database 10-8

Images

What are clearance images? 11-1
Create images in batch using Mockup 11-2

Server-side filters

Server-side filters 12-1

Filter modes 12-1
Referencing part attributes 12-2
Create server-side filters 12-2

Zones

Defining clearance zones 13-1
Understanding clearance zones 13-1
Uploading zones to the database 13-2
Example zones file 13-2

Using ClearanceDB with Teamcenter

Using ClearanceDB with Teamcenter 14-1
Product and system requirements 14-2
Enable batch clearance calculation for ClearanceDB 14-2
Steps to analyze managed products 14-3
Performing adhoc ClearanceDB analysis 14-4
Define ClearanceDB roles 14-4
Configuring ClearanceDB to work with Teamcenter 14-4
Teamcenter global configuration options 14-4
Specify to use absolute occurrence IDs 14-13
Configure the Clearance.cfgproduct file 14-13
Encrypt the Teamcenter user password 14-14
Configuring ClearanceDB for multifield key data 14-15
Optionally configure Teamcenter export for 4GD worksets 14-16
Caching managed data for analysis 14-18
Set the default owner for each clearance result 14-23
Analyzing managed ClearanceDB products 14-24
Analyze managed ClearanceDB products 14-24
Analyze a managed product 14-24
Example output from managed analysis 14-26
View results in the Lifecycle Viewer and Structure Manager 14-30
Working with Design Context 14-31
Work with Design Context 14-31
Design Context clearance analysis modes 14-32
Specify connection information for Design Context 14-34
Specify to create an SCO for clearance results 14-35
Display additional clearance results information 14-36
Load ClearanceDB results in Design Context 14-36
Perform real-time analysis 14-37
Open clearance results in an SCO 14-37
Working with variants 14-38
Understanding variant-based clearance analysis 14-38
Variant-based clearance analysis methods 14-41
Variant-based clearance analysis (client-side) 14-42
Variant-based clearance analysis (server-side) 14-44
Using ClearanceDB with Active Workspace 14-46
Using ClearanceDB with Active Workspace 14-46

Clone stable IDs 14-47
Run the clone stable IDs utility 14-48

Maintaining the database

Database maintenance overview 15-1
Understanding the Simple Security Model 15-1
Automatic database maintenance 15-3
Execute stored procedures 15-4
Enforcing the integrity of input data 15-5
Available status reports 15-6

Troubleshooting

Loading results in the viewer 16-1
Frozen Clearance Results window 16-1
Clearance results disappear after user deletes a filter 16-1
Viewer licensing in Teamcenter 16-2
Viewer preferences and settings 16-3
Perl compatibility and configuration 16-3
Loading clearance results after an Oracle upgrade 16-4

Other resources 17-1

1. What is Clearance?

What is Clearance?

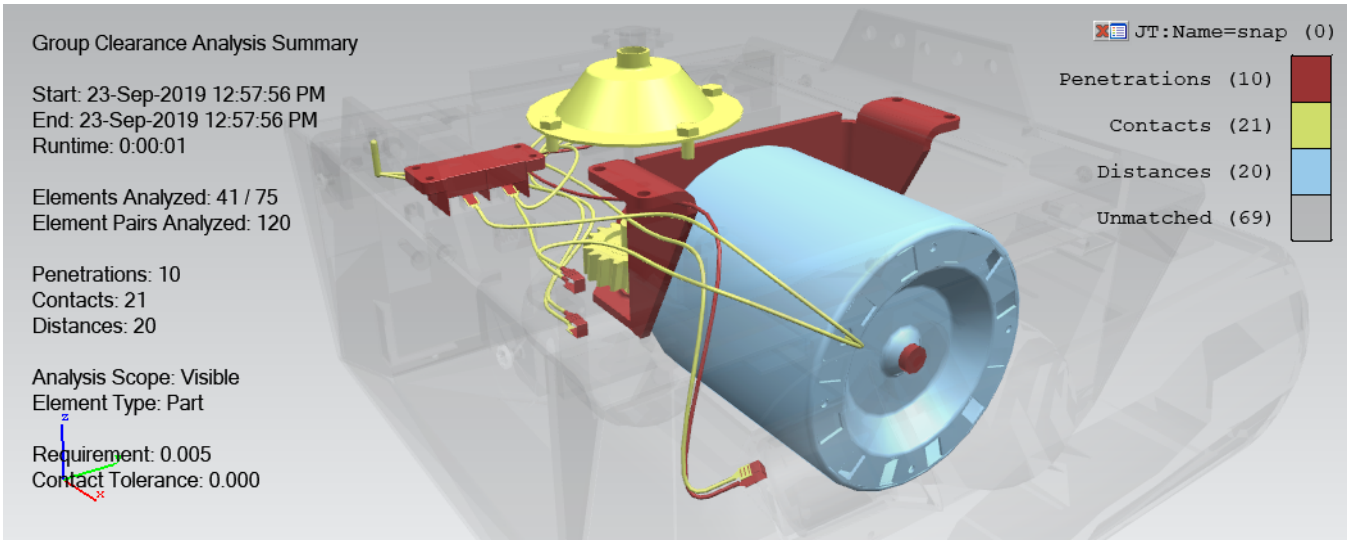
Clearance analysis

Clearance analysis identifies points of contact and/or penetration between elements or groups in your 3D model. Elements may be parts, leaf components, or end items.

The clearance requirement is the minimum allowed distance between any two elements within the analysis scope. Clearance requirements are measured in the same units as your 3D model. Elements that are in contact have a clearance requirement of 0.

Clearance analysis involves:

- Specifying the clearance settings that affect the type, precision, and nature of the analysis.
- Performing the clearance analysis on your model.
- Analyzing clearance violations (results) in the Viewing window.
- Disposition clearance issues by assigning a priority, status, and owner to element pairs on the **Results** list.



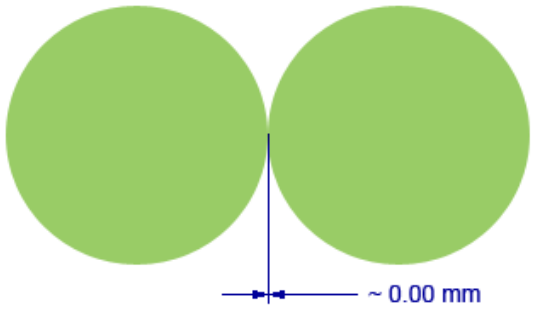
Clearance results types

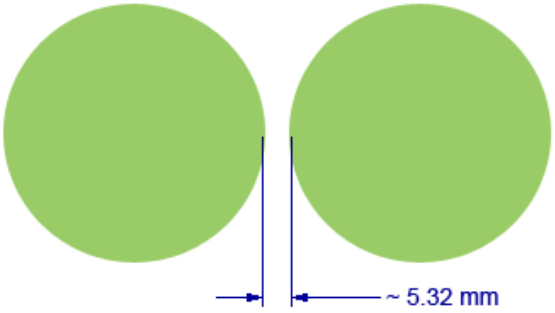
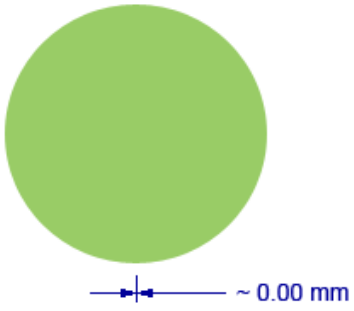
When you perform a clearance check, any two parts or elements that are closer to one another than a specified distance are identified as an issue for further investigation. The **Results** list displays the

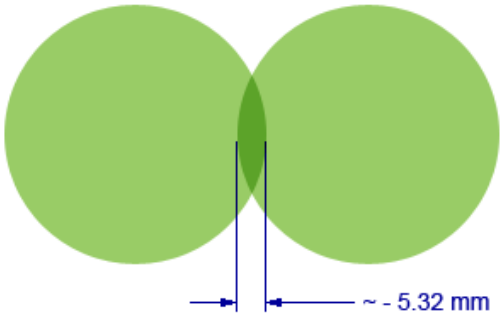
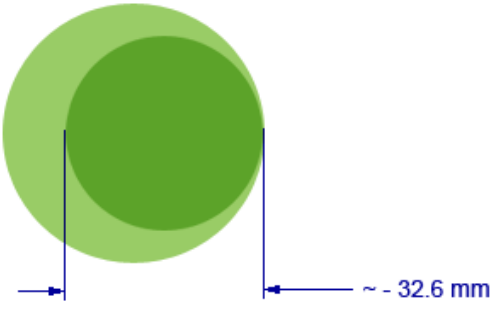
clearance issues identified by the analysis, along with the distance value between element pairs and other relevant information such as the interference type.

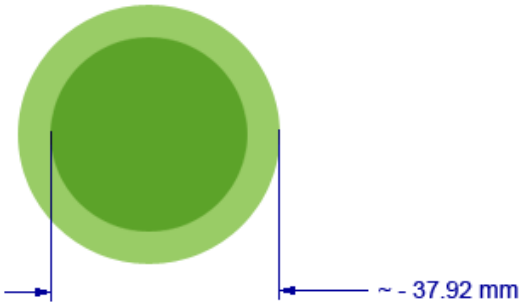
Clearance interferences can be *soft* or *hard*. Soft interferences are distance violations. They occur when the distance between two elements is equal to or closer than the specified minimum distance. Elements involved in a soft interference do not touch or penetrate one another. Hard interferences are contact or penetration violations. They occur when two elements touch or occupy the same geometric space.

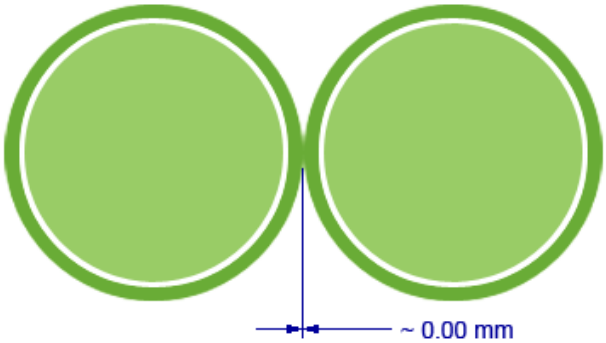
The table below describes each clearance type and provides examples to help you understand the criteria used to identify them.

This type	Appears in the results list as	Indicating this
Contact	**C**	A hard interference. The elements are in contact but do not penetrate. The distance result is always ~ 0.0. Note: Clearance results are approximate if your elements do not consist of precise geometry. 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.  **Contact**
Distance	**D**	A soft interference. The elements are closer than the specified minimum clearance requirement, but do not come into contact with or penetrate one another. The distance result is always a value greater than ~ 0.0.

This type	Appears in the results list as	Indicating this
		 **Distance**
Duplicate	**Dup**	The clearance calculation has determined that the elements are duplicates based on the bounding box, surface area, volume, and center of gravity. If these four attributes are the same then the parts are considered duplicates.  **Duplicate**
Failure	**F**	The clearance calculation has failed to complete. This clearance type can appear only in results generated in Batch Mode, most likely with ClearanceDB analysis. This may happen if the Clearance Calculator crashes.
Penetration	**P**	A hard interference. One element penetrates or is completely contained by another. The distance result is always a value less than ~ 0.0. Note: Penetration values are always approximate, even when using precise geometry. Note: Penetrations are not detected automatically unless the **Distinguish between Contact and Penetrations** check

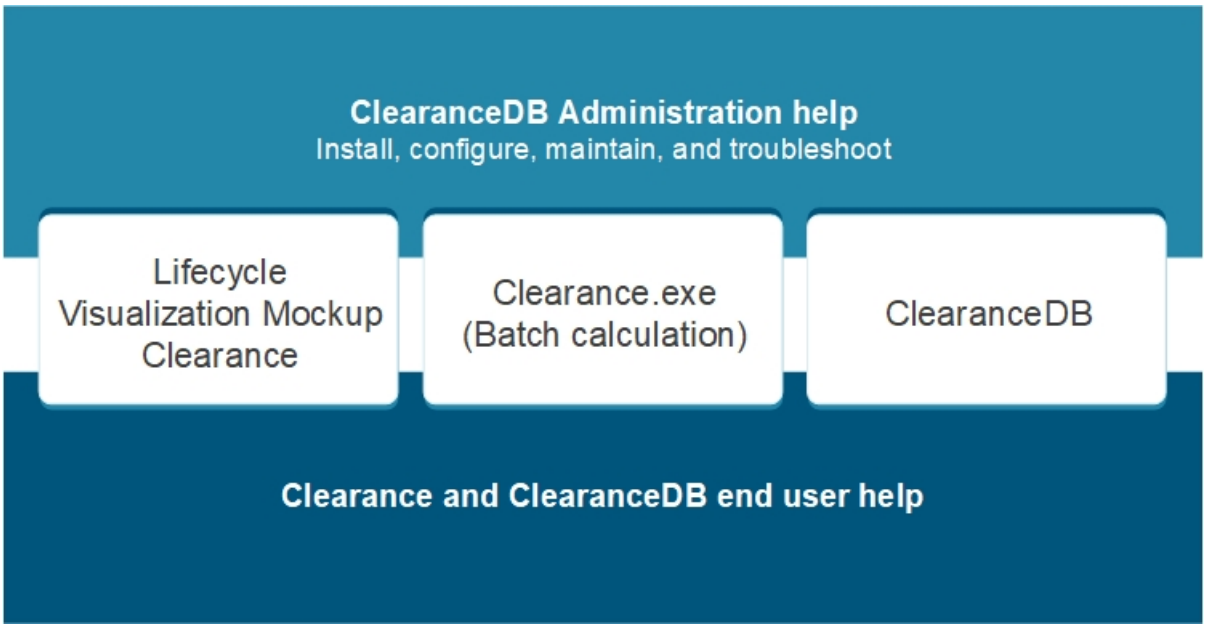
This type	Appears in the results list as	Indicating this
		box is selected in the calculator settings. If this option is not turned on, penetrations are categorized as the **Unknown** clearance type, and you must perform further analysis to identify the issue as a penetration.  **Partial penetration** 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 ~ 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. 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.  **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

This type	Appears in the results list as	Indicating this
		penetration depth is the minimum distance plus the distance the minimum distance vector extends to the further side of the inner elements' bounding box.  **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. • An inner element with a material thickness value that causes the element to extend beyond the outer element is considered completely contained and identified as a penetration.
Thickness	**T**	A violation where material thickness is a factor of the calculation. Further analysis 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. 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

This type	Appears in the results list as	Indicating this
		issues that contain element pairs with material thickness values.  **Thickness**
Unknown	**U**	A hard interference. Further analysis is required to classify as a contact or penetration.

What is ClearanceDB?

Teamcenter Clearance Database (ClearanceDB) enables customers to detect, manage, and resolve spatial interference issues as they arise during a product's lifecycle. By automating this process and constantly re-evaluating on a regular basis, customers can identify design flaws that may have otherwise delayed the release of new product offerings, increased costs, or jeopardized quality.



Capabilities

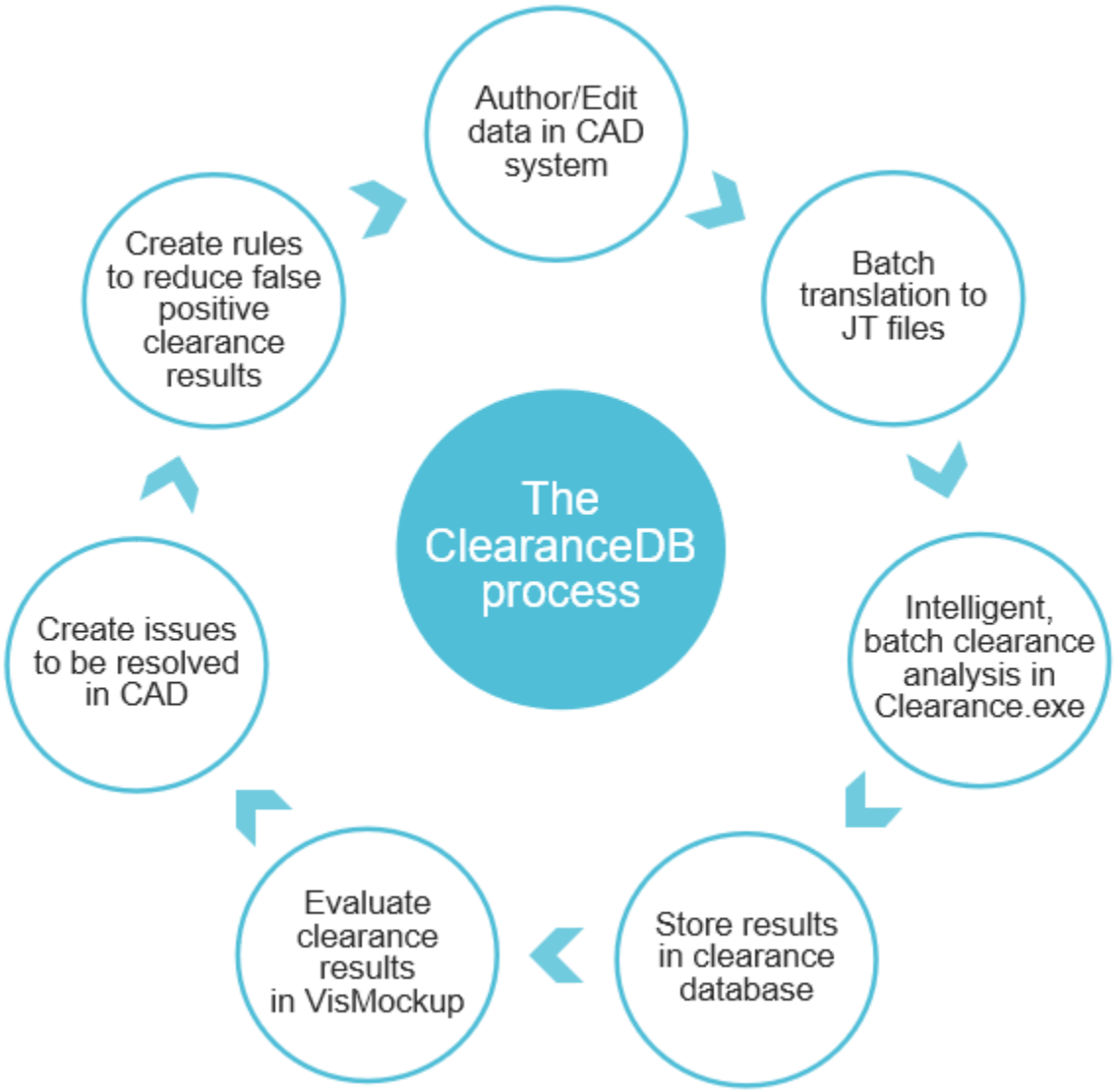
The Clearance db server option provides the following capabilities:

- Completely automated, intelligent batch clearance analysis that is continuously updated.
- Analysis results are stored in the Clearance database.
- The Clearance database can be accessed by users at any time to understand the number and severity of clearance issues for any product, or to investigate specific issues in Teamcenter lifecycle visualization mockup.
- Customers can analyze and manage clearance issues across multiple product configurations, derivatives, and options by integrating the clearance analysis process into a Teamcenter-managed product lifecycle.

Benefits

- Automates the checking and reporting process to facilitate faster, repeatable, and more accurate clearance evaluations.
- Reduces development cost by enabling engineers to catch design flaws early in the lifecycle.
- Fosters innovation by enabling designers and engineers to focus on products instead of DMU (Digital Mockup) issues.
- Reduces design cost by enabling design engineers to digitally integrate the entire product.

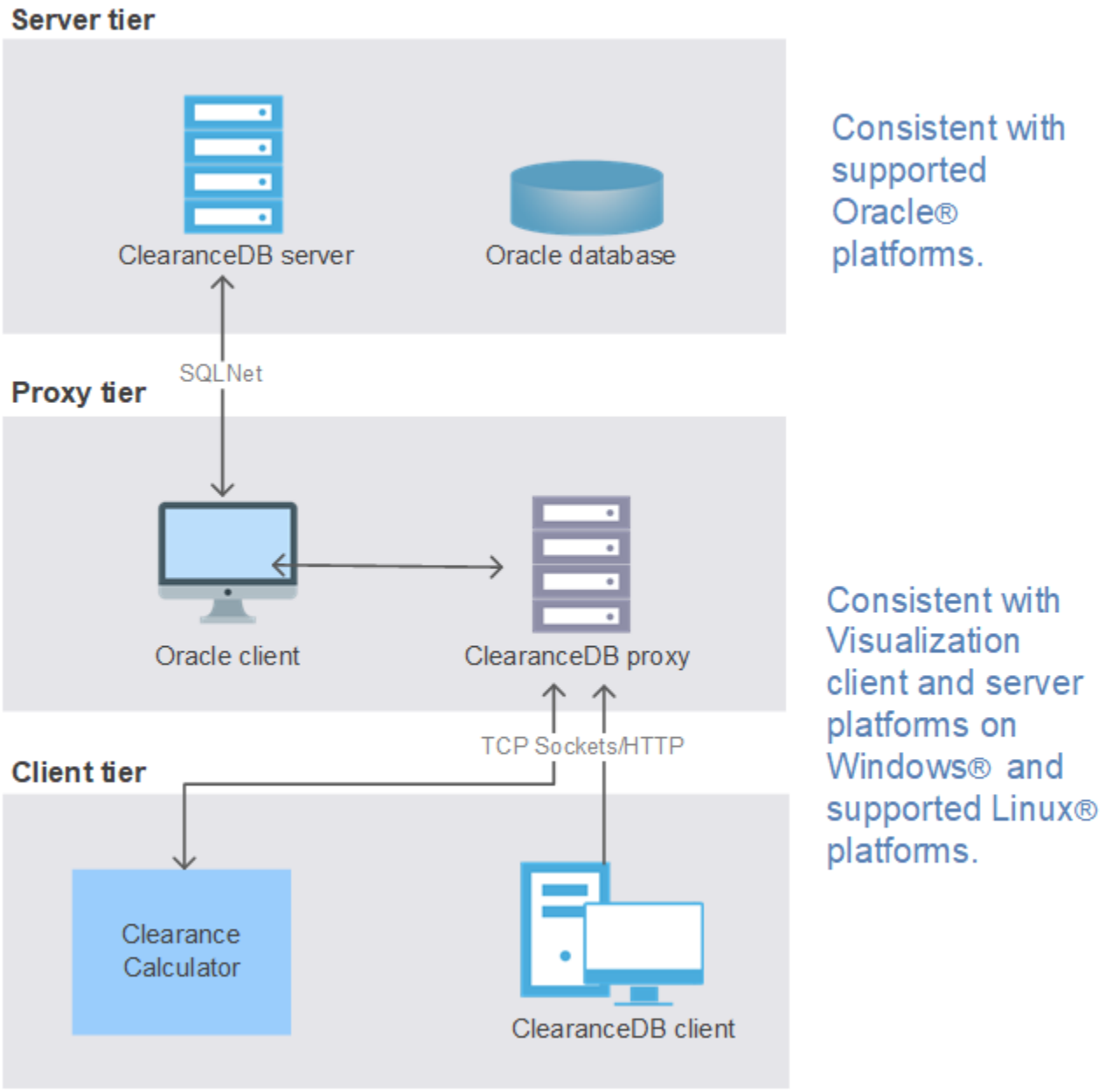
The ClearanceDB process



2. System requirements

System components

The basic ClearanceDB architecture along with descriptions of the role of each component in the ClearanceDB process is as follows.



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 along with the client and proxy components must be from the same release. Additionally, the server component must be at least as recent as or higher than the client and proxy components.

  The following scenarios can help understand the support criteria.

Server release	Client release	Proxy release	Whether supported
8.1	8.0	8.0	Yes
8.0	8.1	8.1	No
8.0	8.0	8.1	No

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 uploaded to 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 Design Context 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 three applications, the ClearanceDbProxyServer, ClearanceDBProxyHttpServer, and the ClearanceDbProxyClient. Only the ClearanceDbProxyServer and ClearanceDBProxyHttpServer (optional) need 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.

  ClearanceDBProxyHttpServer.exe is a web service executable added to the proxy tier to support HTTP/s-based communication between clients and Clearance database. This communication protocol is optional and can be configured via settings defined in the configuration file.

- 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.

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 Enterprise and Standard Editions.

The ClearanceDB Server requires Oracle Net Services, sqlplus, and sqlldr, to run the ClearanceDB SQL scripts.

For information about system hardware and software requirements, choose from links found in the *Hardware and Software Certifications* knowledge base article on Support Center (scroll down to the *Teamcenter Visualization Software Certifications* heading).

For supported platform requirements, see **Supported platforms**.

ClearanceDB Http Server requirements

- SSO (single sign-on) setup on the client machines. For information on setting up the SSO environment, see Security Services Installation/Customization in *Security Services Configuration*.

- Teamcenter Web Applications Manager with the Identity and Login providers on the Server. For information on setting up the Identity, and Login providers, see Security Services Installation/ Customization in *Security Services Configuration*.

  Teamcenter Server with SSO support is required when using this feature.

- Apache Web Server, Tomcat, or WebLogic in conjunction with ApacheDS (Directory Services) or equivalent LDAP for authentication.

- Ideally, *ClearanceDbProxyHttpServer.exe* should be configured to run behind an Apache reverse proxy for security reasons. It is also possible to run *ClearanceDbProxyHttpServer.exe* as a secure server.

ClearanceDB Proxy requirements

The ClearanceDB proxy tier requires the Oracle Instant Client Basic (not Basic Lite) package.

For information about system hardware and software requirements, choose from links found in the *Hardware and Software Certifications* knowledge base article on Support Center (scroll down to the *Teamcenter Visualization Software Certifications* heading).

For supported platform requirements, see **Supported platforms**.

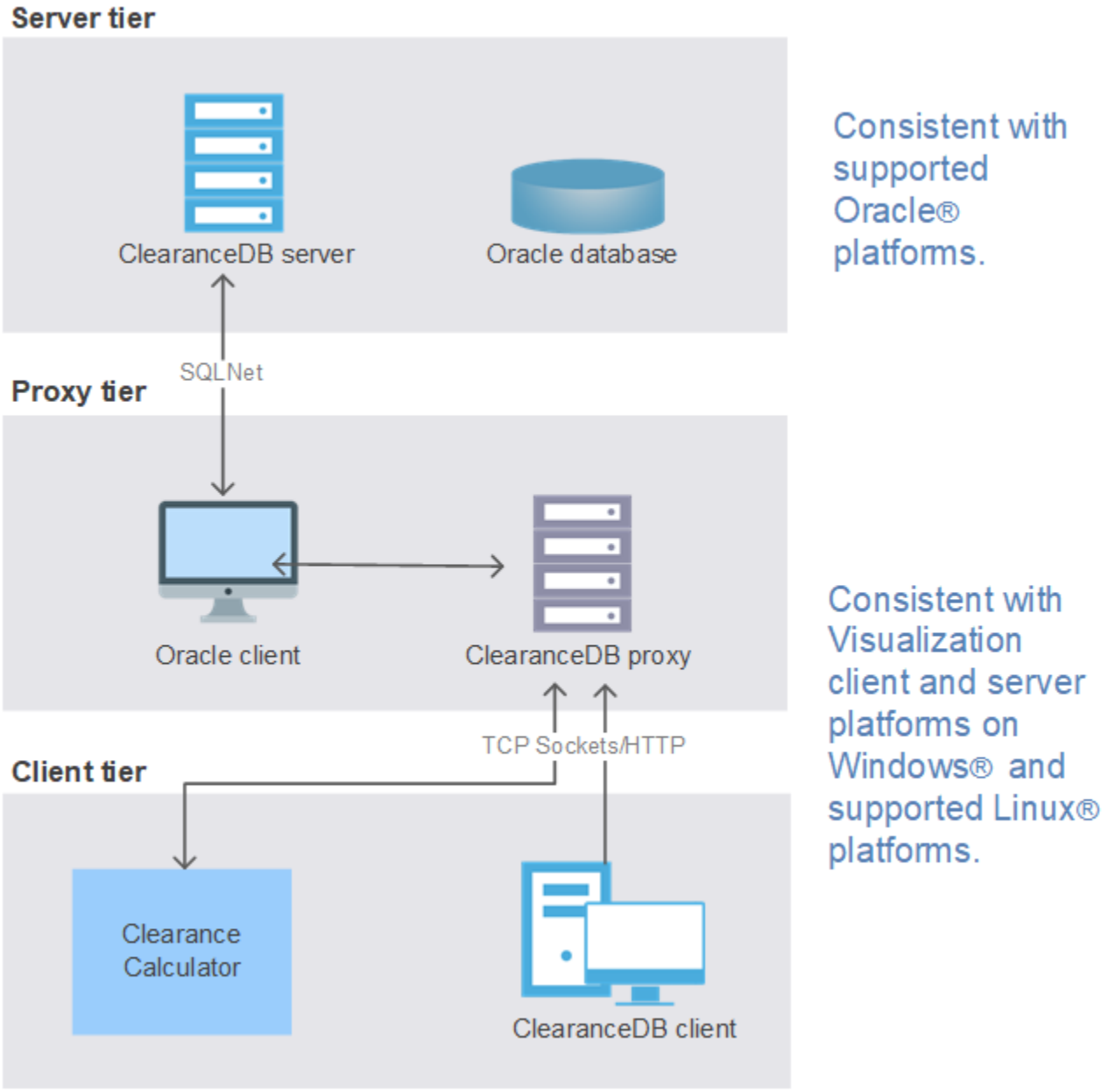
Clearance Calculator Client requirements

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 512 GB of RAM and 32 cores (or more).

Supported platforms

For more information about supported virtual machines, please contact your Siemens Digital Industries Software representative.

For information about the platforms on which you can run Teamcenter lifecycle visualization, download the appropriate platform matrix spreadsheet from **Support Center**.



1. On the **lifecycle visualization page**, click the **Support White Papers x.x** tile.

2. Download a platform matrix spreadsheet from the list.

3. (Optional) To see supported platforms for other lifecycle visualization versions, click **More Versions** 

and then select a version under the **Support White Papers** heading.

For information about Siemens Digital Industries Software that supports Teamcenter lifecycle visualization 2412, download the **Teamcenter Interoperability Matrix** spreadsheet found on the Support White Papers Certifications page on **Support Center**.

Version compatibility

Version compatibility with Teamcenter

ClearanceDB support for Teamcenter versions is consistent with Lifecycle Visualization support. Download the **Teamcenter Interoperability Matrix** spreadsheet found on the Support White Papers Certifications page on **Support Center**.

Version compatibility within ClearanceDB tiers

- **Client** (Mockup, *Clearance.exe*, *ClearanceDbClient.exe*) and **Proxy** (*ClearanceDbProxyServer.exe*, *ClearanceDbProxyHttpServer.exe*, and *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 release, for example, 10.x. In some cases, a server that is a higher 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 Digital Industries Software representative.

Supported locales

In general, ClearanceDB results can be loaded and managed using localized Teamcenter and Teamcenter lifecycle visualization in any single-byte locale. However, the ClearanceDB application is only *partially* localized, as outlined below.

> Tip:
>
> A single byte (8 bits) is sufficient to represent an alphabet with 256 characters, which is sufficient for languages such as English, French, Italian, German, and Spanish. But languages such as Chinese or Japanese can have thousands of characters, which is too many to be represented by a single byte, so those languages are referred to as *multibyte*.

ClearanceDB Server (Clearance database)

- Strings originating from the Clearance database (including ClearanceDB reports, column names, Oracle errors, and ClearanceDB server errors) are not localized. They will always appear in English.

- Any user data entered into the Clearance database, such as product names, issue dispositions, and rules and conditions, must be single-byte.

Teamcenter

Any data coming into ClearanceDB from Teamcenter via BOMwriter PLM XML export (including product names, part names, and metadata) must be single-byte, since ClearanceDB can only consume single-byte data.

Clearance.exe

Strings originating from the Clearance.exe (informational and error messages) are fully localized.

ClearanceDB Proxy

Strings originating from the ClearanceDB Proxy (informational and error messages) are fully localized.

ClearanceDB Working Directory

All files under the ClearanceDB working directory, such as .cfg files, .csvcldb files, and product names, must be single-byte.

ClearanceDB Perl Scripts

Strings originating from the ClearanceDB Perl scripts, (informational and error messages) are not localized. They will always appear in English.

3. Installing and configuring ClearanceDB

ClearanceDB installation and configuration overview

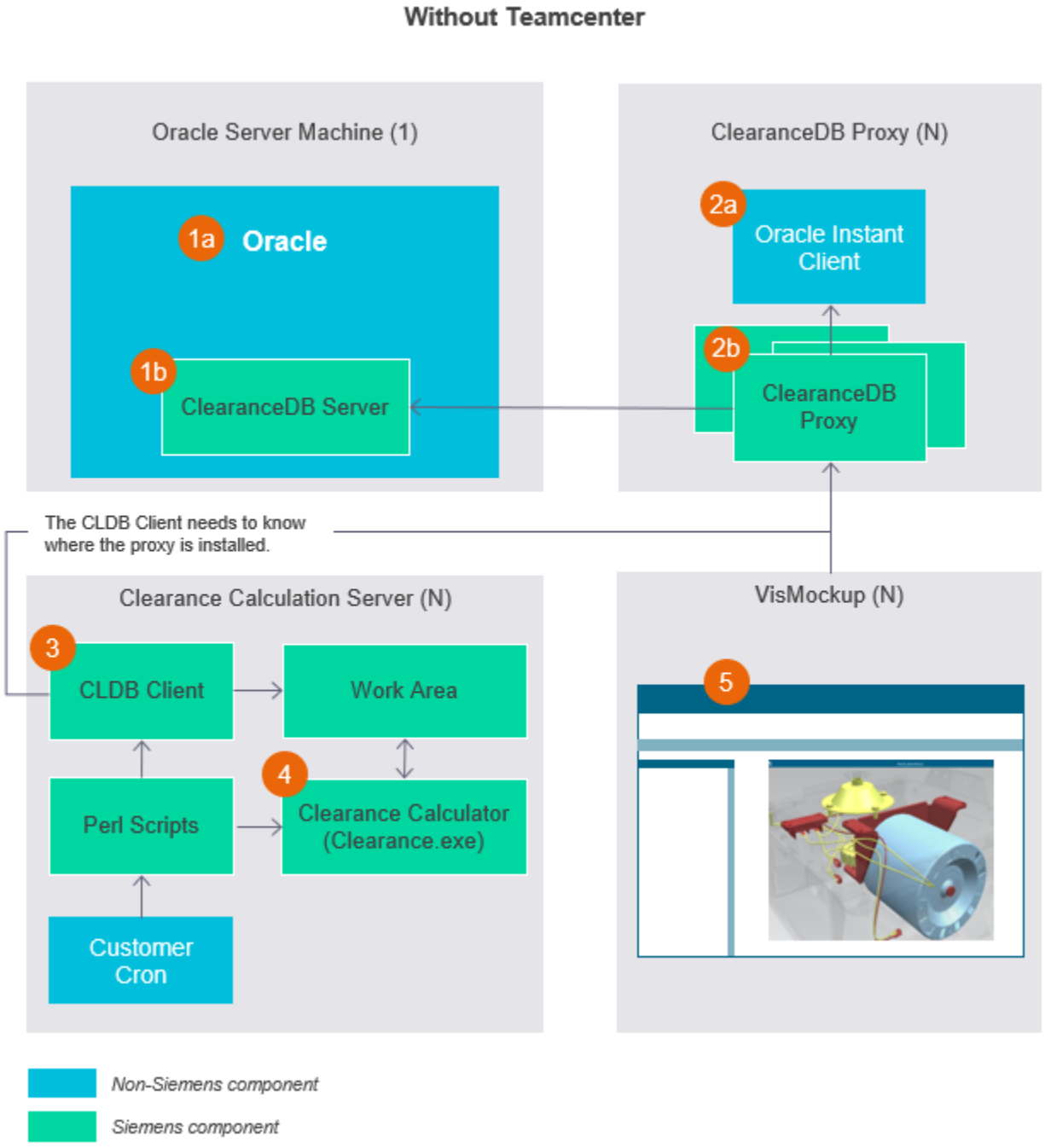
Follow the **steps** in the *Installing and configuring — quick start* section to set up a ClearanceDB environment from scratch. The quick start topics contain links to sections within *Installing and configuring — detailed instructions* that provide in-depth coverage of each step. For best results, follow the exact order of the quick start steps when setting up your ClearanceDB environment.

The *Next steps* portion of the quick start process provides links you to locations within the *ClearanceDB — Administration* deliverable that detail tasks you can complete once the ClearanceDB environment is set up. For example, creating auxiliary users with roles (permissions), and using Mockup, Active Workspace, and Teamcenter with ClearanceDB.

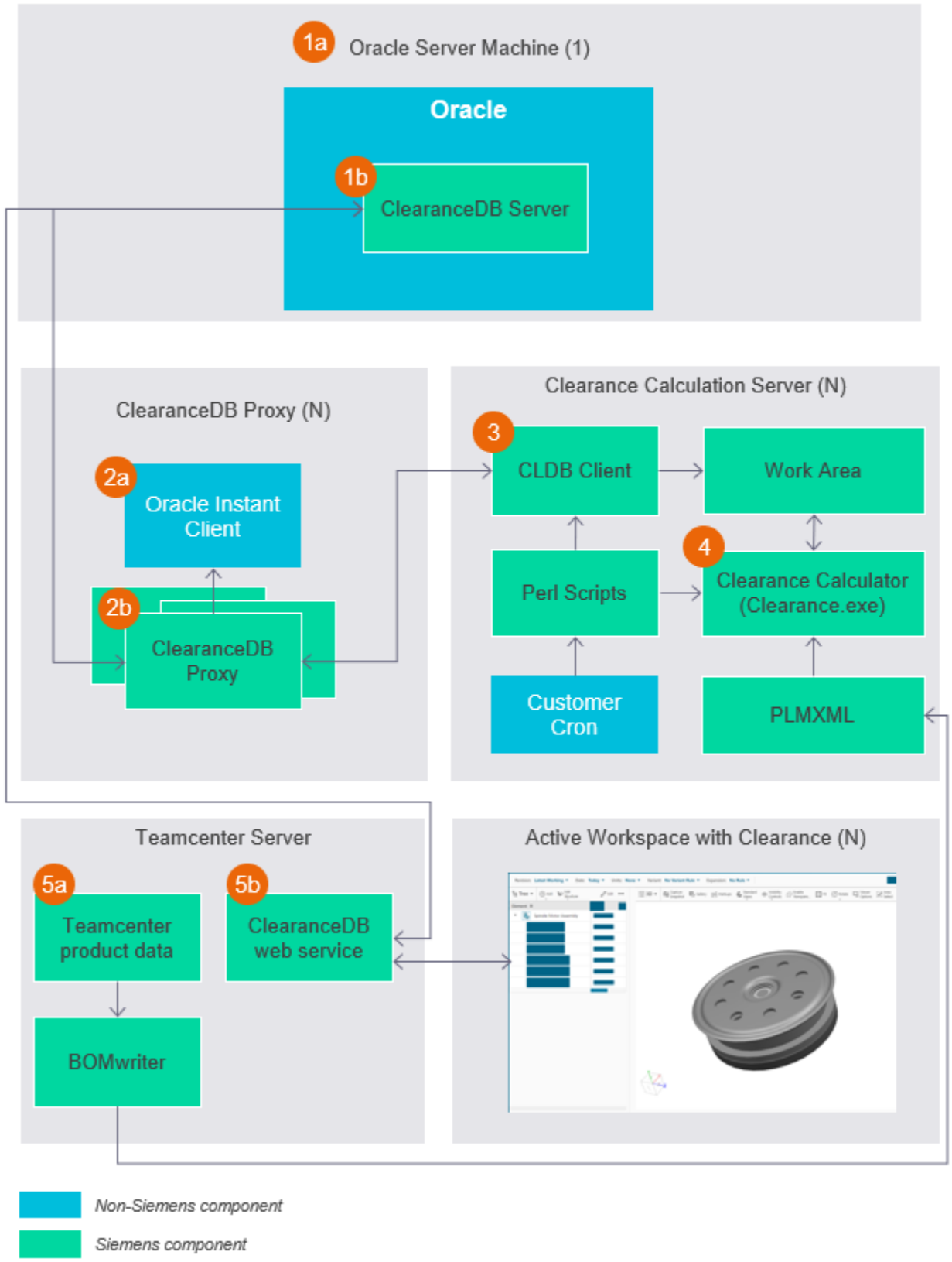
Typical ClearanceDB environments

The images below provide examples of typical ClearanceDB environments. The following apply:

- Most often, the Server, Proxy, Client/Calculator, and Mockup are located on different machines.
- The ClearanceDB Server is usually installed and configured by an Oracle specialist.
- Generally, there is a single Calculation Server, which runs on a dedicated machine; however, you can configure multiple Calculation Servers, if desired.
- There can also be as many Proxy Servers as desired.



With Teamcenter



Installing and configuring — quick start

Quick start steps

To set up a ClearanceDB environment, complete the following steps.

1. Installing the ClearanceDB components

1. Install the ClearanceDB components.
2. Configure the ClearanceDB Server.
3. Configure the ClearanceDB Proxy.
4. Configure the ClearanceDB Client and Clearance calculator.

2. Creating and managing ClearanceDB products

- Create and manage the ClearanceDB default product.

3. Completing the next steps

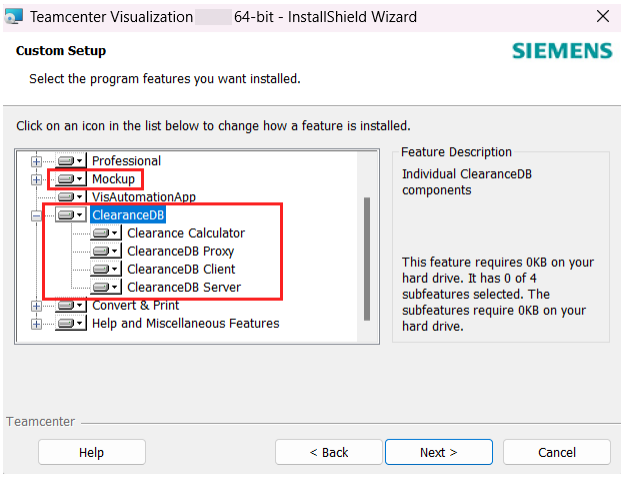
- Use roles to grant or restrict access.
- Create auxiliary users.
- Use Mockup to view and manage Clearance results.
- Use ClearanceDB with Teamcenter.
- Use ClearanceDB with Active Workspace.

Install the ClearanceDB components (condensed version)

Use the to install any combination of the following ClearanceDB components:

Components can be installed on a single machine or on separate machines by running the installer on multiple machines.

- ClearanceDB Server
- ClearanceDB Proxy
- ClearanceDB Client
- Clearance Calculator
- Teamcenter lifecycle visualization mockup



See the following topics for more information on installing the ClearanceDB components:

- *Install ClearanceDB components on Windows*
- *Install ClearanceDB components on Linux*

Configure the ClearanceDB Server (condensed version)

For more detailed information on the steps below, see **Configuring the Server (expanded version)**.

1. If you did not install the ClearanceDB Server when you **installed other ClearanceDB components**, use the to do so now.
2. Create the ClearanceDB database on your Oracle Server.
3. Use either Oracle Enterprise Manager Database Control or sqlplus to create the following two ClearanceDB tablespaces in your database: *CL_TABLE* and *CL_INDEX*.
4. Use sqlplus and the *cre_ClearanceDB_user.sql* script to create a ClearanceDB Oracle user.

   Configure this user to store product configurations, unique definitions of rules and conditions, zones, analysis results, and issue dispositions.
5. Use sqlplus and the *upd_ClearanceDB_user.sql* script to update the Oracle user created in the previous step.

Configure the ClearanceDB Proxy (condensed version)

For more detailed information on the steps below, see **Configuring the Proxy (expanded version)**.

1. If you did not install the ClearanceDB Proxy when you **installed other ClearanceDB components**, use the to do so now.

2. Install the Oracle Instant Client (Windows or Linux).

   Use the Oracle Instant Client to connect from the Proxy Server to the ClearanceDB on the Oracle Server.

3. Create the ClearanceDB Work Area.

4. Define the following values in the *global.dbc* file for your work area:

   - **DATASOURCE** — Proxy Server machine name and port number.
   - **CONNECT_DATA** — name identifying the connection between the Proxy Server and ClearanceDB.

5. Define the following default Proxy settings using one of the methods outlined in the table below.

   - **OracleClientDirectory** — full path to the Oracle Instant Client.
   - **Port** — port number of the ClearanceDB Proxy Server. Use the following format:

     <port number>,<number of threads listening to the port>

   - **CONNECT_DATA** — <connect_name>,<user>/<password>@<service_name>

To	Do this
Apply these default settings to all Proxies created after this step.	Define the default Proxy settings in the *Clearance.cfgglobal* file of your ClearanceDB_Work_Area.
Apply these default settings to only the Proxy defined in the next step of this procedure.	Define the default Proxy settings in the Proxy's *Clearance.cfgglobal* file after you define it in the next step. These settings are applied only to this Proxy.

6. Use the **create_proxy.pl** script to create a ClearanceDB Proxy.

   (Optional) Define the default Proxy settings in this Proxy's *Clearance.cfgglobal* file. Refer to the previous step for details.

7. Start the ClearanceDB Proxy using one of the following methods.

   - Use the **start_proxy.pl** script.
   - (Windows, only) Right-click the **Clearance.cfgproxy** file and select **Start Proxy**.

8. Ping the **Proxy** and **database** to verify the Proxy is working properly using one of the following methods.

To	Do this
Ping the Proxy.	• Use the *ping_proxy.pl* script. • (Windows, only) Right-click the **global.dbc** file of your ClearanceDB_Work_Area and select **Ping Proxy**.
Ping the ClearanceDB database.	• Use the *ping_database.pl* script. • (Windows, only) Right-click the **global.dbc** file of your ClearanceDB_Work_Area and select **Ping Database**

Configure the ClearanceDB Client and Clearance Calculator (condensed version)

For more detailed information on the steps below, see **Configuring the Client (expanded version)**.

1. If you did not install the ClearanceDB Client and Clearance Calculator when you **installed other ClearanceDB components**, use the to do so now.

   For the purposes of the Quick Start procedure, it is assumed that the ClearanceDB Client and Clearance Calculator are located on the same machine; however, you can install them on separate machines, if preferred. See *Performing Clearance batch analysis* for more information.

2. Create the ClearanceDB Work Area.

   If you are installing the ClearanceDB Client and ClearanceDB Proxy on the same machine, you can use the same ClearanceDB_Work_Area that you set up when you installed the **ClearanceDB Proxy**. If you are installing the ClearanceDB Client on a separate machine, create a ClearanceDB Work Area on the machine where you are installing the Client.

3. Define the following values in the *global.dbc* file for your work area:

   If you are using the same ClearanceDB_Work_Area that you set up when you installed the **ClearanceDB Proxy**, you can skip this step. If you are installing the ClearanceDB Client on a separate machine, define the *global.dbc* file values on the machine where you are installing the ClearanceDB Client.

   - **DATASOURCE** — Client Server machine name and port number.
   - **CONNECT_DATA** — name identifying the connection between the Client Server and ClearanceDB.

4. To be able to generate a ClearanceDB result and upload the file, do the following.

a. Open the *Clearance.cfgglobal* file in a text editor.

b. Navigate to *Section 4: Clearance Analysis configuration*.

c. Comment out this line:

```
RulesObject - Universal Clearance Requirement
```

d. Uncomment these lines:

```
RulesObject = Requirement Rules from ClearanceDB Server

ResultsObject = ClearanceDB Results
```

Create and manage the ClearanceDB products (condensed version)

You must create a ClearanceDB default product before creating new products in the Clearance database. The default product is a global configuration that you can customize to match your unique analysis requirement and automatically apply to new products that you create.

1. If you did not create a default product when you created the ClearanceDB_Work_Area, **create a default product**.

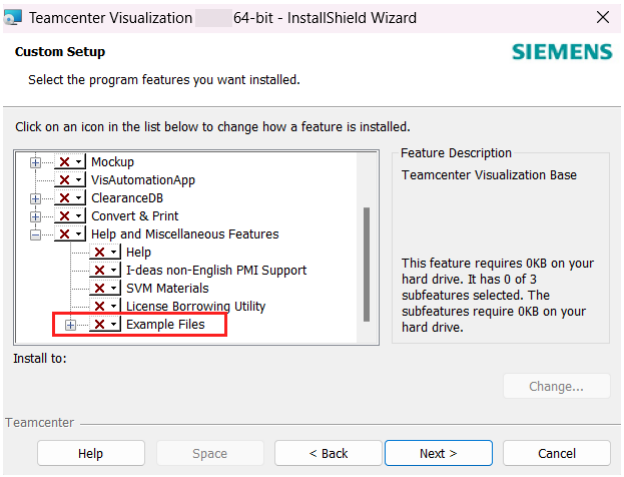
2. **Update the product to your database**.

3. Do one of the following to view all products in your database.

   - Run the *list_all_products.pl* script.
   - (Windows, only) Right-click the *global.dbc* file in the ClearanceDB_Work_Area and select **List All Products**.

4. Create, analyze, and update local products in your ClearanceDB.

If	Then do this
You have local data to use to create products.	**Create a new ClearanceDB product**.
You do not have local data but would like to do a test run you can use the Example data provided during installation.	a. Download the Example files through the Teamcenter Visualization - InstallShield Wizard found under **Help and Miscellaneous Features**.

If	Then do this
	 b. **Create a new ClearanceDB product**.
You would like to use data managed in Teamcenter.	Read through *Using ClearanceDB with Teamcenter* and complete the configuration tasks outline in this content before **creating a new ClearanceDB product**.

5. **Specify new ClearanceDB products** in the **Clearance.cfgproduct** file.

6. (Optional) Specify options for Clearance analysis by **configuring the Clearance Calculator**.

   If you are not concerned about Clearance analysis options you can analyze the products using the default options.

7. Analyze the products by **running the Clearance calculator**.

8. Use the *update_product.pl* script to **upload the** *ClearanceResultsDbUpload.csvcldb* **file**.

   The *ClearanceResultsDbUpload.csvcldb* file will display in the product directory once analysis is complete. Use the [-ud] flag to upload the file to the database.

9. Use the **list_product.pl** script to view the product data.

Installing and configuring — detailed instructions

Installing the ClearanceDB components (expanded version)

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:

Choose this option	Then do this
Install a node-locked license from this file	Enter the file path, or click **Browse** to locate the appropriate file and then click **Open**.
Obtain a license from a license server machine Note: This option is also used when the same	Do one of the following: • 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. • If you are setting up a redundant license server:

Choose this option	Then do this
machine acts as the License Server. It is for a 1 license SERVER type of license.	a. Enter the server name and port numbers in the **Server Name** box: `port@host1;port@host2;port@host3` b. Leave **Server Port** blank. 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). Note: For more on redundant license servers see the licensing guides, available in the Licensing directory.
Do not modify the existing license file(s)	No further action is necessary.

Note:

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

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 Linux

Note:

Teamcenter lifecycle visualization mockup is supported only on the Windows platform; however, the server components are supported on Linux and Windows.

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:

Choose this option	Then do this
Do not modify the existing license file(s)	No further action is necessary.
Obtain a license from a license server machine	Enter the Server Name and Server Port.

> Note:
>
> For more information on licensing Teamcenter lifecycle visualization, see Installation.

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.

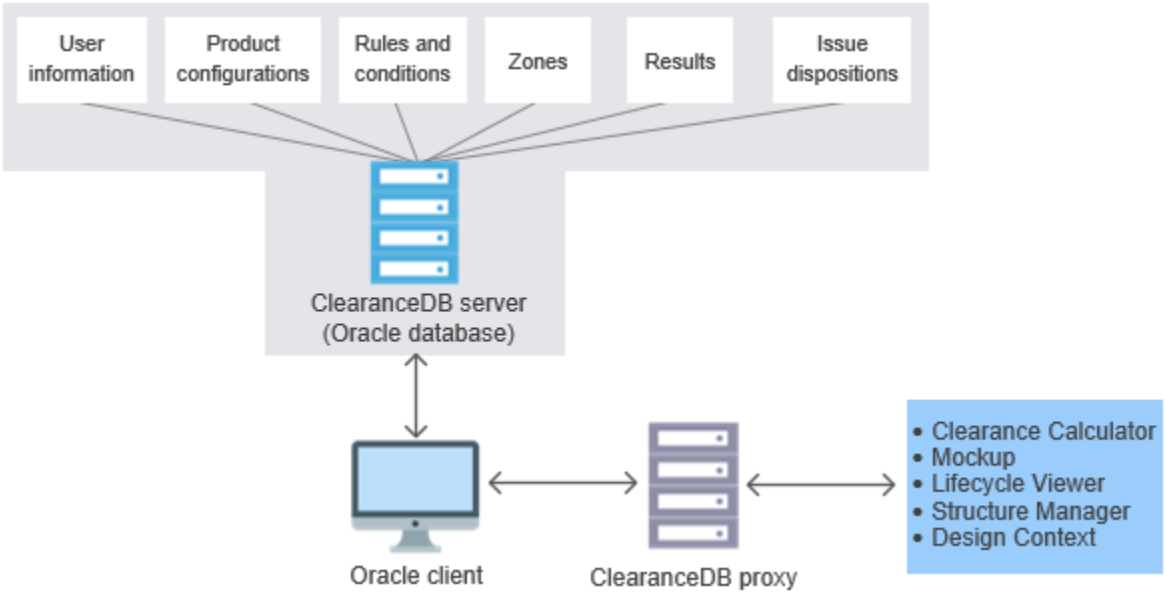
14. In the **Installation Complete** window, click **Finish**.

15. To close the installer, click **Exit**.

Configuring the Server (expanded version)

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**.
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.
10. On the Step 6 of 12: Storage Options page, accept the default, and then click **Next**.
11. On the Step 7 of 12: Database File Locations page, accept the default, and then click **Next**.
12. On the Step 8 of 12: Recovery Configuration page, accept the defaults, and then click **Next**.
13. On the Step 9 of 12: Database Content page, ensure **Sample Schemas** is not selected, and then click **Next**.
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**.

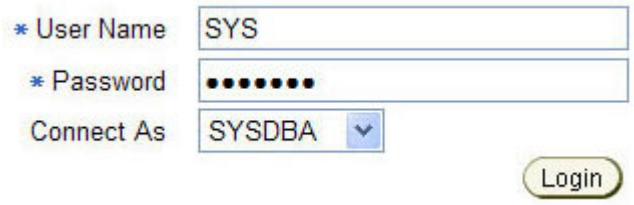
    Creation of the database begins. When the database is created, the **Database Configuration Assistant** dialog box appears.
18. Click **Exit**.

You are now ready to create the listener.

Tune the database memory parameters

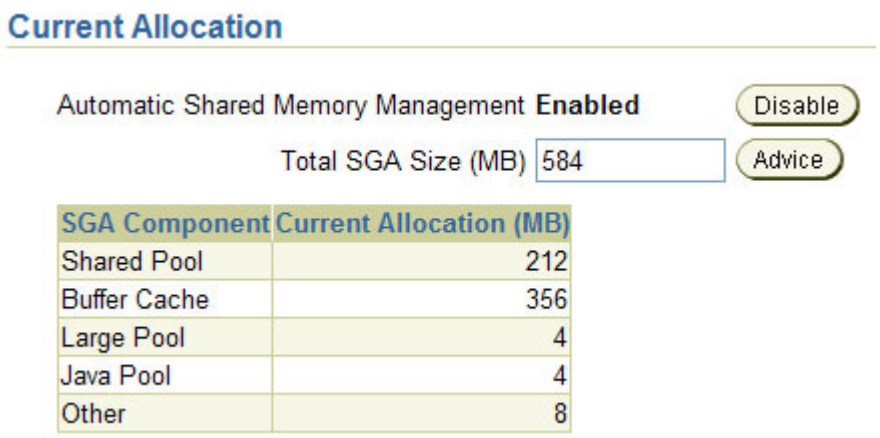
Modify the following settings for optimal database performance.

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



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

3. On the Memory Parameters page, in the **Current Allocation** section, click **Disable**.



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

SGA Component	Current Size (MB)	New Size (MB)
Shared Pool	212	100
Buffer Cache	356	600
Large Pool	4	4
Java Pool	4	4
Other	8	8
Total SGA	584	584

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

SGA Component	Current Size (MB)	New Size (MB)
Shared Pool	212	100
Buffer Cache	356	600
Large Pool	4	4
Java Pool	4	4
Other	8	8
Total SGA	584	584

6. Click **OK**.

   You are now ready to create the ClearanceDB database tablespaces.

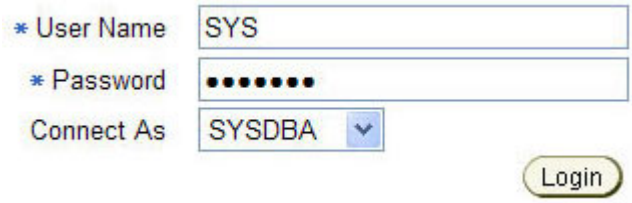
Create the ClearanceDB tablespaces

ClearanceDB requires the following tablespaces within your database instance:

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

Create the CL_TABLE tablespace

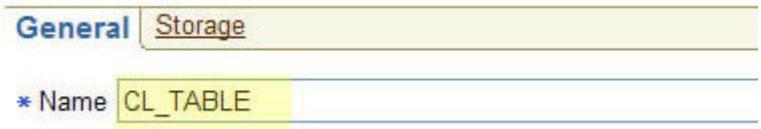
1. Using the Oracle Enterprise Manager **Database Control**, login as a user with SYSDBA privileges to your ClearanceDB database instance.



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

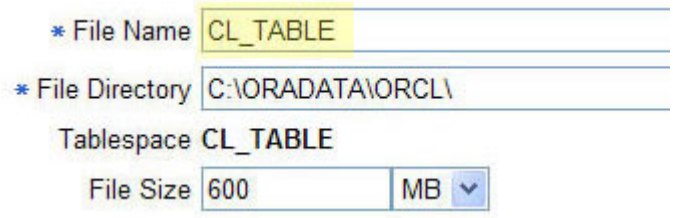
3. On the Tablespaces page, click **Create**.

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

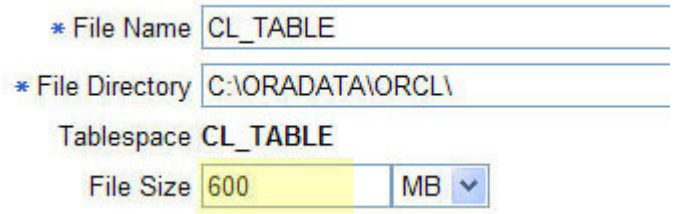


5. Click **Add**.

6. On the Add Datafile page, in the **File Name** box, type `CL_TABLE`.



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

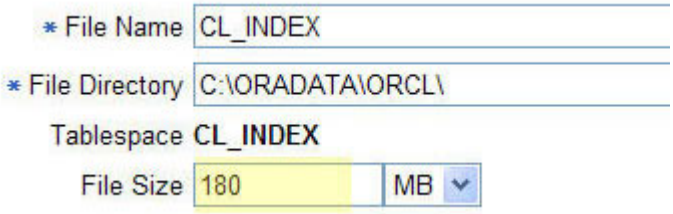
1. On the Tablespaces page, click **Create**.

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.

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 Teamcenter Rich Client Installation on Windows or Teamcenter Installation on Linux Using TEM.

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*
   - (Linux) *<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

For this parameter	Type
owner	The user name of the ClearanceDB Oracle user account.
password	The password for the Oracle user account.
connection	The Local Net Service Name for the ClearanceDB Oracle database.
logfile	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*.

For this parameter	Type
dflt_tablespace	**CL_TABLE**. This is the name of a **ClearanceDB tablespace**.
indx_tablespace	**CL_INDEX**. This is the name of a **ClearanceDB tablespace**.

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.

Upgrading a ClearanceDB from Oracle 12c to Oracle 19c

1. Update the Lifecycle Visualization client.
2. For Lifecycle Visualization versions 13.1.x and below, do the following:
   a. Log in to SQLPlus from the command line with a user account that has SYSDBA privileges.
   b. Run the following command:

      ```
      exec eaicl_p_clean;
      ```
3. Upgrade the ClearanceDB instance from Oracle 12c to Oracle 19c.
4. Log in to SQLPlus from the command line with a user account that has SYSDBA privileges and run:

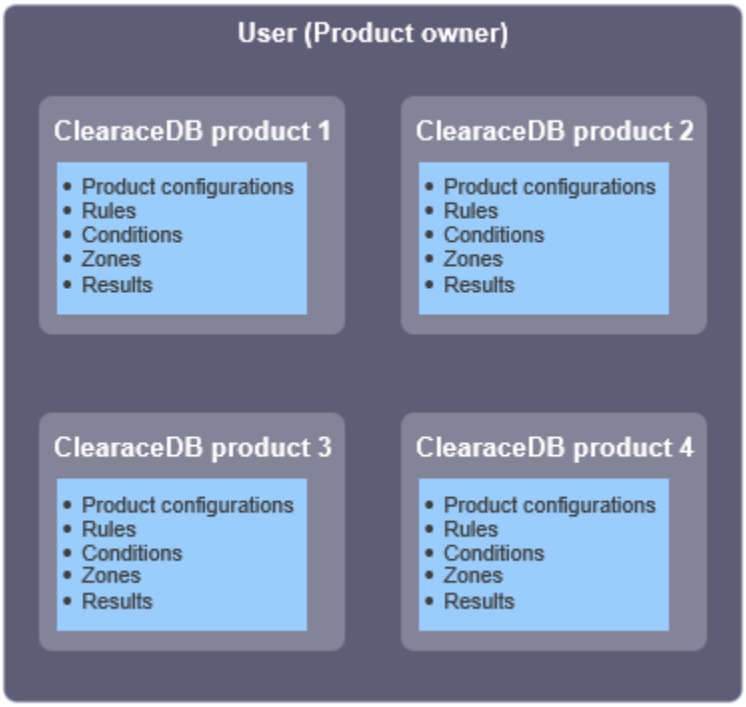
   ```
   GRANT CREATE JOB TO <Your Clearance DB user>;
   ```
5. Log in to SQLPlus from the command line as <ClearanceDBUserName> and run the **upd_ClearanceDB_user.sql** script.

Create 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** , **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*

- (Linux) *<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

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

Example:

```
sqlplus "sys/password@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*
   - (Linux) *<Mockup installation directory>/Visualization/bin/ClearanceDB/Server*

2. At the command prompt, type

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

   where

For this parameter	Type
owner	The user name of the Oracle user account. For example, **CLOWNER**.
password	The password for the Oracle user account. For example, **CLOWNER**.
connection	The Local Net Service Name for the ClearanceDB Oracle database.
logfile	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*.
dflt_tablespace	**CL_TABLE**. This is the name of a ClearanceDB tablespace.
indx_tablespace	**CL_INDEX**. This is the name of a ClearanceDB tablespace.

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 will display a message similar to this:

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

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.

Role	Description
CONFIG	This role allows auxiliary users to define and maintain the ClearanceDB process configuration (per product) for all of the products in the database.
RULES	This role allows auxiliary users to define and maintain ClearanceDB rules and conditions (per product) for all of the products in the database.
ZONES	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.
MAINT	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.
CALC	This role allows auxiliary users to perform the incremental clearance analysis against the ClearanceDB requirement rulebase either from Mockup or with the Clearance Calculator.
UPDATE	This role allows auxiliary users to update the ClearanceDB database with clearance analysis results.
ISSUES	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.
CLDB	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.
ADMIN	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.
VIS	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.

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):

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

where

Parameter	Description
dba	The user name of the SYS or SYSTEM user.
password	The SYS or SYSTEM password.
connection	The Local Net Service Name for the ClearanceDB Oracle database.
logfile	The name that you want to give the log file that is created when you run the *cre_aux_ClearanceDB_user.sql* script. For example, *create_auxuser_log.txt*.
AuxClearanceDBUserName	The name that you want to use for the Oracle ClearanceDB auxiliary user.
AuxUserClearanceDBPassword	The password that you want to use for the Oracle ClearanceDB auxiliary user.
DefaultTableSpaceName	Oracle default tablespace name to be assigned to the ClearanceDB auxiliary user (cannot be of temporary type). For example, **USERS**.
TemporaryTableSpaceName	Oracle temporary tablespace name to be assigned to the ClearanceDB auxiliary user (must be of temporary type). For example, **TEMP**.
ClearanceDBUserName	Oracle ClearanceDB Oracle user account name. For example, **CLOWNER**.

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

Parameter	Description
owner	The user name of the Oracle user account. For example, **CLOWNER**.
password	The password for the Oracle user account. For example, **CLOWNER**.
connection	The Local Net Service Name for the ClearanceDB Oracle database.
logfile	The name that you want to give the log file that is created when you run the *upd_aux_ClearanceDB_user.sql* script. For example, *updateauxlog.txt*.
AuxClearanceDBUserName	The Oracle ClearanceDB auxiliary user name.

Parameter	Description
RoleAction	GRANT or REVOKE.
RoleName	Any defined ClearanceDB role name. For example, **VIS**. If empty all roles are revoked.

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

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

Example:

```
Sqlplus SYSTEM/MANAGER@ORCL @del_aux_ClearanceDB_user.sql
del_vis_orcl.log VIS
```

Note:

The ClearanceDB SQL scripts are case-sensitive.

SQL script reference

Use the following scripts to manage the ClearanceDB database.

Note:

The ClearanceDB SQL scripts are case-sensitive.

Database management scripts

Use this script	To do this
cre_aux_ClearanceDB_user.sql	Create auxiliary ClearanceDB users. **Usage:** **sqlplus system/manager@connection @cre_aux_ClearanceDB_user logfile user password dflt_tablespace temp_tablespace owner**
cre_ClearanceDB_user.sql	Create ClearanceDB users. **Usage:** **sqlplus system/manager@connection @cre_ClearanceDB_user logfile user password dflt_tablespace temp_tablespace**
del_aux_ClearanceDB_user.sql	Delete auxiliary ClearanceDB users. **Usage:** **sqlplus system/manager@connection @del_aux_ClearanceDB_user logfile user**
upd_aux_ClearanceDB_user.sql	Update auxiliary ClearanceDB users. **Usage:** **sqlplus user/passwd@connection @upd_aux_ClearanceDB_user logfile auxuser action role**
upd_ClearanceDB_user.sql	Update ClearanceDB users.

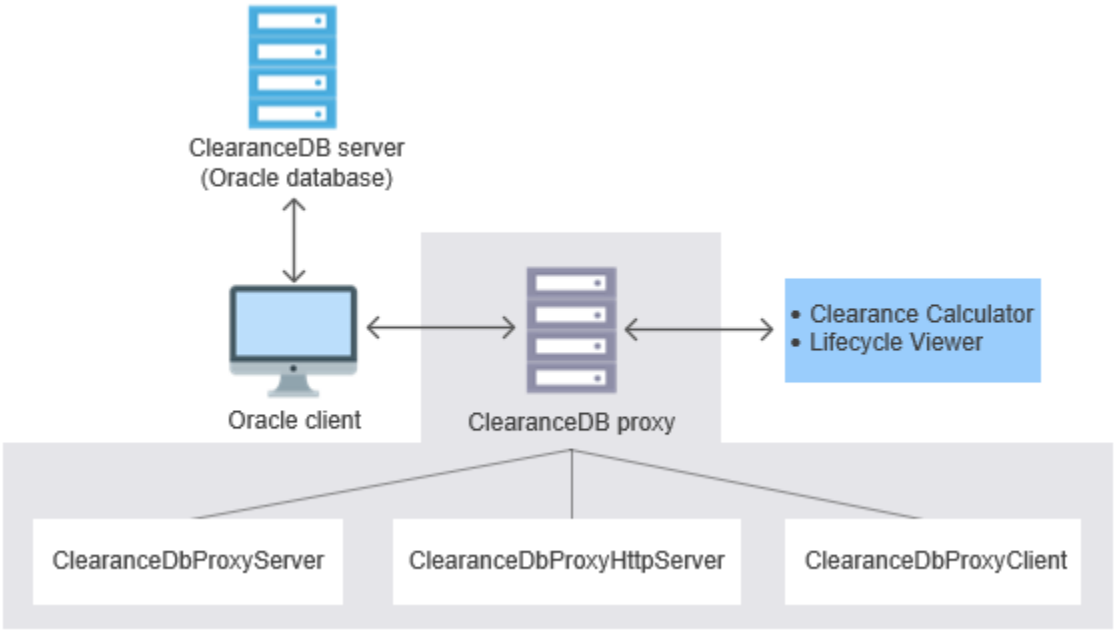
Use this script	To do this
	Usage: **sqlplus user/passwd@connection @upd_ClearanceDB_user logfile dflt_tablespace indx_tablespace**

Configuring the Proxy (expanded version)

ClearanceDB Proxy overview

ClearanceDB Proxy is essentially the “bridge” that connects the Clearance Calculator and the viewer with the database. It consists of three applications, the ClearanceDbProxyServer, ClearanceDbProxyHttpServer, and the ClearanceDbProxyClient. Only the ClearanceDbProxyServer and ClearanceDbProxyHttpServer (optional) are 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.

For information about system hardware and software requirements, choose from links found in the *Hardware and Software Certifications* knowledge base article on Support Center (scroll down to the *Teamcenter Visualization Software Certifications* heading).

For supported platform requirements, see *Supported platforms*.

1. Open the Oracle Instant Client downloads page:

   http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html

   > 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 64-bit (x64).

3. After accepting the Oracle license agreement, download the Basic (not Basic Lite) package for your platform.

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 *oci.dll* exists.

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<version>**

9. Save the file.

Install the Oracle Instant Client (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:

   http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html

> 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.y 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:

   > Note:
   >
   > These global settings are used to populate the *Clearance.cfgproxy* file, which is created when you use the *create_proxy.pl* script to create a new instance of the proxy. If necessary, you can adjust the settings in *Clearance.cfgproxy* to override the global defaults.

   OracleClientDirectory

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

   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.

STANDARD_PROXY

This is an optional setting. The default communication between the ClearanceDB client applications and the ClearanceDB Proxy uses TCP/IP protocol. Choose one of the following:

True — Scripts use the default TCP/IP-based communication protocol.

False — Set to false for the scripts to only execute HTTP flow.

> Note:
>
> If this setting is not defined, the default TCP/IP protocol is used.

4. Specify the following settings in the *Section 5: ClearanceDB HTTP Proxy Configuration* section of the file to configure the Http proxy:

> Note:
>
> These settings are required to use HTTP/S-based communication protocol between the client and Clearance database server. SSO setup is prerequisite to enabling this feature.
>
> For more information on setting up the SSO environment, see Security Services Installation/ Customization in *Security Services Configuration*.

WEB_SERVICE

Specifies whether the communication between the ClearanceDB client applications and the ClearanceDB Proxy uses the HTTP protocol. Choose one of the following:

True — HTTP-based communication protocol.

False — This feature is not enabled.

WEB_SERVICE_DISPLAY_NAME

Specifies the display name for the web service.

WEB_SERVICE_DESCRIPTION

Specifies the description for the web service.

PROXY_SERVER

Specifies the ClearanceDB Proxy Server that will communicate with the Oracle Database.

PROXY_SERVER_PORT

Specifies the port number for the ClearanceDB Proxy Server to use.

WEB_SERVICE_PROXY

Specifies the web service URL where the ClearanceDB HTTP Proxy is running.

WEB_SERVICE_PORT

Specifies the port number for the ClearanceDB HTTP Proxy Server to use

SSO_SERVICE_IDENTITY_URL

Specifies the URL used by the TcSS solution for single sign-on. For example, **SSO_SERVICE_IDENTITY_URL=https://myserver.com/TcClearanceIdentity** where TcClearanceIdentity is the TcSS identity provider name defined by Teamcenter Web Application Manager.

CLEARANCE_APPID

Specifies the application id defined in the TcSS Application Registry. For example, **CLEARANCE_APPID=TcClearanceDB**.

WEB_SERVICE_URL

Specifies the HTTP or HTTPS web service URL used by ClearanceDB. The URL should point to a reverse proxy, which will redirect the request to the ClearanceDB Http proxy server running behind the firewall. For example, **WEB_SERVICE_URL=https://myserver.com/TcVisCLDB**.

> Note:
>
> You must specify this value to stop remote proxy if you are using cfgproxy file as an input.

5. Save the *Clearance.cfgglobal* file.

Configure the ClearanceDB HTTP Server

ClearanceDB Proxy HTTP Server overview

ClearanceDbProxyHttpServer.exe is a web service executable which is added to the Teamcenter lifecycle visualization ClearanceDB solution to support HTTP/S-based communication between Teamcenter visualization clients and the Clearance database.

ClearanceDbProxyHttpServer.exe relies on TcSS (Teamcenter Security Services) to provide SSO (single sign-on) and user authentication.

To enable the *ClearanceDbProxyHttpServer.exe* feature, set up the following components.

- **Set up the Apache web server**
- **Install SSL certificates**
  - **Extract the required key/certificate information**
  - **Import certificates into the client machine**
  - **Export certificates as a .pfx**
  - **Import the certificate into the Java Virtual Machine certificate trust store**
- **Set up the Teamcenter Web Application Manager**
- **Configure Teamcenter for ClearanceDB Http Server**
- **Configure secure mode service (optional)**

- **Configure the proxy and database connection**
- **Configure the ClearanceDB PERL scripts**

Set up the Apache web server

Configure the ClearanceDB Proxy HTTP Server (HTTP Proxy / Web Service) to run behind an Apache reverse proxy.

1. Download the **Apache HTTP server**.
2. Configure SSL (secure sockets layer) information in either the http-vhosts.conf file or the httpd-ssl.conf file.
3. Set the SSL attributes ProxyPass and ProxyReversePass in the Apache config files.

   Example:

   SSLEngine on

   SSLCertificateFile ${SRVROOT}/key/certificate.crt

   SSLCertificateKeyFile ${SRVROOT}/key/decrypted.key

   ProxyPass "/TcVisCLDB" "http://myclearanceserver.com:9090/TcVisCLDB"

   ProxyPassReverse "/TcVisCLDB" "http://myclearanceserver.com:9090/TcVisCLDB"

Refer to the Apache documentation for more information on the correct syntax to configure the reverse proxy.

Installing SSL certificates

Extract the required key/certificate information

The Apache Web Server requires certificate authentication as part of the secure communication protocol between the client and server. Install the certificate on both the client and server. Ensure the certificate is generated and signed by a trusted source.

OpenSSL can be installed to extract relevant information from the certificates. The private key and the public certificate can be extracted from the generated certificate (usually a pfx file).

Following are the commands to extract the required information.

Private key

Use the following command to extract the private key:

```
openssl pkcs12 -in [yourfile.pfx] -nocerts -out [keyfile-encrypted.key]
```

The system will ask for the import password for the .pfx certificate file. This is the same password that was used to protect the keypair when the .pfx file was first created.

When the import password is entered OpenSSL requests a new password, twice. This new password protects the extracted private .key file.

Public certificate

Use the following command to generate the public certificate:

```
openssl pkcs12 -in [yourfile.pfx] -clcerts -nokeys -out [certificate.crt]
```

Uncrypted key

Use the following command to extract the encrypted key:

```
openssl rsa -in [keyfile-encrypted.key] -out [keyfile-decrypted.key]
```

The password is the same password used to protect the private key pair.

PEM formats

You may be required to convert the keys/certificates to PEM format. PEM format keys are required to run the ClearanceDB Proxy HTTP Server in secure mode (currently only supported on Windows without using an Apache reverse proxy configuration).

Use the following command to convert the keys/s=certificates to PEM format:

```
openssl rsa -in [keyfile-encrypted.key] -outform PEM -out [keyfile-
encrypted-pem.key].
```

```
openssl pkcs12 -in [yourfile.pfx] -out [certificate.pem]
```

For more information on converting certificates to appropriate formats, refer to the openssl documentation (**https://www.openssl.org/**).

Import certificates into the client machine

Use the Microsoft Management Console (MMC) to import the certificates on Windows.

1. Start the MMC (Start→Run→MMC).

2. Choose **File→Add/Remove Snap In**.

3. In the Snap-In window, double-click **Certificates**.
4. Select **Computer Account**.
5. Select **Local Computer→Finish**.
6. Click **OK** to exit the Snap-In window.
7. Click **+** next to **Certificates→Personal→Certificates**.
8. Right-click **Certificates** and select **All Tasks→Import**.
9. Click **Next**.
10. Click **Browse**.
11. Select the .cer, .crt, or .pfx you want to import, and then click **Open**.
12. Click **Next**.
13. Select **Automatically select the certificate store based on the type of certificate**.
14. Click **Finish** and then **OK**.

Export certificates as a .pfx

Use the Microsoft Management Console (MMC) to export certificates.

1. Start the MMC (Start→Run→MMC).
2. Choose **File→Add/Remove Snap In**.
3. In the Snap-In window, double-click **Certificates**.
4. Select **Computer Account**.
5. Select **Local Computer→Finish**.
6. Click **OK** to exit the Snap-In window.
7. Click **+** next to **Certificates→Personal→Certificates**.
8. Right-click the certificate for the domain and select **All Tasks→Export**.
9. Click **Next**.

10. Click **Yes, export the private key**.

11. Do the following.

    a. Select **Personal Information Exchange - PKCS#12 (.PFX)** as the certificate file format.

    b. Select **Include all certificates in the certificate path if possible**.

    c. Select **Export all extended properties**.

    d. Click **Next**.

12. Enter a password for your private key and then click **Next**.

13. Click **Save** to save your .pfx.

14. Click **Next**, then click **Finish**.

15. Click **OK**.

Import the certificate into the Java virtual machine certificate trust store

Since SSO client relies on certificate validation, the certificate must be imported into the client JVM so that it can be validated by reverse proxy.

1. Check the JAVA_HOME environment variable to determine the Java version used by the system.

2. At a command prompt, enter the following, where server_name is the web server and the public.crt is the public certificate for the web service.

   ```
   <JAVA_HOME>/bin/keytool -import -alias <server_name> -keystore
   <JAVA_HOME>/jre/lib/security/cacerts -file public.crt
   ```

Certificates and Linux

Teamcenter visualization maintains its own certificate store on Linux. Any new certificate needs to be copied over to the *<Install>/Visualization/etc/certs* folder.

Certificates and Mozilla/Firefox

Mozilla/Firefox clients manage certificates differently. The certificates can be imported through **Preferences→ Advanced→Certificates** (Certificate Manager).

Set up Teamcenter Web Application Manager

Apache Tomcat must be installed prior to setting up Web Application Manager.

Web Application Manager is a JAVA-based application that deploys its WAR files in Tomcat. This application consists of the Identify provider and Login provider.

For detailed instructions on setting up the Web Application Manager, see Security Services Installation/Customization in *Security Services Configuration*.

Configure Teamcenter for ClearanceDB HTTP Server

It is best practice to use the Teamcenter SSO (single sign-on) setup in conjunction with Teamcenter Web Application Manager. This will enable SSO authentication against the Teamcenter user base via the CLEARANCE_APPID property defined in the *Clearance.cfgproxy* file. The value can be set to **Teamcenter**, or equivalent, for authentication against the Teamcenter user base. If authentication is required against a different LDAP, this information should be configured in Teamcenter Web Application Manager Login Service.

ApacheDS (Apache directory studio) can be installed to work as an LDAP for testing.

For more information on configuring ApacheDS to work with TcSS (Teamcenter Security Services), see Security Services Configuration in the Teamcenter help.

Configure secure mode service

The ClearanceDB Proxy HTTP Server can also work as a forward proxy, as well as a reverse proxy. The communication protocol as a forward proxy can be secure or unsecure.

SSO (single sign-on) is a prerequisite for both deployment modes to perform authentication. The property WEB_SERVICE_URL, defined in *.dbc file, determines the communication protocol.

Mode	Value	Description
Reverse Proxy	WEB_SERVICE_URL=https://myserver.com /TcVisCLDB	The client communicates to the Apache reverse proxy, while the ClearanceDB Proxy HTTP Server runs behind this proxy. This is the recommended usage.
Forward Proxy - unsecure	WEB_SERVICE_URL=http://myserver.com:9090/TcVisCLDB	The client communicates directly with the ClearanceDB Proxy HTTP Server in unsecured mode using the HTTP protocol.
Forward Proxy -secure	WEB_SERVICE_URL=https://myserver.com:443/TcVisCLDB	The client communicates directly with the ClearanceDB Proxy HTTP Server in secured mode and uses certificate validation. Currently, only the certificate issuer name between the client and server is validated.

Using the ClearanceDB Proxy HTTP Server in forward proxy secure mode (Windows only) requires the following additional setup.

- For both the server and client, set the environment variable TCVIS_PC_ENABLE_CLIENT_CERTIFICATE_SUPPORT=True

- Do the following.

  - Define the **ClearanceDbProxyHttpServer.properties** properties file.

  - Copy the properties file into the same location as the server executable.

  - Add the certificates to the same location, or give a complete path to the certificate location.

Example:

openSSL.server.privateKeyFile = private_key.pem
openSSL.server.certificateFile = public_cert.pem
openSSL.server.verificationMode = relaxed
openSSL.server.verificationDepth = 9
openSSL.server.loadDefaultCAFile = true
openSSL.server.cipherList = ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH
openSSL.server.privateKeyPassphraseHandler.name = KeyFileHandler
openSSL.server.privateKeyPassphraseHandler.options.password = securePassword
openSSL.server.invalidCertificateHandler.name = AcceptCertificateHandler
openSSL.server.extendedVerification = false
openSSL.server.cacheSessions = true
openSSL.server.sessionIdContext = ClearanceDbProxyHttpServer
openSSL.server.sessionCacheSize = 100
openSSL.server.requireTLSv1 = false

Refer to the OpenSSL documentation for details on the property values.

Configure the proxy and database connection

Use the options in the *Clearance.cfgglobal* and *global.dbc* files to configure *ClearanceDbProxyHttpServer.exe*. These options control SSO authentication and the communication protocol for ClearanceDB.

For more details on using these options, see the *Clearance.cfgglobal* and *global.dbc* files found in <Install Path>Visualization\Program\ClearanceDB\Support\Work_Area_Template.

Configure the ClearanceDB PERL scripts

Do the following to define the PERL scripts:

- Define the following settings in the .dbc and. cfgproxy files to control communication between the Teamcenter Visualization client and server.

  - WEB_SERVICE=true, communication protocol for ClearanceDB will be HTTP.

  - WEB_SERVICE=false, communication protocol for ClearanceDB will be TCP/IP.
  - STANDARD_PROXY=true, used to start/stop/ping only the TCP/IP proxy.

- It is best practice to create a .dbc file with the relevant connection information so that the communication with ClearanceDB uses the secured HTTP communication protocol while using these scripts. The following scripts have command line arguments which are used to create a local .dbc file used for TCP/IP communication.

  - create_product.pl
  - list_all_products.pl
  - ping_database.pl
  - ping_proxy.pl
  - stop_proxy.pl

- So that the settings from the proxy are used for communication, set UploadMethod=Proxy in the **Clearance.cfgglobal** file. Some scripts, like update_product.pl, rely on these settings. For more details, see **Updating products in the database**.

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 configure your environment in the *Teamcenter Utilities*.

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\oracle_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.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, including the ClearanceDB HTTP Proxy. 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.

   Console

   Specifies to manually run the process, without using a Windows service, from the proxy directory or from the Windows command prompt.

   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 mebibyte 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:

- **SESSIONS** — Obtaining and releasing of Oracle sessions. If **LogFlags** is not included in the *Clearance.cfgproxy* file, the **SESSIONS** 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 **SESSIONS** 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 **SESSIONS**, **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
```

6. Save the *Clearance.cfgproxy* file.

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:

   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.

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:

To	Do this
Start the service from the proxy directory	Right-click the *Clearance.cfgproxy* file and choose **Start Proxy**.
Start the service from the command line	Type: **sc start <service name>** Note: You must provide the *service name* not the *display name* of the service. To display a list of the services on your machine, type: **sc query state= all**. Scroll through the list until you find the entry for the ClearanceDB Proxy service, which includes the service name. Tip: For descriptions of all of the commands supported by the SC program, type **sc**.
Start the service from the Control Panel	a. Open **Administrative Tools** and then **Services**.

To	Do this
	b. In the **Services** list, right-click the ClearanceDB Proxy service and choose **Start**.
Stop the service from the proxy directory	Right-click the *Clearance.cfgproxy* file and choose **Stop Proxy**.
Stop the service from the command line	Type: **sc stop <service name>**
Stop the service from the Control Panel	a. Open **Administrative Tools** and then **Services**. b. In the **Services** list, right-click the ClearanceDB Proxy service and choose **Stop**.
Delete the proxy service	In the proxy directory, right-click the *Clearance.cfgproxy* file and choose **Remove Proxy from Services**.

Note:

If **WEB_SERVICE=True**, the same commands and behavior are applicable to the ClearanceDB HTTP Proxy. If **STANDARD_PROXY=False**, the scripts will execute only the HTTP flow.

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:

To start the proxy	Do this
(Windows) From the shortcut menu	Right-click the *Clearance.cfgproxy* file and choose **Start Proxy**.
From the command prompt	Type **start_proxy.pl <path_to_proxy_directory>\ clearance.cfgproxy**

To start the proxy	Do this
	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].

If **WEB_SERVICE=True**, then the script will start the ClearanceDB HTTP Proxy. The system will display an additional message, **Clearance Web Service started and running. Type http:// [WEB_SERVICE_PROXY]: [WEB_SERVICE_PORT] to use it or type CRLT+C to finish it**.

Ping a ClearanceDB Proxy

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

Do any of the following:

To ping the proxy	Do this
(Windows) From the shortcut menu	Right-click a DBC file that references the proxy and choose **Ping Proxy**.
From the command prompt	Type **ping_proxy.pl <path_to_DBC_file>\<DBC_file>** Example: **ping_proxy.pl C:\ClearanceDB_Work_Area\ global.dbc** **ping_proxy.pl C:\ClearanceDB_Work_Area\ Proxies\Proxy1\proxy1.dbc**

Note:

If **WEB_SERVICE=True**, the script will ping the HTTP proxy. The system will display an additional message, **Successfully connected to the http proxy at https:// [WEB_SERVICE_URL]**.

Ping the database

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

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

To ping the database	Do this
(Windows) From the shortcut menu	Right-click a DBC file and choose **Ping Database**.
From the command prompt	Type **ping_database.pl <path_to_DBC_file>\<DBC_file>** Example: **ping_database.pl C:\ClearanceDB_Work_Area\ global.dbc** **ping_database.pl C:\ClearanceDB_Work_Area\ Proxies\Proxy1\proxy1.dbc**

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:

To list products in the database	Do this
(Windows) From the shortcut menu	Right-click a DBC file and choose **List All Products**.
From the command prompt	Type **list_all_products.pl <path_to_DBC_file>\<DBC_file>** Example: **list_all_products.pl C:\ClearanceDB_Work_Area\ global.dbc** **list_all_products.pl C:\ClearanceDB_Work_Area\ Proxies\Proxy1\proxy1.dbc**

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**.
From the command prompt	Type **stop_proxy.pl <path_to_proxy_directory>\ <proxy_configuration_file>** Example: **stop_proxy.pl C:\ClearanceDB_Work_Area\ Proxies\Proxy1\ clearance.cfgproxy**

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.

> Note:
>
> If **WEB_SERVICE=True**, the script will stop the HTTP proxy.

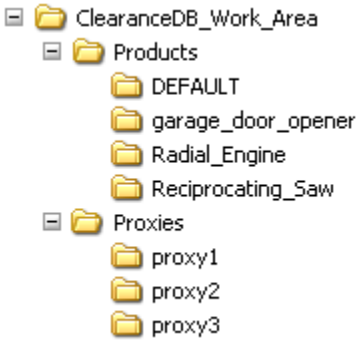
Configuring the Client and Clearance Calculator (expanded version)

Setting up the work area

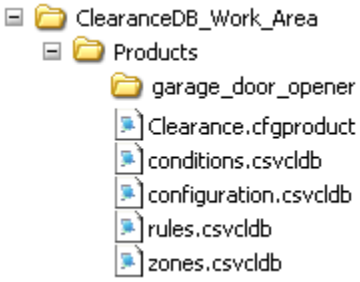
The ClearanceDB Work Area is the location on your machine where you can create and manage ClearanceDB products. Created by the *create_ClearanceDB_working_dir.pl* script, the ClearanceDB Work Area contains configuration files that you use to create a connection with the database, run the proxy, and update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



Product-specific configuration files are located in each product directory

A proxy-specific configuration file is located in each proxy 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. (Linux) 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 indeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

Yes — If indeterminable variant conditions exist, evaluate them lexicographically.

> Note:
>
> Variant conditions that cannot be evaluated based on textual representations are indeterminable.

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.
EXCUDEDUIDPAIRS	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 **EXCUDEDUIDPAIRS** 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 .plmxml 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:
>
> **TeamcenterWebServerPath=http://tcserver:80/tc8_2008/**

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 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 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**.

Section 2: ClearanceDB Proxy Configuration

OracleClientDirectory

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

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 upload data to 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

-d <value> — Specifies the clearance requirement, 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 email messages to owners, as defined in your Clearance Manager preferences.

> Note:
>
> - This option may produce a large number of e-mails and images (if specified).
> - For this option to work, each part must have metadata containing the owner's email address.

-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.

-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.

> Note:
>
> The *results.txt* file is designed for use by Teamcenter lifecycle visualization mockup and is not intended for external use. If you choose to use this file in an external process you do so at your own risk, as this file is unsupported.

ElementType

The clearance element type. Valid options are:

Part
Component
EndItem
EndItemComponent

The default element type is **Part**.

ResultsGranularity

Specifies how results are reported. Valid options are:

Part
Element

The default results granularity is **Element**.

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

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

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

> 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

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

> 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) |

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

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.

SnapshotImageCPPointsColor

(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.

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.

For this parameter	Type
	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.
WEB_SERVICE	HTTP communication protocol for ClearanceDB.
WEB_SERVICE_URL	HTTP or HTTPS web service URL used by ClearanceDB.
SSO_SERVICE_LOGIN_URL	The URL used by the TcSS solution for single sign-on.
CLEARANCE_APPID	The application id defined in the TcSS Application Registry.

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:

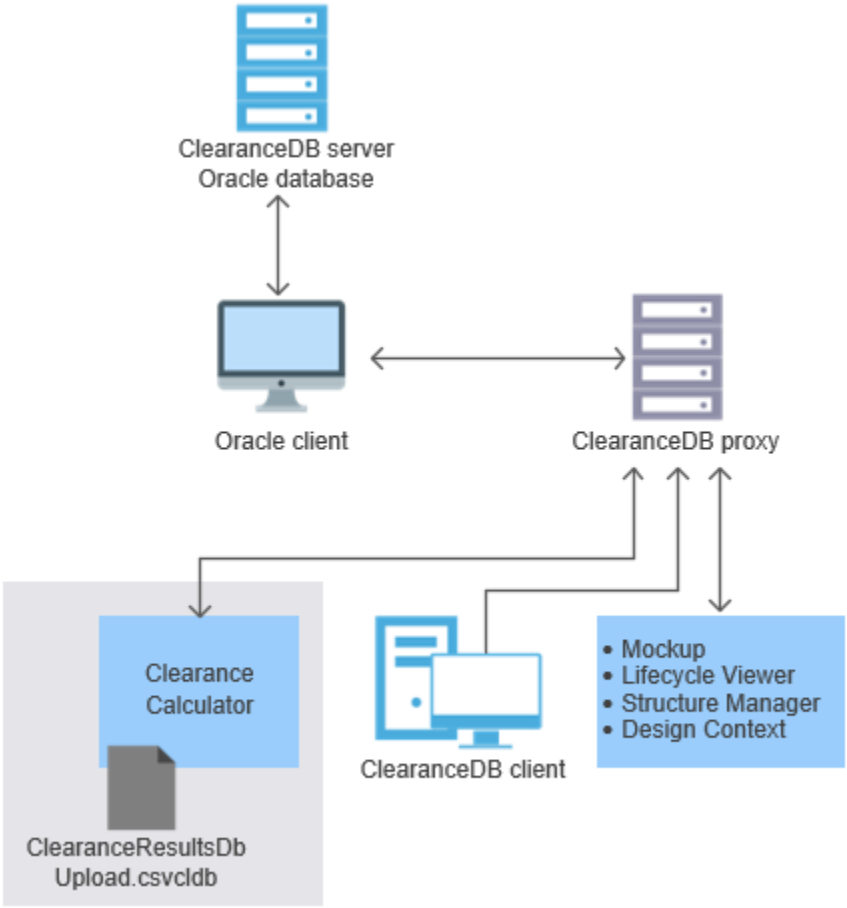
   0_Protocol = http

4. Save the file.

4. Performing the analysis

Performing analysis overview

Clearance checks are performed using the viewer (stand-alone Mockup, the Lifecycle Viewer, Structure Manager, or Design Context) 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 uploaded to the database.

To perform analysis on a product and update the database with the results, you must take the following steps:

- Configure the **batch analysis options** for the Clearance Calculator

- Use the Clearance Calculator to **perform analysis on your product**
- **Upload the results** generated by the Clearance Calculator to 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**.

Setting up products for analysis

Perl and SQL script reference

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

> Note:
>
> The ClearanceDB SQL scripts are case-sensitive.

Proxy management scripts

Use this script	To do this
create_proxy.pl	Create a new ClearanceDB Proxy. **Usage:** **create_proxy.pl <proxy name>**
ping_database.pl	Test database connectivity. **Usage:** **ping_database.pl [<proxy host name> <port> <connect data>]**
ping_proxy.pl	Test to determine if the proxy is running. **Usage:** **ping_proxy.pl [<proxy host name> <port>]**
start_proxy.pl	Start a ClearanceDB Proxy. **Usage:** **start_proxy.pl <proxy configuration file>**
stop_proxy.pl	Stop a ClearanceDB Proxy. **Usage:** **stop_proxy.pl <proxy configuration file>**

Product management scripts

Use this script	To do this
analyze_managed_product.pl	Perform analysis on a Teamcenter managed product. **Usage:** **analyze_managed_product.pl <product>**
analyze_product.pl	Perform analysis on a product. **Usage:** **analyze_product.pl <product>**
copy_product.pl	Copy an existing ClearanceDB product and use it to create a new product in both your work area and the database. **Usage:** **copy_product.pl <product name> <new product name>**
create_product.pl	Create a new product. **Usage:** **create_product.pl <name> [-uc:-ur:-ud:-uz:-ua] [-ds <datasource> -cd <connection name>]**
delete_product.pl	Delete a ClearanceDB product from both your work area and the database. **Usage:** **delete_product.pl <product name>**
list_all_products.pl	List all of the ClearanceDB products in the database. **Usage:** **list_all_products.pl [<proxy host name> <port> <connect data>]**
list_product.pl	Display information about a ClearanceDB product. **Usage:** **list_product.pl <product>**
rename_product.pl	Rename a ClearanceDB product in both your work area and the database. **Usage:** **rename_product.pl <product name> <new product name>**

General administration scripts

Use this script	To do this
create_ClearanceDB_working_dir.pl	Create the ClearanceDB Work Area. **Usage:** **create_ClearanceDB_working_dir.pl <path> [-df]**
create_default_product.pl	Create the default product. **Usage:** **create_default_product.pl <path>**
update_product.pl	Upload product information to the database. **Usage:** **update_product.pl [product] -uc:-ur:-ud:-uz:-ua [-df]**

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 your 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 uploaded to 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

For this parameter	Type
`path`	The path to the location on your machine where you want to create the default product.

   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.csvcldb*
- *configuration.csvcldb*
- *rules.csvcldb*
- *zones.csvcldb*

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:

To upload configuration files	Do this

From the command prompt	Type
	`update_product.pl [<product>] -uc:-ur:-ud:-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 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.

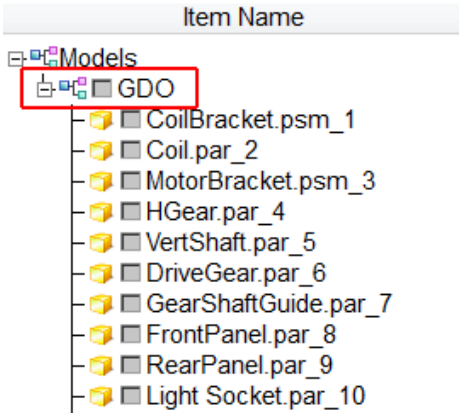
1. Open a command prompt and navigate to the ClearanceDB Work Area.

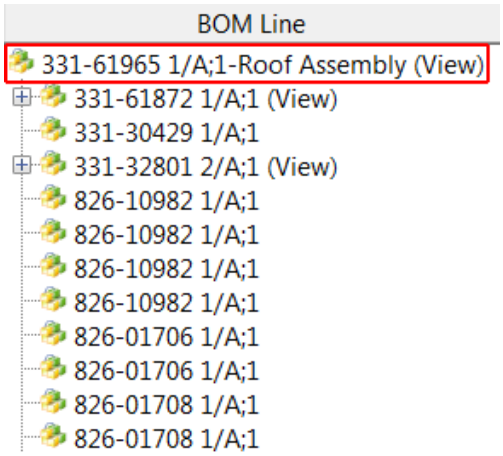
> Note:
>
> You must create new products from the ClearanceDB Work Area.

2. From the command prompt, type

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

   where

This parameter	Does this
name	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"*. 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** 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	Does this
	 **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.csvcldb* file. • **-ur** — Create *rules.csvcldb* and *conditions.csvcldb* files. • **-uz** — Create a *zones.csvcldb* file • **-ua** — Create copies of all of the product configuration files.
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:

To upload configuration files	**Do this**
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**
(Windows) From the shortcut menu	In the product directory, right-click any configuration file and choose **Upload to ClearanceDB**.

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.

Note:

You can also use the product-specific configuration file to override any of the settings contained in the *Clearance.cfgglobal* global configuration file. To override the global settings on a per-product basis, copy the settings from the *Clearance.cfgglobal* file to the *Clearance.cfgproduct* file and modify them appropriately.

1. In the ClearanceDB Work Area, navigate to the product folder.

2. Using a text editor, open the *Clearance.cfgproduct* file and specify the following settings:

For this option	Type this
ProductInputFile	The file name and full path to the product. Example: **ProductInputFile= c:\fishing_reel.jt** Note: This setting is applicable only to product data that is not stored in Teamcenter.
ItemKey	The Teamcenter values (strings) of the multifield key properties for the item containing your product. Example: **ItemKey=item_id= 000092,MFK_strkey1=val1** 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.
ItemId	The Teamcenter item ID of the top level assembly of your product. Example: **ItemId=000092**

For this option	Type this
	Note: This setting corresponds to the **–item** argument for the Teamcenter BOMwriter utility.
ItemRevision	The Teamcenter revision of the top level assembly of your product. Example: **ItemRevision=A** Note: This setting corresponds to the **–rev** argument for the Teamcenter BOMwriter utility.

3. Save the file.

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 requirement 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 requirement 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 requirement, 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.

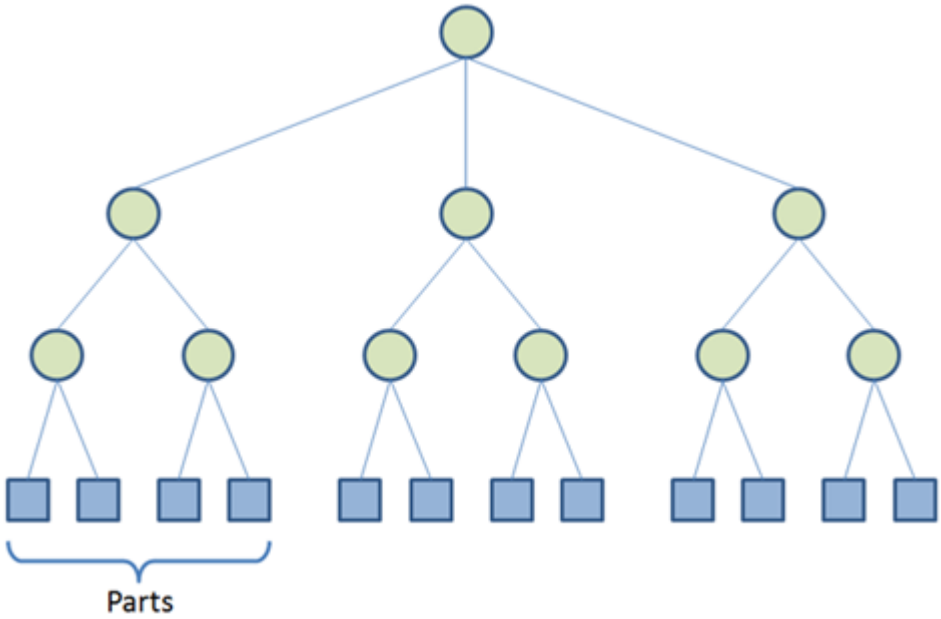
Clearance elements and CAD components

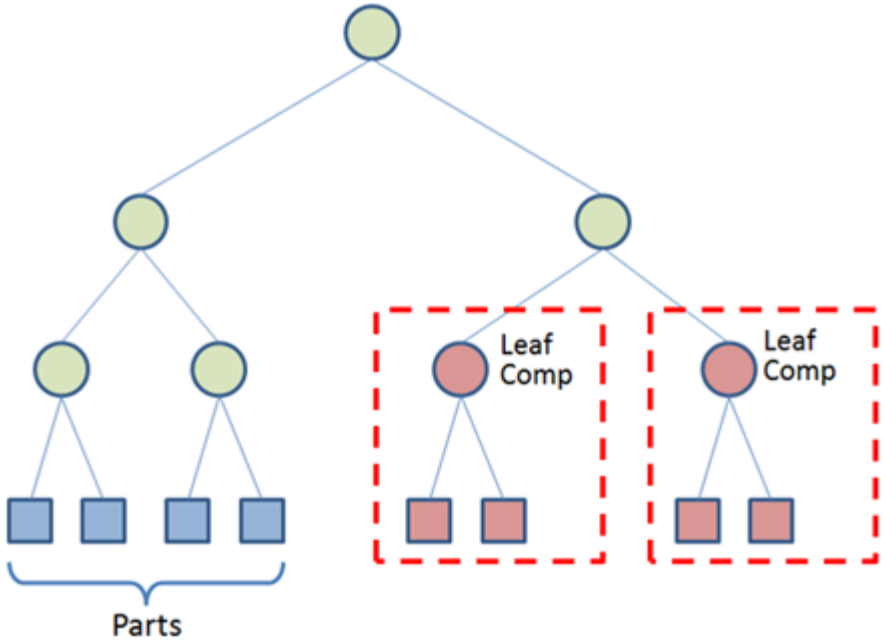
Elements and CAD components are the parts and branches of product structure recognized for clearance analysis in Teamcenter lifecycle visualization. They are used to control the granularity of the analysis.

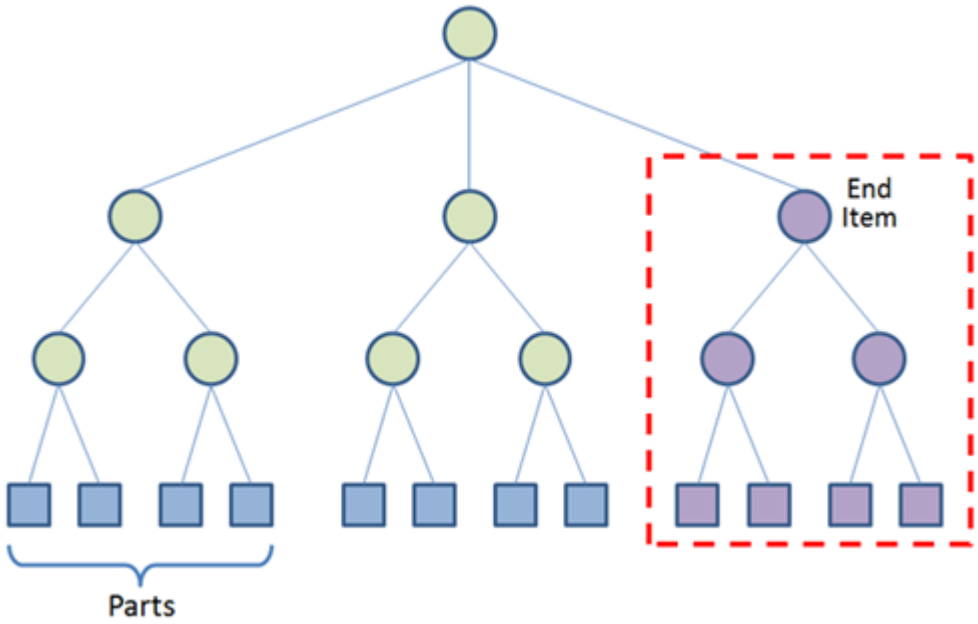
There are three element types: parts, leaf components, and end items. Any combination of these elements can exist in the product structure for use during a clearance analysis. However, the part element type is always turned on.

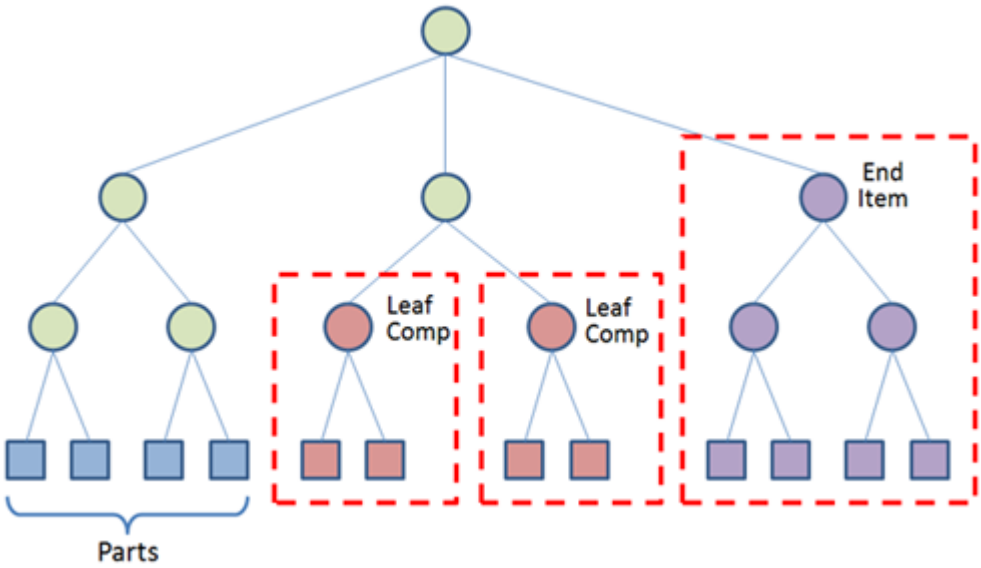
If an element contains other elements within it, no clearance issues between those elements are reported in the clearance results list. Also, any two leaf components or end items can have many clashes between them, but only the single, most severe result between any two elements is reported.

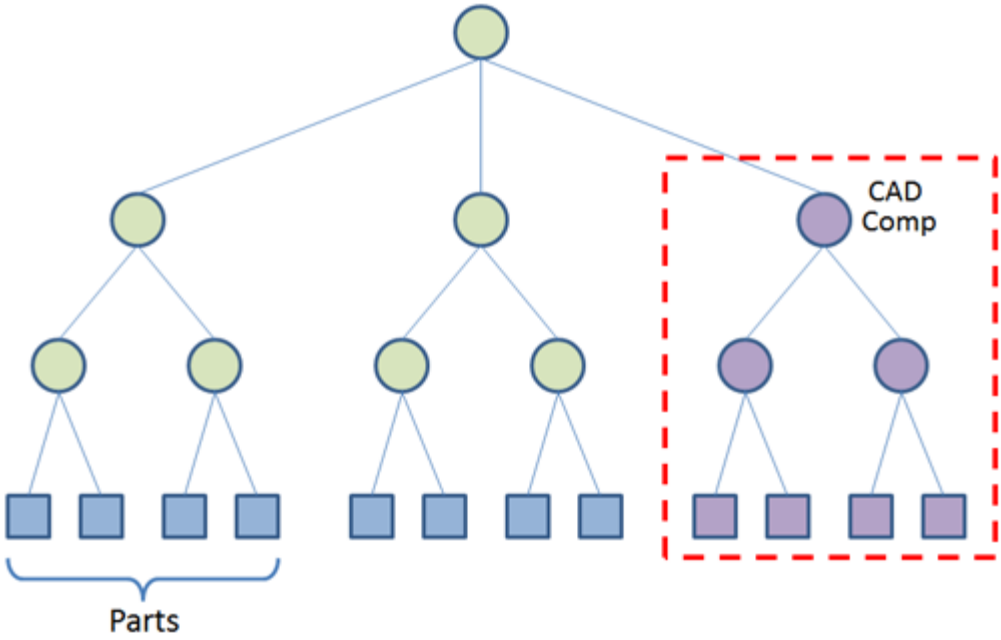
The items that appear in the clearance results list are always elements. A CAD component is essentially an end item that allows any number of clearance results between itself and other nodes. For this reason, a CAD component is not considered an element.

Element type	Behavior
Part	 **Analysis** 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. **Results**

Element type	Behavior
	Only issues between parts are displayed in the clearance results, even if leaf components, end items, or CAD components exist in the structure.
Leaf component	 **Analysis** 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. **Results** 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).

Element type	Behavior
End item	 **Analysis** The lowest level elements in the clearance analysis are parts, unless they are overridden by an end-item. In that case, all of the parts underneath the end item node are considered to be part of the same, monolithic element. **Results** 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). Note: End items are branches of product structure where the root node contains the metadata **__PLM_END_ITEM=True**.

Element type	Behavior
Leaf components and end items	 **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

Element type	Behavior
	other elements is displayed. That result appears in the list as the root item (not the parts within the root item).
CAD components	 **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. To configure the clearance analysis, change any of the settings contained in *Section 4: Clearance Analysis configuration* of the **global configuration file**.

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:

To run the Clearance Calculator	**Do this**

From the command prompt	Type
	`analyze_product.pl [<product>]`
	Example:
	`analyze_product.pl fishing_reel`

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.

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

Record type 3 — clearance node pair definition

1 — Record type = 3
2 — Calculator clearance ID
3 — Left calculator node ID
4 — Right calculator node ID

Record type 4 — node definition

1 — Record type = 4
2 — Calculator node ID
3 — Node attribute name
4 — Node attribute value

Record type 5 — clearance definition

1 — Record type = 5
2 — Calculator clearance ID
3 — Clearance attribute name
4 — Clearance attribute value

Performing clearance batch analysis

Introduction to performing clearance batch analysis

You can manually perform clearance analysis on your 3D data outside of the Mockup application. There are two options:

- Use the supplied Perl scripts to create and manage clearance products within a Clearance Work Area on your local machine.
- Perform clearance analysis **without the Perl scripts**.

Creating a Clearance Work Area

You can create an area on your machine from which you can run and manage clearance batch analysis.

> Note:
>
> Perl version 5.03 or later is required for the creation of a Clearance Work Area. 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**.

1. Open a command prompt and navigate to the ClearanceDB Perl scripts, which are located at the following location within the stand-alone Lifecycle Visualization installation:

   > Example:
   >
   > *<Mockup installation directory>/Visualization/Products/Mockup/ClearanceDB*

2. At the command prompt, type

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

   For `path`, type the location on your machine where you want to create the work area. `lo` is the option 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 *Clearance_Work_Area*.

   > Example:
   >
   > `create_ClearanceDB_working_dir.pl c:\ -lo`

A new directory named *Clearance_Work_Area* is created at the specified location.

Create a clearance product

When you create a clearance product, a directory is added to the Clearance Work Area, from which you can specify clearance analysis options, perform the analysis, and access the results.

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

   > Note:
   >
   > You must create new clearance products from the Clearance Work Area.

2. At the command prompt, type

   ```
   create_product.pl <name>
   ```

   where `name` is the name you want to use for the new product folder.

   > Note:
   >
   > If the name contains spaces, enclose it in quotation marks. For example, *"garage door opener"*.

   > Example:
   >
   > `create_product.pl shock_absorber`

   A new directory with the name you specified is created in the Clearance Work Area. The directory contains a unique *Clearance.cfgproduct* file within which you must specify the path to the data you want to analyze. You can also use the file to control the clearance analysis.

Configure Clearance products for analysis

Use 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 Clearance 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`, only major events such as process restarts or analysis failures are reported in the log file.

1. Navigate to the Clearance 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:

For this option	Type
`ClearanceExeOptions`	`-d <value>` — Specifies the clearance requirement, 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 email 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. `-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:

For this option	Type

	`Part`
	`Leaf Component`
	`EndItem`
	`EndItemComponent`
	
	The default element type is `Part`.

`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.
`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.

For this option	Type
	Note: If both `AbsMemLimit` and `RelMemLimit` options are used, the lesser effective value is chosen.
`RulesObject`	How to obtain the clearance requirement. For local clearance analysis, use `Universal Clearance Requirement`.
`ResultsObject`	How to handle the clearance results. For local clearance analysis, use `General Clearance 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)`
	The following entries are examples:
	`Attribute=Translation Date,Part Only |`
	`Attribute=End Item,Part Only |`
	`Attribute=End Item,Part Parent Only |`

For this option	Type

	`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

For this option	Type
	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`.

For this option	Type
`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

For this option	Type
	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.
`SnapshotImageCPPointsColor`	(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

For this option	Type
	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.

5. Navigate to the directory of the clearance product.

6. Using a text editor, open the *Clearance.cfgproduct* file.

7. For `ProductInputFile=`, type the file name and full path to the data you want to analyze.

   Example:

   `ProductInputFile=c:\fishing_reel.jt`

8. If you want to set any of the analysis options differently for a particular clearance product, copy the settings from the *Clearance.cfgglobal* file into the *Clearance.cfgproduct* file.

9. Save the file.

Analyze clearance products from the Clearance Work Area

1. Navigate to a product folder within the Clearance Work Area.

2. Right-click the *Clearance.cfgproduct* file and choose **Analyze Product**.

   -or-

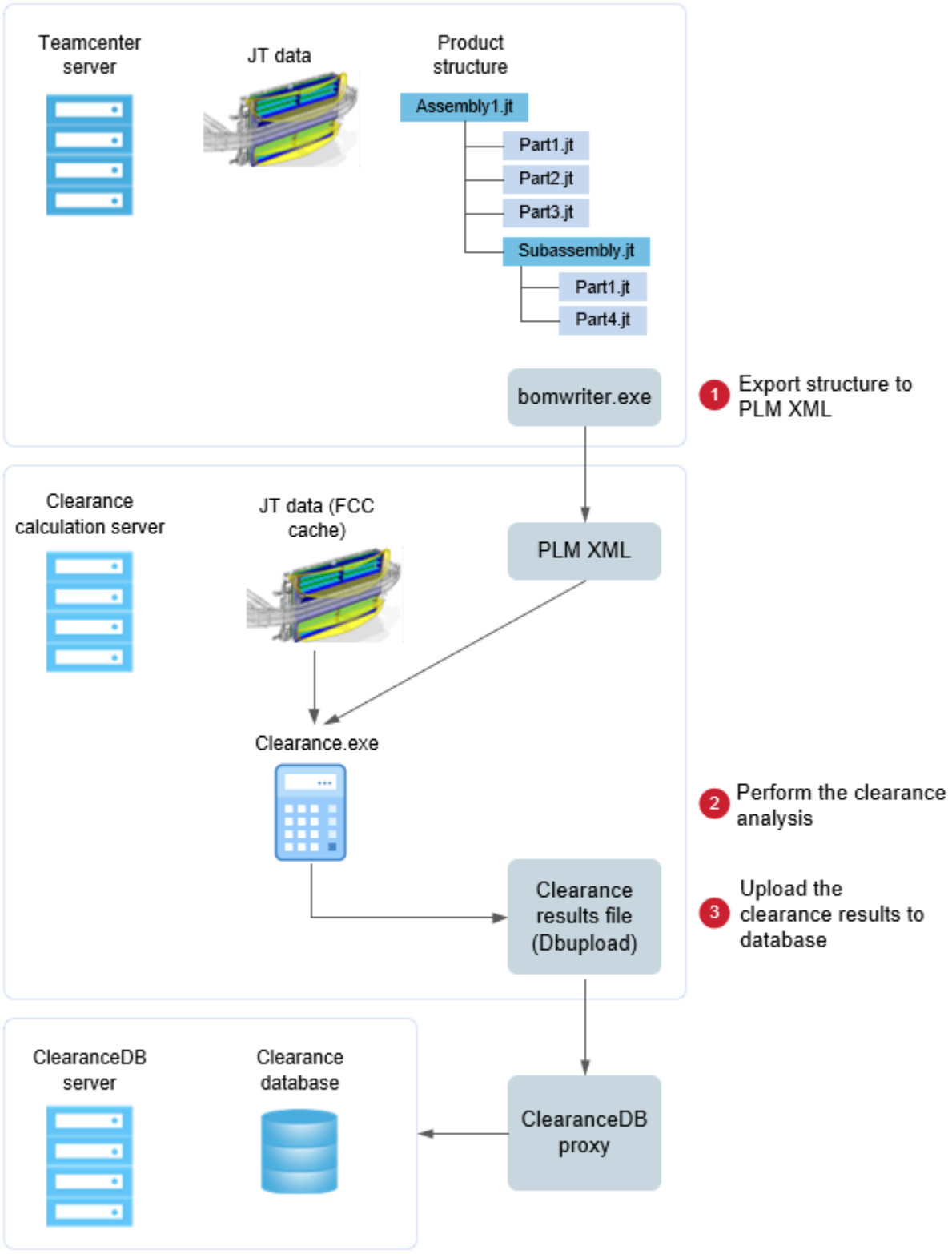
   At the command prompt, type

   ```
   analyze_product.pl
   ```

   The product data is analyzed and a clearance results file is generated.

Set up end item batch calculation for managed products

Use the following procedure to set up clearance batch calculation on a managed Teamcenter structure where some subassembly parts (typically the lowest level element in the clearance analysis) are overridden by an **end item** (**ElementType**=**EndItem**). End items are branches of a product structure where the root node contains the metadata **__PLM_END_ITEM=True**. All parts underneath an end item node are considered to be part of the same monolithic element.



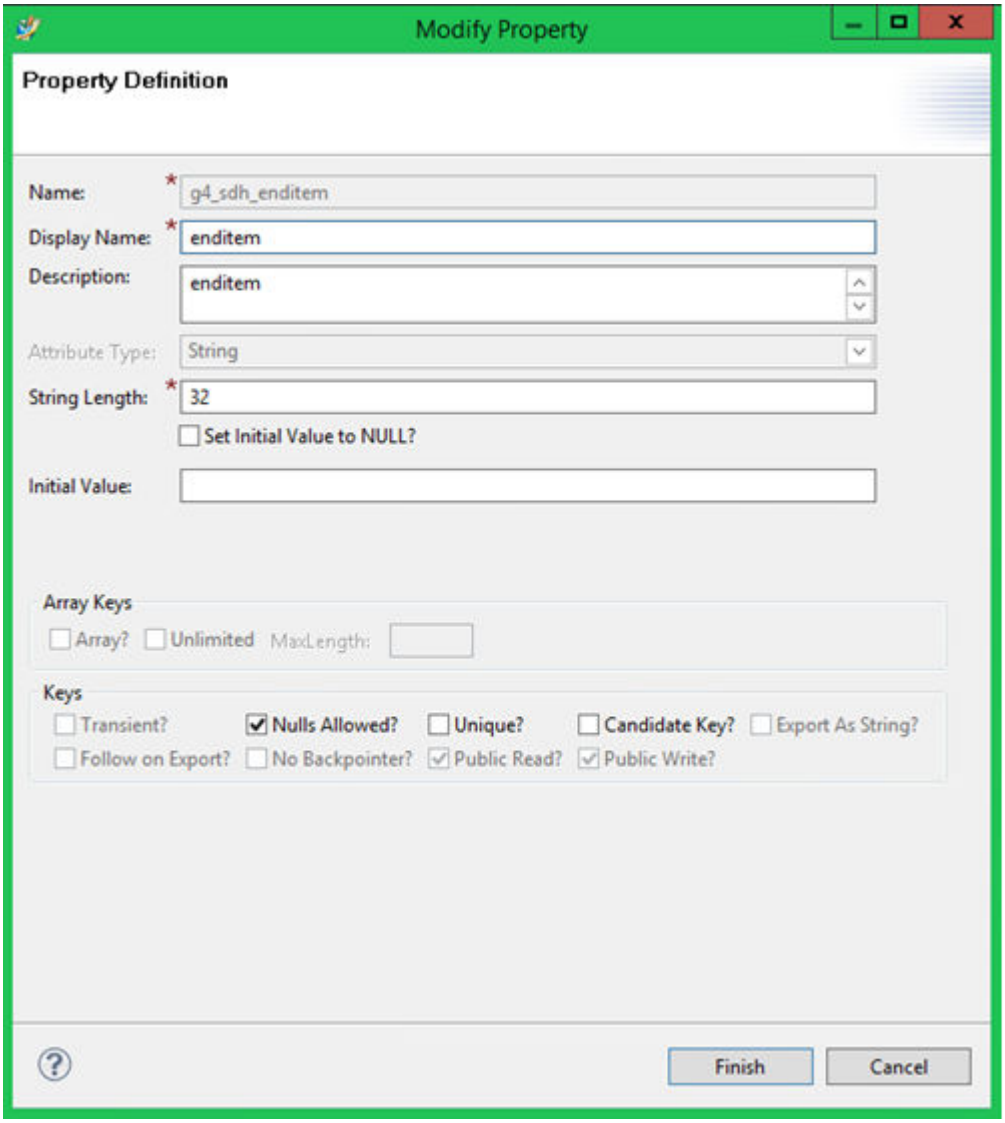
1. In the **Clearance.cfgglobal** file (....\ClearanceDB_Work_Area\Clearance.cfgglobal), configure the calculation to run on an end item by defining the **ElementType** as **EndItem**.

To configure a single product as an end item, configure the **Clearance.cfgproduct** file for the specific product.

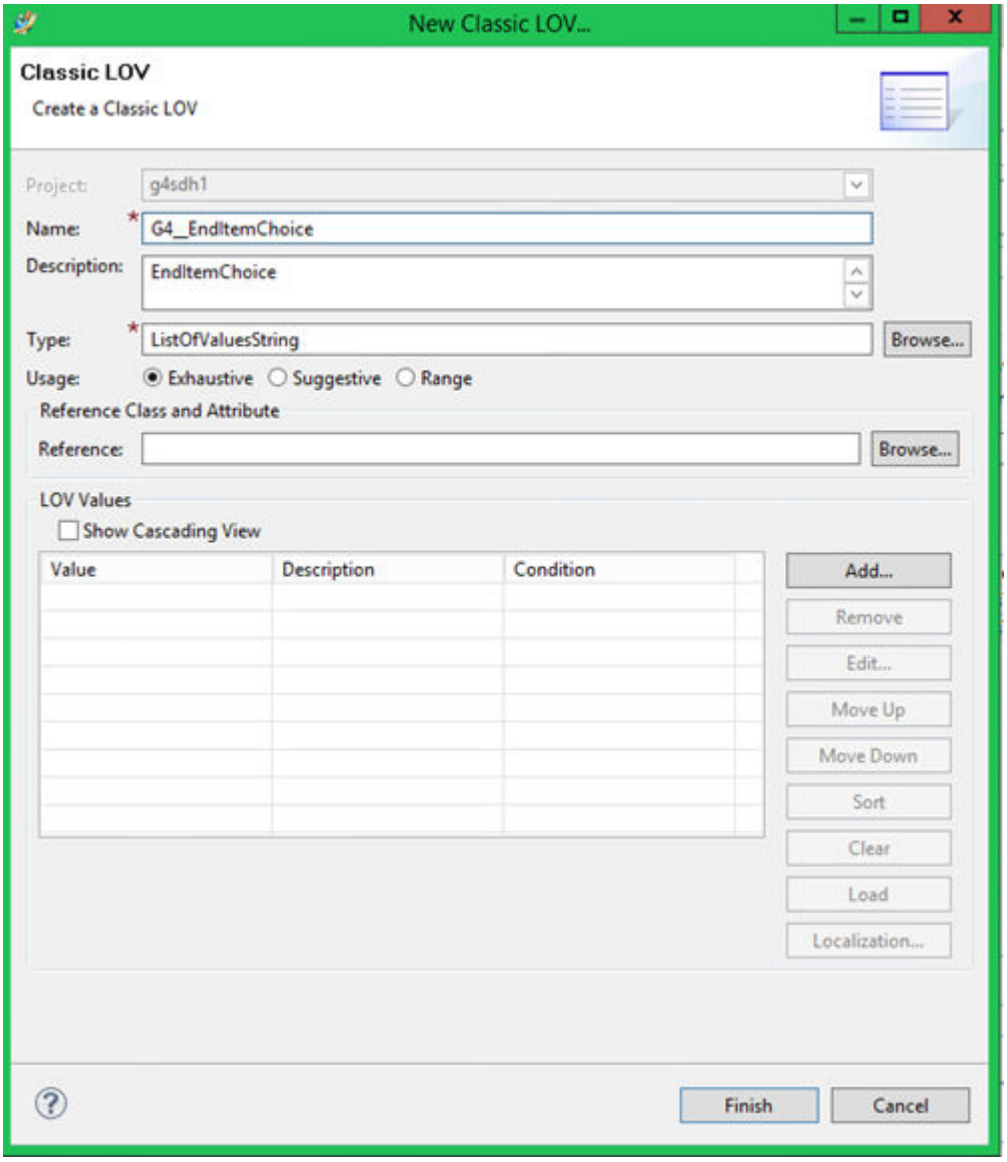
```
# 'ElementType' is used to set up the clearance element type, it must be one
# of the following options: Part, Component, EndItem, EndItemComponent
# default - Part

ElementType = EndItem
```

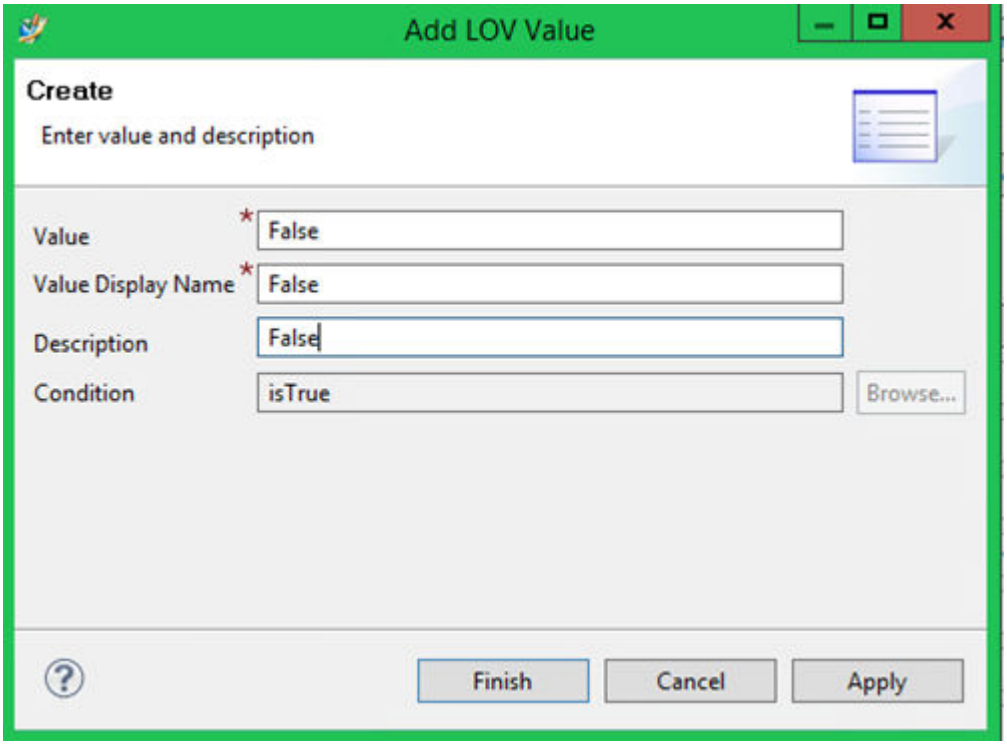
2. In BMIDE, add a new persistent property to the Item Revision.

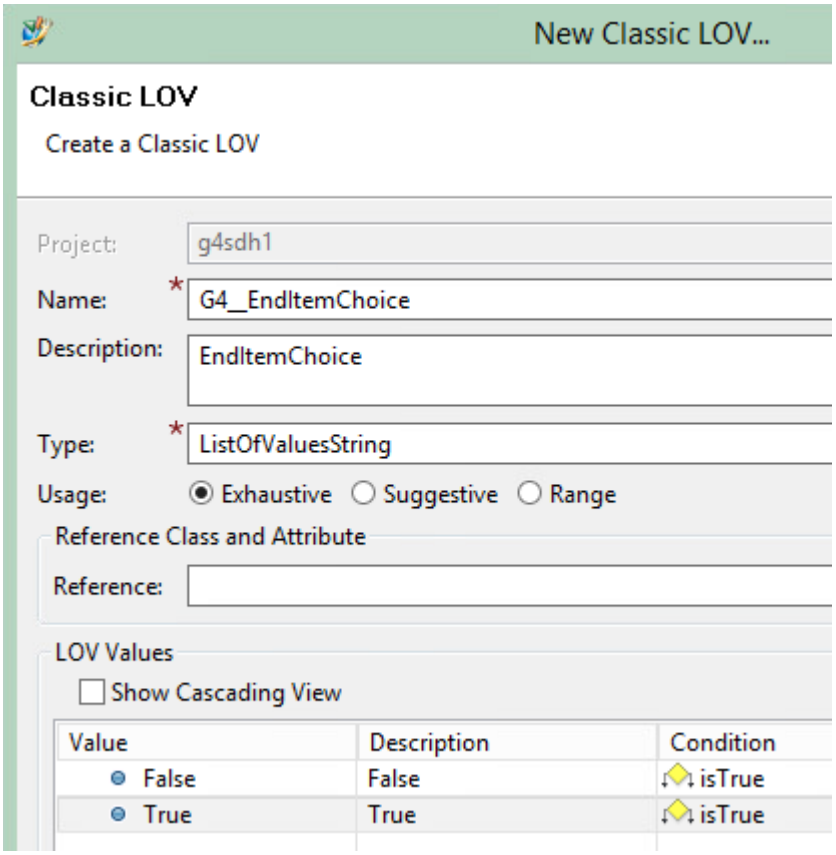


3. Create a new Classic List of Value (LOV) and attach the LOV to the property of the Item Revision created in the previous step.



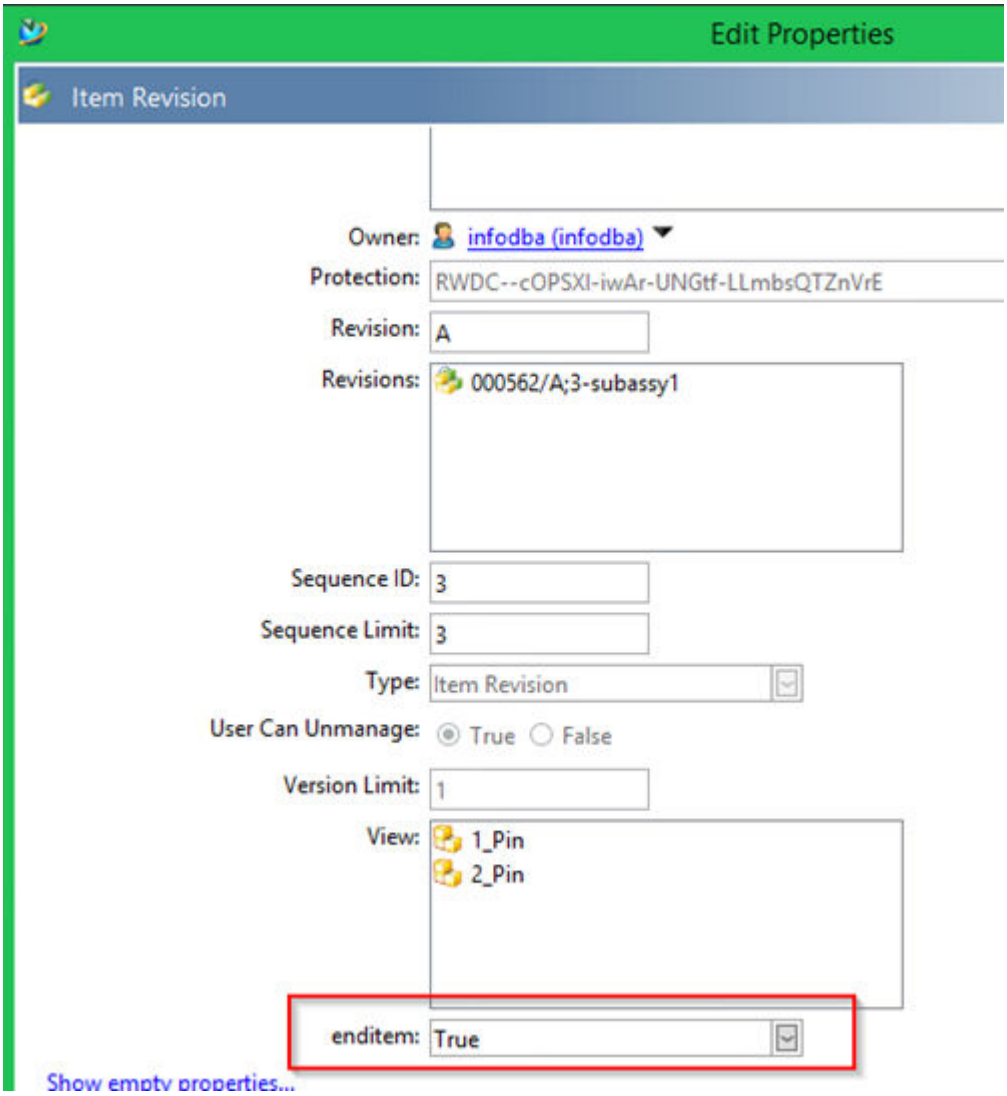
4. Add the True and False values to the LOV.



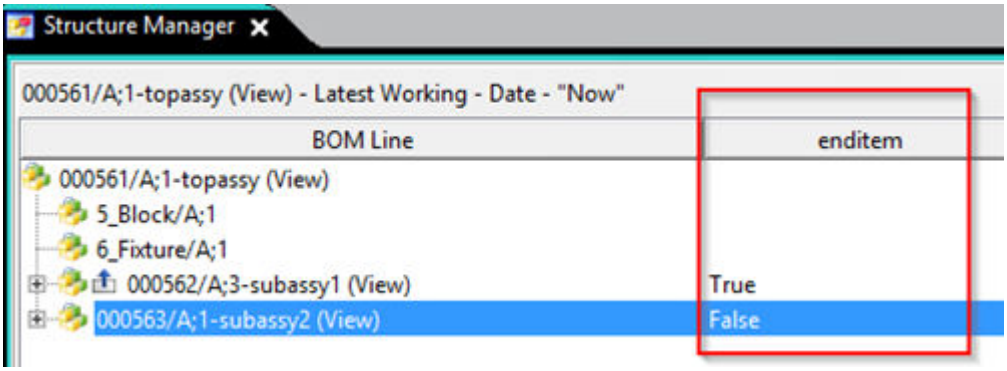


5. Attach the LOV to the Item Revision.

6. In Lifecycle Visualization, edit the properties of the Item Revision by setting the **enditem** to **True**.



This can also be done in Teamcenter Structure Manager.



7. Set **__PLM_END_ITEM=True** in the Clearance.cfgglobal file by doing the following.

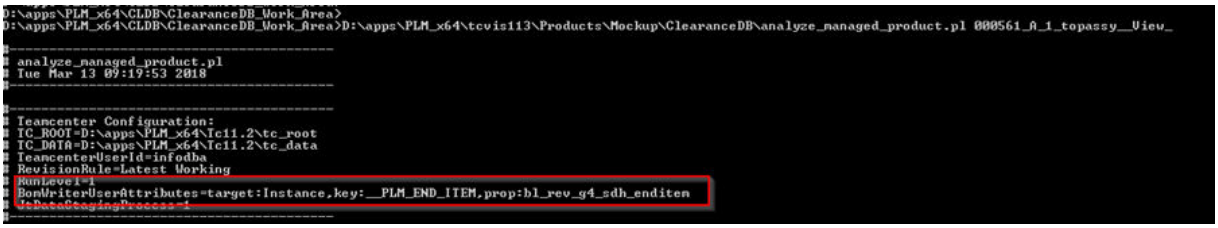
a. Add an additional user attribute using BOMwriter.

```
# 'BomWriterUserAttributes' is an optional Argument. User can specify the
# additional attributes in the following format.
# e.g target:Root,key:myAttribute,literal:\"My Attribute Value\"
# This is the same option of "-user_attributes=" as in 'bomwriter' command line
# options.

BomWriterUserAttributes=target:Instance,key:__PLM_END_ITEM,prop:bl_rev_g4_sdh_enditem
```

b. From the BOMwriter command prompt, type:

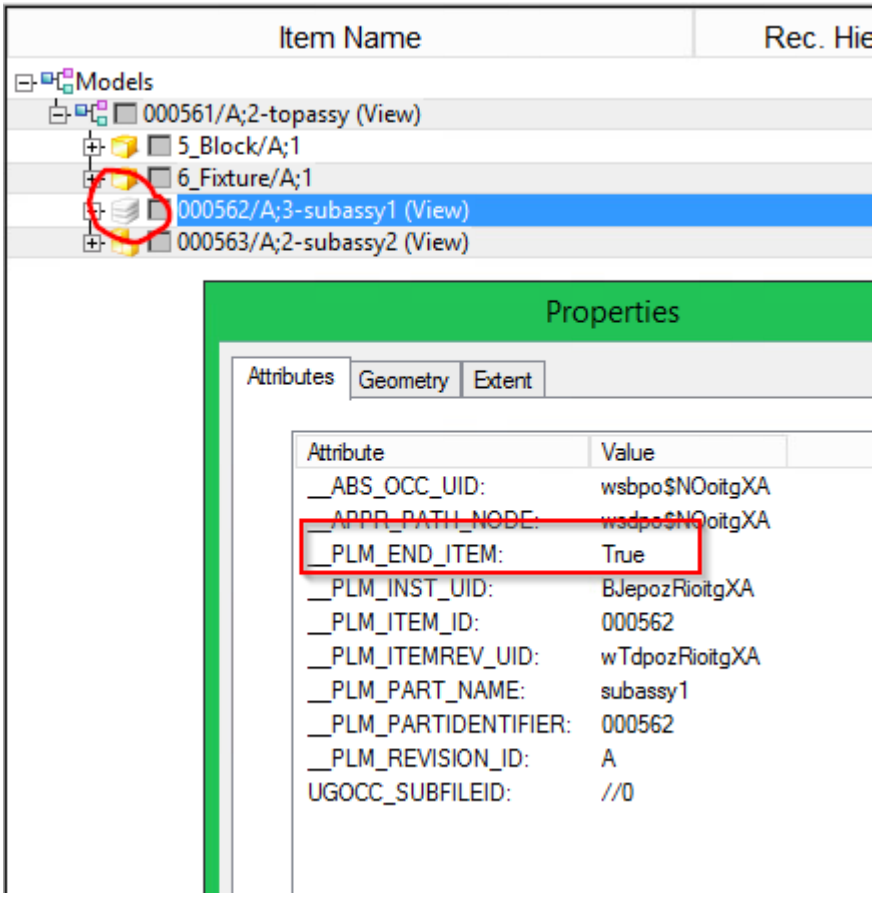
```
: analyze_managed_product.pl <product name>
```



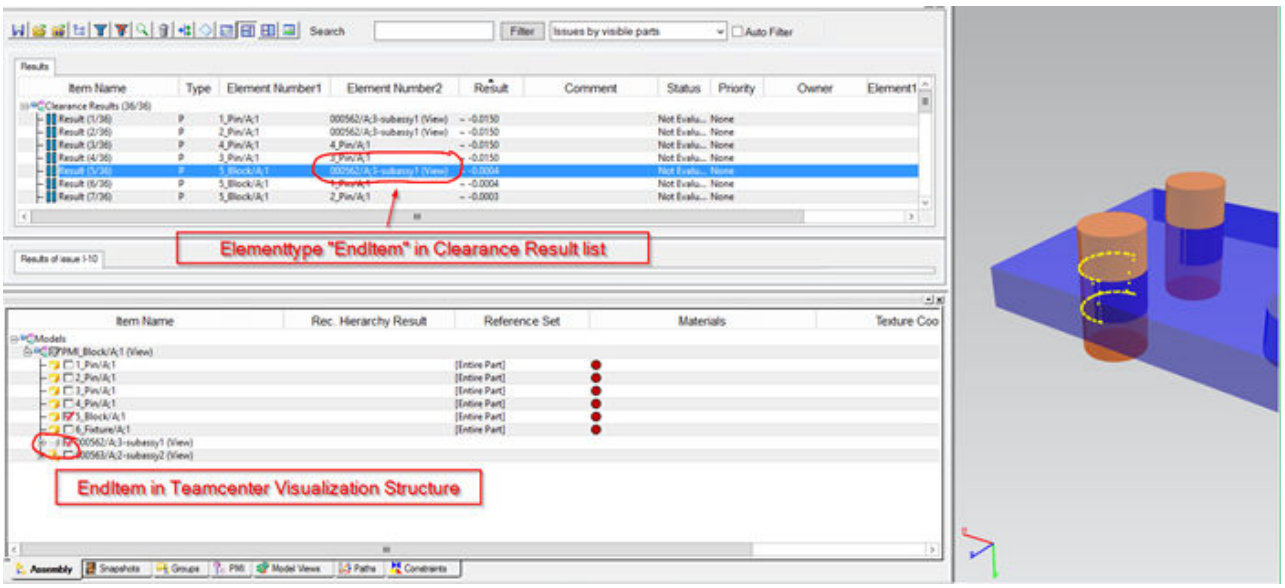
The BOMwriter utility maps the specified property to the Attribute in the .plmxml file.

```
<UserData id="id35">
<UserValue title="__PLM_INST_UID" value="BJepozRioitgXA"></UserValue>
<UserValue title="__PLM_INST_TH_UID" value="BNQpozRioitgXA"></UserValue>
<UserValue title="__PLM_END_ITEM" value="True"></UserValue></UserData>
<Transform id="id36">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1</Transform></ProductInstance>
```

c. To verify that the __PLM_END_ITEM property was set to True, open the created .plmxml file in Lifecycle Visualization.



8. Load the batch calculation result into Lifecycle Visualization where you see the end item in the structure, as well as in the Clearance Results list.



Perform clearance analysis using Clearance.exe

Perform a clearance check from the command line, and save your results as a comma-delimited, quote-enclosed, ASCII text file. (You can also use a **Perl script** to call Clearance.exe.)

From the command prompt, navigate to the directory containing the clearance executable (*Clearance.exe* on Windows and ClearanceExe on Linux), and type:

```
Clearance <input_filename> [options] -f <output filename>
```

where

This option	Does this
-d <value>	Specifies the clearance requirement, 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 email messages to owners, as specified in your Clearance Manager preferences.
-s	Shows only error messages as *Clearance.exe* runs.
-i	Creates images as *Clearance.exe* runs.

Perl scripts for batch clearance analysis

Use the following scripts to manage your products.

Use this script	To do this
analyze_managed_product.pl	Run clearance analysis on a managed product. **Usage:** **analyze_managed_product.pl [<product>]** **analyze_managed_product.pl -h** **Optional Arguments:** **<product>**: Product name. Not required if executing within a product directory. **-h**: Display help message. This script executes two operations in sequence, BOMwriter and Clearance. The BOMwriter generates a .plmxml file based on the

Use this script	To do this
	file supplied by the top level assembly itemId or itemKey, revision, and revision_rule. Clearance uses the .plmxml file as input, and runs clearance analysis on the whole assembly.
analyze_product.pl	Execute batch clearance analysis. **Usage:** **analyze_product.pl [<product>]** **analyze_product.pl -h** **Optional Arguments:** **<product>**: Product name. Not required if executing within a product directory. **-h**: Display help message.
copy_product.pl	Copy a product in the Clearance Work Area. • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **copy_product.pl <from product> <to product>** **copy_product.pl -h** **Required Arguments:** **<from product>**: Product name to copy. **<to product>**: Name of the new product. Illegal characters are replaced by \"_\". **Optional Arguments:** **-h**: Display help message.
create_ClearanceDB_working_dir.pl	Create the \"Clearance Work Area\". • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **create_ClearanceDB_working_dir.pl <path> [-df] [-lo]** **create_ClearanceDB_working_dir.pl -h** **Required Arguments:**

Use this script	To do this
	<path>: An absolute path where the Clearance Work Area will be created. **Optional Arguments:** **-df**: Create the default product in the new Clearance Work Area. **-lo**: Create a local (no database connection) Clearance Work Area. **-h**: Display help message.
create_default_product.pl	Create the \"DEFAULT\" product in the specified directory. **Usage:** **create_default_product.pl <path>** **create_default_product.pl -h** **Required Arguments:** **<path>**: An absolute path where the Clearance Work Area will be created. **Optional Arguments:** **-h**: Display help message.
create_images.pl	Creates 2D images of clearance results. **Optional Arguments:** **-h**: Display help message.
create_product.pl	Create a product in the Clearance Work Area. • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **create_product.pl <product> [-uc|-ur|-uz|-ua] [-ds <source> -cd <conn>]** **create_product.pl -h** **Required Arguments:** **<product>**: Product name. First character must be alphanumeric (0-9, A-Z, a-z). Illegal characters are replaced by \"_\". Enclose <product> in double quotes (\"\") if it contains spaces. **Optional Arguments:**

Use this script	To do this
	-uc: Copy configuration from the default product to the new product. **-ur**: Copy rules and conditions from the default product to the new product. **-uz**: Copy zones from the default product to the new product. **-ua**: Copy all the above from the default product to the new product. **-ds <data source> -cd <connect data>**: Create a local .dbc file. **<source>**: <proxy host machine name>:<port> **<conn>**: Proxy CONNECT_DATA. **-h**: Display help message.
delete_product.pl	Delete a product from Clearance Work Area. • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **delete_product.pl <product>** **delete_product.pl -h** **Required Arguments:** **<product>**: Product name to delete. **Optional Arguments:** **-h**: Display help message.
list_all_products.pl	List all products. • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **list_all_products.pl [<file>] | [<host> <port> <connect data>]** **list_all_products.pl -h** **Optional Arguments:**

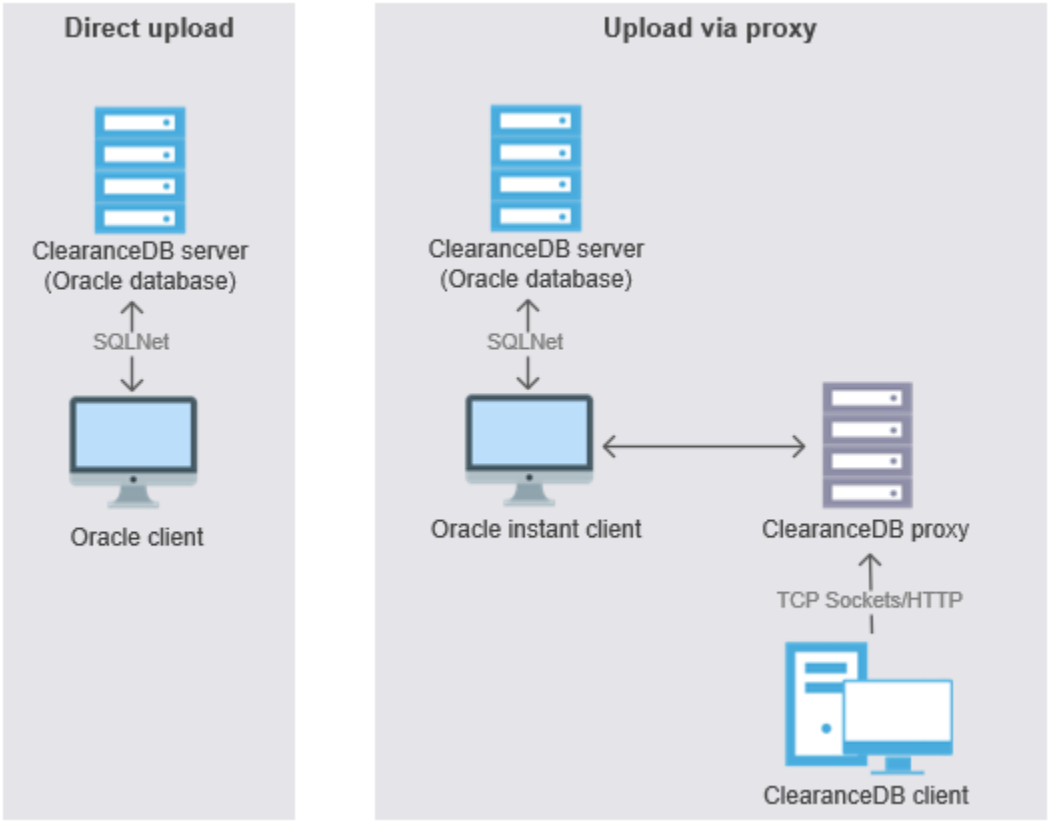
Use this script	To do this
	<file>: .dbc file. <file> can include an absolute path. If <file> is not passed, then the .dbc file in the current directory is used. **<host> <port> <connect data:**: **<host>**: Machine name where the proxy is running. **<port>**: Port number to use. **<connect data>**: Name of the proxy CONNECT_DATA. **-h**: Display help message.
list_product.pl	Report the current state of a product in the database (#issues, #parts, and last update). **Usage:** **list_product.pl [<product>]** **list_product.pl -h** **Optional Arguments:** **<product>**: Product name. Not required if executing within a product directory. **-h**: Display help message.
rename_product.pl	Rename a product in the Clearance Work Area. • Clearance_Work_Area: Local disk only. • ClearanceDB_Work_Area: Local disk and database. **Usage:** **rename_product.pl <from product> <to product>** **rename_product.pl -h** **Required Arguments:** **<from product>:** Product's old name. **<to product>**: Product's new name. Illegal characters are replaced by \"_\", for the product's new local disk directory name. **Optional Arguments:** **-h**: Display help message.

5. Updating products in the database

Updating products in the database

Product data, including configurations, rules, conditions, and zones, must be uploaded to 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 uploading product data to 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 upload data to the ClearanceDB database. You can specify one of the following:

Proxy — Perform updates using the ClearanceDB Proxy.

> Note:
>
> If **WEB_SERVICE=True**, the script will use the HTTP protocol. Product updates via HTTP protocol are supported only via the DBC file configurations.

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**.

4. Save the file.

Upload product data to the database

You can upload your product data, including configurations, rules, conditions, zones, images, and results, to the database.

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

2. Do any of the following:

To upload product attributes	Do this
From the command prompt	Type `update_product.pl [<product>] -uc:-ur:-ud:-uz:-ua [-df]` where `product` — is the product name. 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. –ur — The argument to upload the rules and conditions. –ud — The argument to upload the results. –uz — The argument to upload the zones. –ua — The argument to upload all of the product data, including the configuration, rules and conditions, results, and zones.

To upload product attributes	Do this
	–`df` — The argument to upload the default configuration of the product. –`ui` — The argument to upload clearance images. Example: `update_product.pl -ua`
From the shortcut menu	In the directory, right-click any of the following files and choose **Upload to ClearanceDB**: • *configuration.csvcldb* — ClearanceDB process configuration file • *rules.csvcldb* — ClearanceDB rules definition file • *conditions.csvcldb* — ClearanceDB conditions definition file • *zones.csvcldb* — ClearanceDB zones definition file • *ClearanceResultsDbUpload.csvcldb* — ClearanceDB results file

Your data is uploaded to the database and log files are created and saved to the product directory.

Upload data using the ClearanceDbClient

You can use the ClearanceDbClient application to upload product configurations, rules, conditions, zones, and analysis results to the database.

1. Open a command line window and navigate to the **...\Siemens\Teamcenter {current version}\Visualization\Products\Mockup** directory.

2. At the command prompt, type:

```
ClearanceDbClient -[file type] [file path] -q [DBC file path] -l [log file path]
```

This parameter	Represents
file type	The option for the file type you want to upload to the database.

<table>
<tr><th>This parameter</th><th>Represents</th></tr>
<tr><td></td><td>
<table>
<tr><th>Use this value</th><th>To upload these types of files</th></tr>
<tr><td>-uc</td><td>Configuration files (Configuration.csv)</td></tr>
<tr><td>-ur</td><td>Rules and conditions (Rules.csv and Conditions.csv)</td></tr>
<tr><td>-uz</td><td>Zones (Zones.csv)</td></tr>
<tr><td>-ud</td><td>Results (ClearanceResultsDbUpload.dat)</td></tr>
</table>
</td></tr>
<tr><td>file path</td><td>The path to the .csv file containing the data you want to upload.

Note:
• The file path must be enclosed in quotation marks.
• The rules option requires two .csv file paths, the rules .csv file path followed by the conditions .csv file path.

Example:
<pre>ClearanceDBClient -ur "C:\Rules.csv" "C:\Conditions.csv" -q "C:\sfdver08.dbc" -l "C:\log.txt"</pre></td></tr>
<tr><td>-q</td><td>The DBC file option.</td></tr>
<tr><td>DBC file path</td><td>The path to a properly constructed DBC file.

Note:
The file path must be enclosed in quotation marks.</td></tr>
<tr><td>-l</td><td>The log file option.</td></tr>
<tr><td>log file path</td><td>The path where you want to save the log file.

Note:
The file path must be enclosed in quotation marks.</td></tr>
</table>

Example:

```
ClearanceDBClient -uc "C:\Configuration.csv" -q "C:\sfdver08.dbc" -l
"C:\log.txt"
```

Merging results with the database

For visualization clients to access analysis results, you must upload and merge the *ClearanceResultsDbUpload.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**

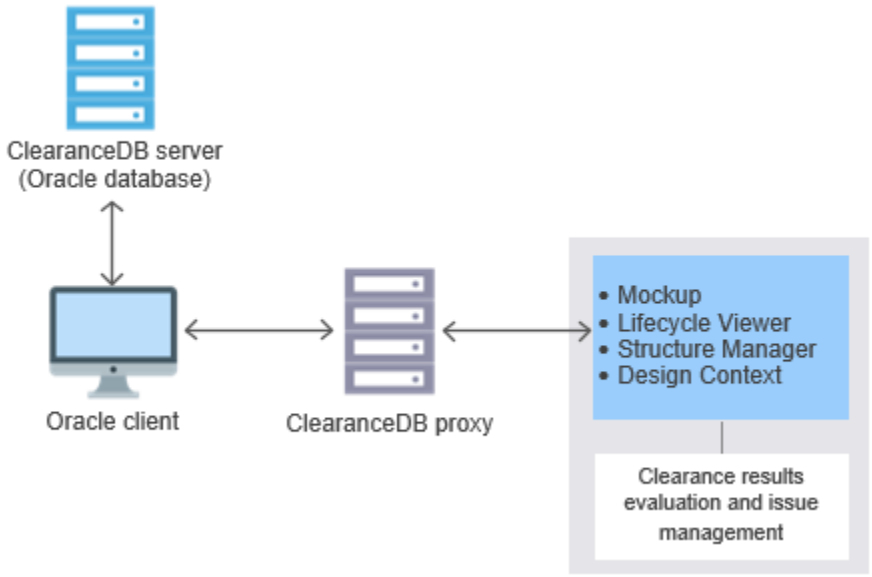
From the command prompt	Type
	`update_product.pl [<product>] [-ud] [-df]`
	Example:
	`update_product.pl fishing_reel -ud`

From the shortcut menu	In the *ClearanceDB_Work_Area\ Products\product_name* directory, right-click the *ClearanceResultsDbUpload.csvcldb* file and choose **Upload to ClearanceDB**.

6. Using Mockup to view and manage Clearance results

Viewing and managing results overview

ClearanceDB results evaluation and issue management occurs within the viewer. Supported visualization clients include standalone 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 **Tools→Clearance**.
2. Choose **Clearance→Preferences→Requirement Components**.
3. In the **Requirement Rule Components** dialog box, select **Requirement Rules from ClearanceDB Server**.
4. Choose **Clearance→Preferences→Result Components**.
5. In the **Result Components** dialog box, select one of these options:

Use	To enable this
General Clearance Results	Visual representation of results within the application. Select this option if you only want to view and manage ClearanceDB results from the database.
ClearanceDB Results	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. 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 To use HTTP-based communication protocol, the following additional environment variables must be set: `EAI_CL_WEB_SERVICE` — Enable HTTP-based Communication protocol for ClearanceDB. `EAI_CL_CLEARANCE_APPID` — Application id defined in the TcSS Application Registry. `EAI_CL_SSO_SERVICE_LOGIN_URL` — URL used by the TcSS solution for single sign-on. `EAI_CL_WEB_SERVICE_URL` — HTTP or HTTPS web service URL used by ClearanceDB.

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 **Clearance→Results→Load**.

   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.

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.

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**.

7. 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 **uploaded to the database**.

Use the following attributes to define ClearanceDB product configurations:

Use this attribute	To do this
EAICL_RELEVANT_PART_ATTRIBUTE	Store part metadata essential to the clearance management process in the database.
EAICL_GEOMETRY_CHANGE_PART_ATTRIBUTE	Use part metadata to indicate model design changes.
EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT	Turn on the initial clearance owner assignment.

Use this attribute	To do this
EAICL_INITIAL_CLEARANCE_OWNER_CHOICE	Specify the metadata used for the initial clearance owner assignment.
INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION	Specify the metadata used to determine the initial clearance owner assignment.
INITIAL_CLEARANCE_OWNER_SEARCH_METHOD	Specify how product structure is searched for the metadata used for the initial clearance owner assignment.
EAICL_INITIAL_CLEARANCE_PRIORITY	Specify an initial clearance issue priority value.
EAICL_INITIAL_CLEARANCE_STATUS	Specify an initial clearance issue status value.
EAICL_INITIAL_CLEARANCE_COMMENT	Specify an initial clearance issue comment value.
EAICL_INITIAL_CLEARANCE_IMPORTANT	Specify an initial clearance issue important value.
EAICL_INITIAL_CLEARANCE_CHANGE_REQUIRED	Specify an initial clearance issue change required value.
EAICL_INITIAL_CLEARANCE_CHANGE_REFERENCE	Specify an initial clearance issue change reference value.
EAICL_CLEARANCE_CHANGE_DETECTION	Specify the change detection method.
EAICL_CLEARANCE_DISPOSITION_PERSISTENCE	Specify how you want to persist user dispositions of clearance owner, status, priority, comment and zone.
EAICL_PART_MATCHING_METHOD	Specify the part reconciliation method.
EAICL_PART_NAME_PROPER	Specify the part property which identifies metadata uniquely.
EAICL_RULES_PART_ATTRIBUTE	Turn on incremental clearance analysis.
EAICL_OCCURRENCE_KEY	Turn on part occurrence persistence by specified metadata.
EAICL_CHANGE_HISTORY_COLUMN	Specify the information to include in Issue Change History reports.
EAICL_PRESERVE_RESOLVED_ISSUES	Specify to track a history of ClearanceDB issues.
EAICL_PRODUCT_HISTORY_MAX_ENTRIES	Specify the maximum number of ClearanceDB history issues.
EAICL_PRODUCT_HISTORY_RETENTION_DAYS	Specify how long to store database events.
EAICL_PRODUCT_STRUCTURE_CHANGE_TEST	Track changes in the product structure size.

Use this attribute	To do this
EAICL_PRODUCT_STRUCTURE_CHANGE_REJECTION_PERCENT	Specify the relative change in the product structure, between the old and new product snapshot, above which the clearance result upload will be rejected.
EAICL_RULEBASE_CHANGE_TEST	Specify to track changes in the size of the clearance requirement rulebase.
EAICL_RULEBASE_CHANGE_REJECTION_PERCENT	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.
EAICL_TMP_OBJECT_CLEANUP_AGE_IN_DAYS	Clean up session-specific database objects older than a specified duration.
EAICL_IDENTICAL_MATCHING_ATTRIBUTE	When configuring issue inheritance, specify the criteria that reports issues as identical.
EAICL_SIMILAR_MATCHING_ATTRIBUTE	When configuring issue inheritance, specify the criteria that reports issues as similar.
EAICL_RESEMBLING_MATCHING_ATTRIBUTE	When configuring issue inheritance, specify the criteria that reports issues as resembling.
EAICL_MATCHING_CRITERIA_NUMERIC_PRECISION	When configuring issue inheritance, specify the tolerance for numeric criteria in determining whether values are matching.

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.

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 *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_CLEARANCE_COMMENT

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

EAICL_INITIAL_CLEARANCE_IMPORTANT

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

EAICL_INITIAL_CLEARANCE_CHANGE_REQUIRED

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

EAICL_INITIAL_CLEARANCE_CHANGE_REFERENCE

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

EAICL_CLEARANCE_CHANGE_DETECTION

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

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

<table>
<tr><th>Use this</th><th>To do this</th></tr>
<tr><td></td><td>
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>
<tr><td>EAICL_RESULT_CHANGE_ TOLERANCE</td><td>Clearance result. Positive number. Expressed in model units.</td></tr>
<tr><td>EAICL_RESULT_CLEARANCE_ TYPE_CHANGE</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>EAICL_RESULT_POSITION_ CHANGE_TOLERANCE</td><td>Either part has moved. Positive number. Expressed in model units.</td></tr>
<tr><td>EAICL_RESULT_ORIENTATION_ CHANGE_TOLERANCE</td><td>Either part has been rotated. Positive number. Expressed in degrees.</td></tr>
<tr><td>EAICL_RESULT_INTERSECTION_ VOLUME_CHANGE_TOLERANCE</td><td>Volume of the intersection between parts. Positive number. Expressed in model units cubed.</td></tr>
<tr><td>EAICL_RESULT_REQUIREMENT_ CHANGE_TOLERANCE</td><td>Change in requirement. Positive number. Expressed in model units.</td></tr>
<tr><td>EAICL_RESULT_VIOLATION_ CHANGE_TOLERANCE</td><td>Difference between requirement and result. Positive number. Expressed in model units.</td></tr>
</table>
</td></tr>
</table>

Use this	To do this
	Note: The default for all of these new tolerance parameters is “OFF”.

Note:

Synonyms for the values of the existing *EAICL_CLEARANCE_DISPOSITION_PERSISTENCE* parameter are supported in this release for the sake of clarity. Both the existing values and the new synonyms are supported as exact equivalents.

- **Preserve Issue on Change**

  New value. Synonym for the existing value eaiclCADID.

- **New Issue on Change**

  New value. Synonym for the existing value eaiclCHANGE.

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_OCCURENCE_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*.

Use this	To do this
eaiclCADID	Use CADIDs associated with your model to keep track of part occurrences in the ClearanceDB database.
eaiclNGID	Use NGIDs associated with your model to keep track of part occurrences in the ClearanceDB database. 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.

Use this	To do this
	• 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. • Once NGID-based product structure persistence and resolution is adopted for a product, you cannot change back to the product's original identifiers. • Product structure that does not resolve unambiguously is not committed to the database. Such products should use the **legacy spatial method** of product structure reconciliation. • For products configured to use NGID-based product structure encoding, ClearanceDB Server 2007.1 and later releases do not support earlier ClearanceDB client applications.
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.

EAICL_CHANGE_HISTORY_COLUMN

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

Note:

These values are case and space-sensitive.

- Comment
- Important
- Owner
- Priority
- Requirement

- Result
- Db Rule Descriptions
- Db Rule Names
- 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:

Use this	To do this
Yes	Resolved issues are preserved.
No	Resolved issues are not preserved. This setting is the default behavior.

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 *EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY* attribute to the configuration file.

Example:

```
ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Owner

   ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Priority
```

```
   ALL,EAICL_SIMILAR_ISSUE_ATTRIBUTE_COPY,Status
```

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

EAICL_RESEMBLING_MATCHING_ATTRIBUTE

Use the *EAICL_RESEMBLING_MATCHING_ATTRIBUTE* attribute in the **ALL** product to specify the criteria that reports issues as *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.

Example:

```
ALL,EAICL_RESEMBLING_MATCHING_ATTRIBUTE,Part Name

   ALL,EAICL_RESEMBLING_MATCHING_ATTRIBUTE,eaiclRESULT
```

Use the *EAICL_IDENTICAL_MATCHING_ATTRIBUTE* and *EAICL_SIMILAR_MATCHING_ATTRIBUTE* attributes to define *identical* and *similar* 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 resembling issues. To do this, add the *EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY* attribute to the configuration file.

Example:

```
ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Owner

   ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Priority

   ALL,EAICL_RESEMBLING_ISSUE_ATTRIBUTE_COPY,Status
```

Use the *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.

EAICL_RULE_CATEGORY

Use the *EAICL_RULE_CATEGORY* attribute to define a category name and category search token for the product rule results. You can have multiple instances of the *EAICL_RULE_CATEGORY* attribute in a *configuration.csvcldb* file, each with a different <categoryName;<catSearchToken> value.

Example:

```
jet_plane,EAICL_RULE_CATEGORY,WireCurrent;Current
jet_plane,EAICL_RULE_CATEGORY,HydrLines;Hydraulic
jet_plane,EAICL_RULE_CATEGORY,FuelLines;Fuel
```

Where jet_plane is the product name, and wireCurrent, HydrLines, and FuelLines are the category names, and current, hydraulic, and fuel are the category search tokens.

Upload product attributes to the database

In order to create a ClearanceDB product and associate product attributes with it, you must upload 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:

To upload product attributes	**Do this**
From the command prompt	Type `update_product.pl [<product>] [-uc] [-df]`

To upload product attributes	Do this
	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
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION,LastModified
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT,ON
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_OWNER_SEARCH_METHOD,1
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_OWNER_UNKNOWN,Unknown
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_PRIORITY,None
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_STATUS,Not Evaluated
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_IMPORTANT,Yes
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_CHANGE_REQUIRED,No
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_CHANGE_REFERENCE,
A-728AIRCRAFT,EAICL_INITIAL_CLEARANCE_COMMENT,
A-728AIRCRAFT,EAICL_PART_MATCHING_METHOD,eaiclSPATIAL
A-728AIRCRAFT,EAICL_RULES_PART_ATTRIBUTE,eaiclRULES
```

8. Issue inheritance

Inheritance overview

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 **Clearance Results** window, providing details about the selected occurrence.

4. Select other issues in the **Results** list to view the details for the occurrences.

Configuring the ALL product for issue inheritance

Set up 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 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_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_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:

```
exec eaicl_p_reset_gc_attr (eaicl_p_product.product_id
('<product name>'));
```

   From a command line:

```
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:

```
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.

9. Log files

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

This field	Records this information
`Timestamp`	The date and time of the operation.
`Event`	The name of the event. The following events are recognized: • `Pair added` — A part pair was added to the test bed (*TmpPairs.txt*). • `Loading pairs` — The test bed is being loaded for analysis. • `Pair loading complete` — All of the test bed is loaded for analysis. • `Result added` — A clearance result (incident) was added to the temporary result set (*TmpResults.txt*). • `Loading results` — The temporary result set is being translated to the *ClearanceResultsDbUpload.csvcldb* format. • `Results loading complete` — All of the temporary result set was translated to the *ClearanceResultsDbUpload.csvcldb* format. • `Failures loading` — The set of failures is being added to the result set when requested. • `Failure loading complete` — All of the failures were processed.

This field	Records this information
	• `Failure added` — A failure to arrive at a clearance result due to failure to calculate within a time limit or at all is recognized. • `All processing is complete` — Analysis is complete. • `Pair to be analyzed exactly the second time` — The second attempt at NURBS analysis. • `Pair to be analyzed approximately the second time` — The second attempt at facet based analysis. • `Pair to be analyzed approximately only` — The first attempt at facet based analysis after the NURBS analysis failed.
`Name1`	Determined by the `IdentifyPartOccurrencesWithMetadata` setting in the *Clearance.cfgglobal* file.
`Datastore1`	Determined by the `IdentifyPartsWithDatastores` setting in the *Clearance.cfgglobal* file.
`Name2`	Determined by the `IdentifyPartOccurrencesWithMetadata` setting in the *Clearance.cfgglobal* file.
`Datastore2`	Determined by the `IdentifyPartsWithDatastores` setting in the *Clearance.cfgglobal* file.

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:

This setting	Does this
`ProgressDirectory`	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.
`IdentifyPartsWithDatastores`	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`.
`MaxLogFileSize`	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

This setting	Does this
	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.
`RecordPartPairAdditionEvents`	Controls whether the test bed (part pair) creation events are recorded in the log file (`1` or `0`). The default is `1`.
`RecordClearanceResultAdditionEvents`	Controls whether the clearance results creation events are recorded in the log file (`1` or `0`). The default is `1`.

In addition to the *Log.txt* file, the following temporary files are created and deleted when you run the Clearance Calculator:

This file	Has this purpose
TmpLock.txt	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.
TmpFail.txt	Counts failures to arrive at the result of clearance analysis. This is a binary file.
TmpPairs.txt	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.
TmpResults.txt	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.
TmpHeader.txt	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: • `Field 1` — The name of the file relative to the directory the Clearance Calculator is started from. • `Field 2` — The timestamp of the start of the Clearance Calculator. • `Field 3` — The timestamp of the start of the Clearance Calculator, or the last re-start, whichever is later. • `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.

This file	Has this purpose
	• `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. • `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+URESULTS`

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.)

10. Rules

What are clearance rules?

What are clearance rules?

Prior to the introduction of the clearance rules editor, Mockup users could only perform clearance analyses using a single, global requirement (usually 0). With clearance rules and conditions, users have complete control over what parts are being analyzed, and what the requirements are between any pair of parts, or between any groups of parts. This functionality was previously only available to ClearanceDB users running Clearance in batch mode using *Clearance.exe*. Using the rules editor in Mockup, users are now able to visualize the scope of any rule or combination of rules, or execute any rule or combination of rules to see the results of that analysis.

Clearance conditions

Clearance conditions define the scope of data to which rules apply. You can use any metadata associated with your model to define the conditions for any rule.

Rules can have one or many conditions. Generally, as more conditions are applied to a rule, the scope of the rule becomes smaller, as will the results. Rules can also have no conditions at all, in which case the scope of the rule is the entire model.

Why are rules and conditions useful?

Clearance rules and conditions enable you to limit the analysis to only those areas that are desired, which in turn helps to optimize the performance of the analysis. Using the rules editor, you can load your existing rules and conditions data into the viewer for visualization and analysis, or you can author rules and conditions from scratch inside the viewer. When done, you can save your rules and conditions either to a file on the local file system as a .csvcldb file, or to the Clearance database (ClearanceDB).

Rules and conditions can give you additional control over the clearance analysis in situations such as the following:

- Limit the analysis to certain parts or assemblies only.
- Exclude parts from the analysis because they will always show up as false positives. For example, a rivet or a flexible rubber seal.
- Prevent parts within a particular subassembly from being analyzed because the assembly is provided by a supplier.
- Define a larger requirement between two groups of parts, such as between fuel lines and engine parts that might get very hot.

What if more than one rule applies to the same part pair?

You can create and apply multiple rules to conduct analyses according to arbitrarily complex design criteria. For part pairs that fall under the scope of multiple rules, Clearance conducts analyses according to the following:

- The largest clearance 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 part sets.
- If no rule applies to the part pair, it is not analyzed.

Using rules in clearance analysis

Use the following rule and condition examples to better understand the full scope of their potential in clearance analysis. Rules can be either inclusion (additive) or exclusion (subtractive).

Design Pattern	Definition
General non-interference	Defines a single, global requirement that applies to all parts in the model. There are no conditions, so the scope of the rule is not limited to any set of parts.
Single inclusion	Defines a single set of parts using a single condition, and a minimum distance between those parts and any other parts in the model.
Single condition (exclusion)	Defines a single set of parts using a single condition, then excludes those parts from the analysis.
Single condition (assembly)	Defines a single set of parts by using a single condition which applies to the ancestors of a part (in other words, a “branch” of the product structure) and a minimum distance between those parts and any other parts in the model.
Multiple conditions per rule	Defines a single set of parts using multiple conditions, and a minimum distance between those parts and any other parts in the model.
Left/right part sets	Defines 2 sets of parts (left/right), and a minimum distance between parts in the left set and parts in the right set.
Self inclusion	Defines a single set of parts (a condition), and a rule with left/right part sets that both share that same condition. This results in a rule that checks a part set against itself.
Self exclusion (end item)	Defines a single set of parts (a condition), and a rule with left/right part sets that both share that same condition. This excludes any issues between parts within that set.

Example: General non-interference rule

Pattern	General non-interference
Description	Defines a single, global requirement that applies to all parts in the model. There are no conditions, so the scope of the rule is not limited to any set of parts This is equivalent to a “matrix clearance” in Mockup with a requirement of zero.
Example	No parts can be closer than 0 mm to any other parts.
Requirement	0 mm
Conditions	None

Example: Single inclusion rule

Pattern	Single inclusion
Description	Defines a single set of parts using a single condition, and a minimum distance between those parts and any other parts in the model
Example	Cubes cannot be closer than 10 mm to any other parts in the model (including other cubes).
Requirement	10 mm
Conditions	Name = Cube

Example: Single condition (exclusion) rule

Pattern	Single condition (exclusion)
Description	Defines a single set of parts using a single condition, then excludes those parts from the analysis. Exclusion rules always require at least one inclusion rule (usually general non-interference) to define the scope of the analysis to be excluded from.
Example	Exclude all cubes from the analysis.
Requirement	None (exclusion)
Conditions	Name = Cube

Example: Single condition (assembly) rule

Pattern	Single condition (assembly)
Description	Defines a single set of parts by identifying an assembly node(s) in the product structure and a minimum distance between the children of those assemblies and any other parts in the model.
Example	Children of Level1a cannot be closer than 10 mm to any other parts in the model (including other children of Level1a).

Requirement	10 mm
Conditions	Name = Level1a
Search	Assemblies

Example: Multiple conditions per rule

Pattern	Multiple conditions
Description	Defines a single set of parts using multiple conditions, and a minimum distance between those parts and any other parts in the model. Multiple conditions on the same rule are conjoined by "And" (condition x AND condition y).
Example	Red spheres owned by Bob cannot be closer than 10 mm to any other parts in the model (including other red spheres owned by Bob).
Requirement	10 mm
Conditions	Color = red Name = Sphere Owner = Bob

Example: Left/right part sets rule

Pattern	Left/right part sets
Description	Defines 2 sets of parts (left and right), and a minimum distance between parts in the left set and parts in the right set.
Example	Green cubes cannot be closer than 10 mm to red spheres.
Requirement	10 mm
Conditions	Left: Color = green, Name = Cube Right: Color = red, Name = Sphere

Example: Self inclusion rule

Pattern	Self inclusion
Description	Defines a single set of parts by using a rule with left/right part sets that both share the same condition. This results in a rule that checks a part set against itself. This is equivalent to a matrix clearance where only the parts within the group of interest are in scope.
Example	Green cubes cannot be closer than 10 mm to other green cubes.
Requirement	10 mm
Conditions	Left: Color = green, Name = Cube

	Right: Color = green, Name = Cube

Example: Self exclusion rule

Pattern	Self exclusion (end item)
Description	Defines a single set of parts by using a rule with left/right part sets that both share the same condition. This results in a rule that checks a part set against itself, and then excludes any issues between parts within that set.
Example	Exclude all issues between green cubes.
Requirement	None (exclusion)
Conditions	Left: Color = green, Name = cube Right: Color = green, Name = cube

ClearanceDB rules and conditions details

Rules definition

ClearanceDB rules consist of the following:

- 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:

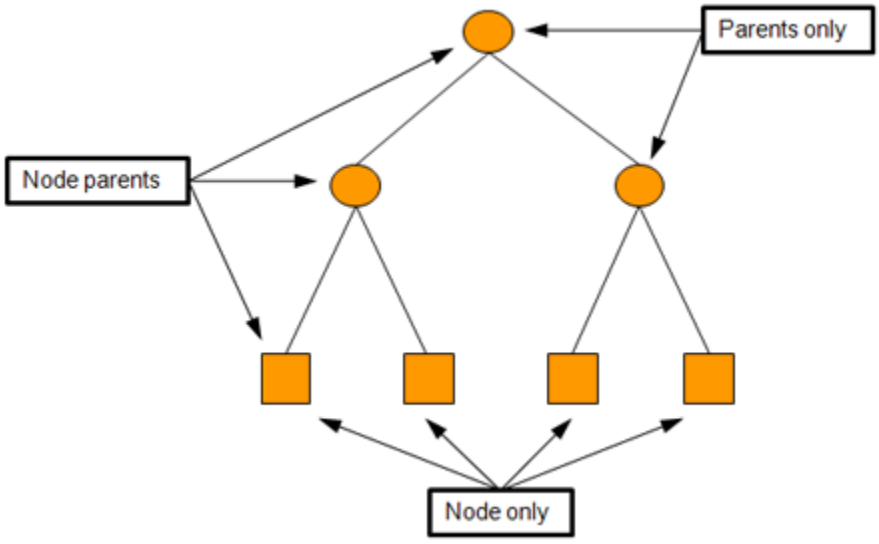
```
Rule Number,Product Name,Rule Description,Requirement
100,fishing_reel,General Non-Interference,0
```

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

```
Rule Number,Condition Description,Condition Domain,Search Type,
Match Type,Attribute Name,Attribute Pattern
100,Match Part Name,Left Node,Node parents,Y,eaiclNAME,spool_assembly%
```

Create ClearanceDB rule categories

ClearanceDB rules can be grouped into user-defined categories, allowing you to track rule result counts and corrective progress to rule violations using the **Rule Violation Counts** report.

1. Navigate to the *<product name> configuration.csvcldb* file in the product directory and open the file.

2. Add the attribute **EAICL_RULE_CATEGORY**, <categoryName>;<catSearchToken>, where <categoryName> is the user-defined name for the rule category and <catSearchToken> is the user-defined string that the system will search for within the rule description to determine if the rule belongs to the category.

   Results generated by a rule that contains the matching search token in its description are counted towards that rule's result count in the database. Rule result counts can be viewed in the Rule Violation Counts report.

   Note:

   A result generated by multiple different rules will be counted towards the rule result counts for each of those rules that is configured into a category.

Example:

```
jet_plane,EAICL_RULE_CATEGORY,wireCurrent;current
```

Where jet_plane is the product name, and wireCurrent is the category name, and current is the category search token.

3. Navigate to the *<product name> rules.csvcldb* file in the product directory and open the file.

4. Add the category search token value defined in step 2 to the rule description, surrounded by CAT(). This is the value that determines whether the analysis results are counted towards that rule's result count and saved in the Clearance database.

5. **Upload the rules** to the database.

Upload rules and conditions to the database

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

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

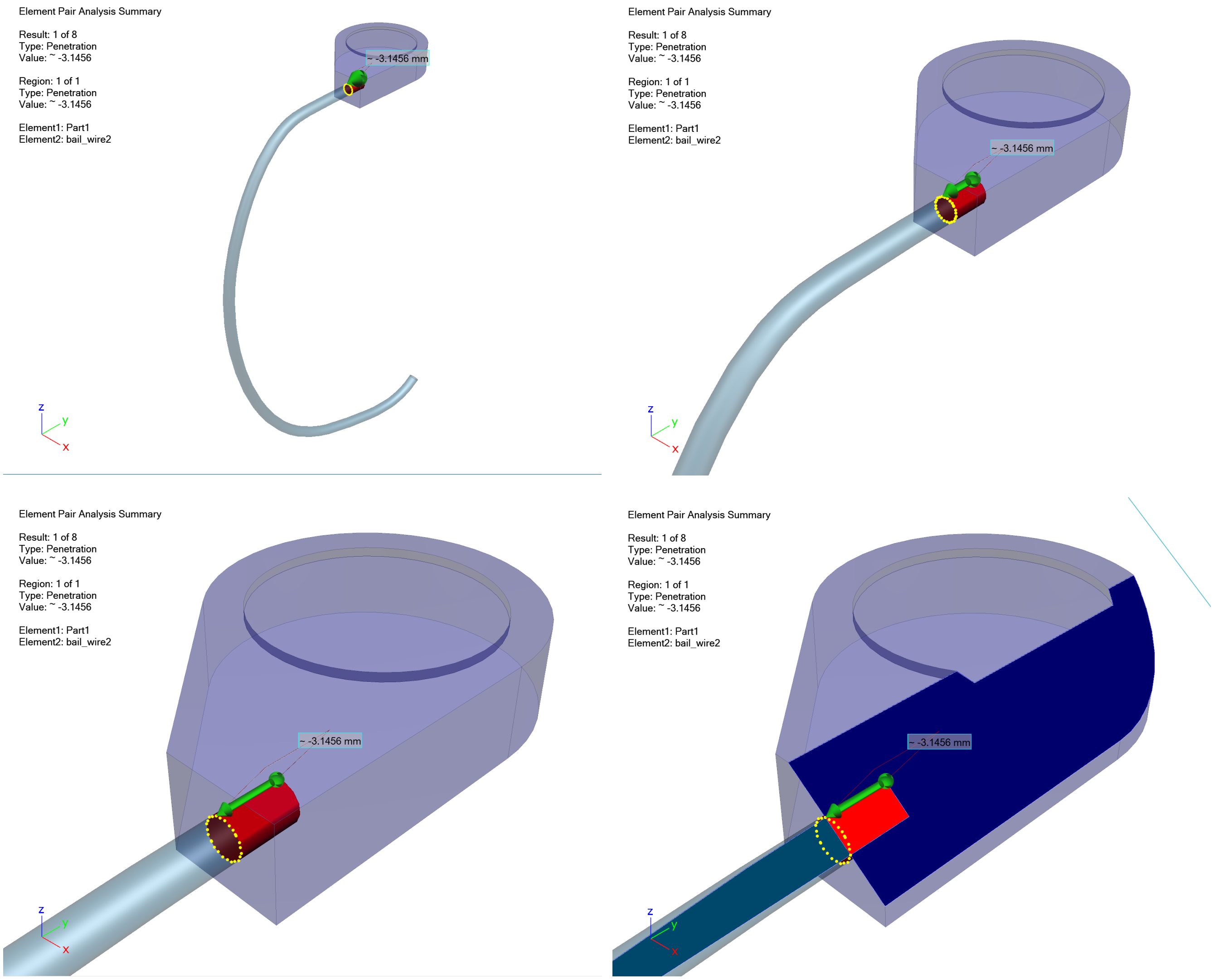
2. Do either of the following:

To upload rules and conditions	Do this
(Windows) From the shortcut menu	Right-click the *rules.csvcldb* file and choose **Upload to ClearanceDB**.
From the command prompt	Type `update_product.pl -ur`

11. Images

What are clearance images?

You can create and export 2D images during a clearance analysis using either or *Clearance.exe*. Mockup will generally create higher quality images, but that process can also take longer than creating the images through *Clearance.exe*. Also, using mockup you can create as many different images as desired for each clearance result, which you cannot do when using Clearance.exe. (*Clearance.exe* generates only one image per result.) Typically, ClearanceDB administrators develop scripts that create and **upload images** in batch as part of an automated process, but end-users can also create them using mockup.



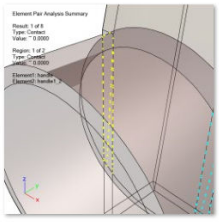
Note:

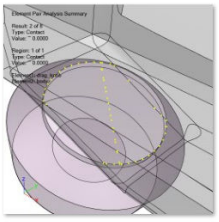
- Creating clearance images can be a very time-consuming process. You should be aware that additional computing resources may be necessary to create images within the desired time frame.
- For businesses that run their clearance analyses on Linux using *Clearance.exe*, be aware that the higher-quality images created in Teamcenter lifecycle visualization mockup is available on Windows only.

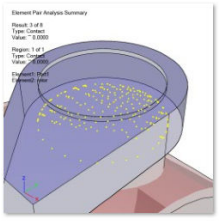
Create images in batch using Mockup

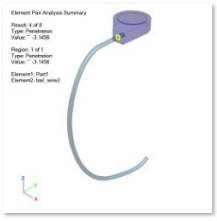
You can create 2D images of clearance analysis results in Mockup either or in batch. To create images in batch, set up a **ClearanceDB Work Area** and then run the create_images.pl script on the products of interest. You can then upload the images to the clearance database, if desired. You can then manually add the images to your products.

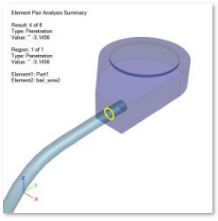
Before you can run the create_images.pl script, you must create a product directory with a modified *Clearance.cfgproduct* (configuration file) and turn on General Clearance Results in the *Clearance.cfgglobal* file. Once complete, you will have an image folder that looks similar to the image below. You can then run the *update_product.pl* script to upload the images to the Clearance database.

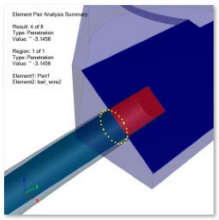

1_C_1.jpg

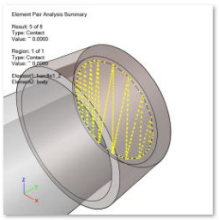

2_C_1.jpg

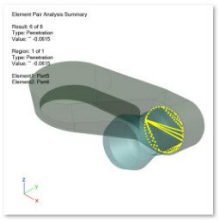

3_C_1.jpg

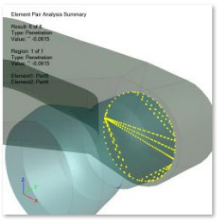

4_P_1.jpg

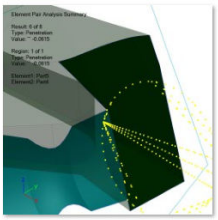

4_P_2.jpg

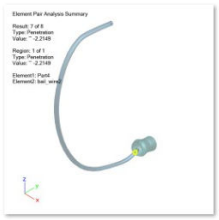

4_P_3.jpg

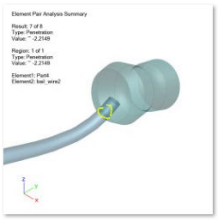

5_C_1.jpg

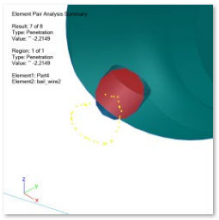

6_P_1.jpg

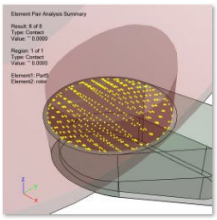

6_P_2.jpg


6_P_3.jpg


7_P_1.jpg


7_P_2.jpg


7_P_3.jpg


8_C_1.jpg

The supplied Siemens scripts operate on a single product only. You must create (or edit) your own script to iterate through your list of products and process each one. The supplied scripts include *list_all_products.pl*, *analyze_product.pl*, and *analyze_managed_product.pl*, and by default are located in *C:\Program Files\Siemens\Teamcenterx.x\Visualization\Products\Mockup\ClearanceDB*.

This option	Does this
<product>	Identifies the name of the product for which the images will be created. This value is not required if executing within a product library.
-h	Displays the help message.

1. (Optional) Activate multi-image (per clearance result) export in Mockup.

   a. Launch Mockup and load any model.

   b. On the **3D Clearance** toolbar, click **Clearance**  .

   c. On the **3D Clearance** toolbar, click **Results Preferences**  .

   d. In the **Clearance Results Preferences** dialog box on the **General** page, click **Generate images during clearance analysis** and **Create multiple images per result**.

   e. Click **Define Images** and define your image settings:

To	Do this
Add a camera definition.	Click **Add**. A new camera definition row is added to the table. Edit the definition to customize the camera settings.
Edit a camera definition.	A. Select the desired camera row from the table. B. Click **Edit**. C. In the **Edit Image Settings** dialog box, use the dropdown lists to define the camera **Zoom %**, **Camera** angle, and **Section** values for this camera definition. D. (Optional) Click the right or left arrows to navigate to the next camera definition or to the previous camera definition. E. Click **OK**.
Delete a camera definition.	A. Select the desired camera row from the table. B. Click **Delete**.

f. In the **Define Clearance Images** dialog box, click OK.

g. Close Mockup.

2. At a command prompt, create a **ClearanceDB Work Area**.

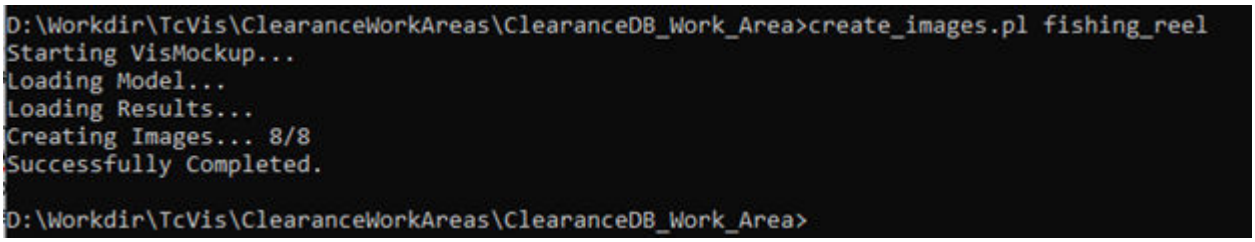
3. Create the desired product folder in your work area, if you haven't already done so.

   a. Change to the ClearanceDB_Work_Area directory.

   b. **Create a product**, or use an existing product.

4. Run the *create_images.pl* script. If you run the script from the work area folder, you must include the product name.

   The images are placed in the *Images* folder in the product directory and you will see an output similar to the image below.



5. Run the *update_product.pl* script to upload the images to the Clearance database.

To create images for all products, create or edit your script that processes the products in your ClearanceDB_Work_Area.

For each product in your product list, add the following line:

```
create_images(); #create the images
```

Example:

```
#get the list of products to process (customer script)
product_list=get_product_list();

for (each product in product_list)
      analyze_product(); #analyze the product

for (each product in product_list)
      create_images(); #create the images
```

Tip:

Siemens Digital Industries Software recommends that you use separate **for** loops for each product, as shown in the example, so that all products are calculated and uploaded *before* images are created and uploaded. Using a single **for** loop processes and uploads images for the current product before the next product calculation begins, which could significantly slow down your product calculations.

12. Server-side filters

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.

> Example:
>
> 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 upload the results to 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 Search and filter clearance results for detailed information on creating clearance filters and saving them to the registry.

2. Create or modify a DBC file in the following way:

To	Add this line
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.

13. 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.

Uploading zones to 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 upload product attributes	**Do this**

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

(Windows) From the shortcut menu	In the product directory, right-click the *zones.csvcldb* 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,-1630,1500,18500,1
630,3400
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,-1735,2265,5642.8,1735,57
35
A-728AIRCRAFT,DEFAULT,7,ATA532_FWD_FUSELAGE,5642.8,-1735,2265,10570.4,173
5,5735
```

```
A-728AIRCRAFT,DEFAULT,8,ATA533_CENTRE_FUSELAGE,10570.4,-1735,2265,16299,1
735,5735
A-728AIRCRAFT,DEFAULT,9,ATA534_REAR_FUSELAGE,16299,-1735,2265,22056.4,173
5,5735
A-728AIRCRAFT,DEFAULT,10,ATA535_REAR_FUSELAGE,22056.4,-1735,2265,24514,17
35,5735
A-728AIRCRAFT,DEFAULT,11,UNPOS_PARTS,500,500,500,-500,-500,-500
```

14. 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 Design Context 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 standalone Mockup, the Teamcenter Lifecycle Viewer, Structure Manager, and Design Context. However, ClearanceDB results based on 4GD worksets can be evaluated only in standalone 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 server.

  For information about the platforms on which you can run Teamcenter lifecycle visualization, download the appropriate platform matrix spreadsheet from **Support Center**.
- An FMS file client cache (FCC).
- A Teamcenter user account with system administrator privileges.
- 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.

Enable batch clearance calculation for ClearanceDB

You must enable ClearanceDB functionality to utilize clearance results from your database.

1. On the **3D Clearance** toolbar, click **Clearance**  .
2. Choose **Clearance→Preferences→Requirement Components**.
3. In the **Requirement Components** dialog box, select one or more of these options:

Select this option	To obtain results in this way
Universal Clearance Requirement	From analysis that you conduct locally in Mockup.
Requirement Rules from ClearanceDB Server	According to requirement rules and conditions that you access in your organization's ClearanceDB Oracle database.

4. Choose **Clearance→Preferences→Result Components**.
5. In the **Result Components** dialog box, select one or more of these options:

> Note:
>
> You must choose **General Clearance Results** to work with results in the **Results** list.

Select this option	To work with results in this way
General Clearance Results	Visually display the results within Mockup.
ClearanceDB Results	Upload the results to your organization's Oracle database.

6. Click **OK**.

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 ClearanceDB — Administration. 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.

6. Specify your web server protocol in the global .vvi file.

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 upload the results to the database.

Performing adhoc ClearanceDB analysis

Although ClearanceDB analysis is most frequently performed as part of a regularly scheduled batch process from the command prompt, you can also manually analyze ClearanceDB products in real-time within the Viewer, according to the rules and conditions associated with the product in the database.

Adhoc ClearanceDB analysis is subject to all of the considerations and restrictions of ClearanceDB analysis, including the following:

- To analyze data from Teamcenter, you must configure ClearanceDB to work with Teamcenter-managed data. For information on configuring ClearanceDB to work with Teamcenter, refer to the Using ClearanceDB with Teamcenter section of *ClearanceDB — Administration*.

- To access ClearanceDB product information, including rules and conditions, the **clearance requirements** must be set to **Requirement Rules from ClearanceDB Server**.

- To upload the results of adhoc ClearanceDB analysis to the ClearanceDB database, the **clearance results settings** must be set to **ClearanceDB Results**.

Define ClearanceDB roles

To protect Teamcenter ClearanceDB data viewed in Active Workspace, assign users to one of the ClearanceDB roles within the ClearanceDB group. For more information, see Define ClearanceDB roles in Active Workspace.

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.

ObjectType or ItemRevisionType (Optional)

Specifies to perform analysis of 4GD worksets. If **ObjectType=Workset**, then **item**, **rev**, and **rev_rule** arguments are interpreted as those of a workset. If **ObjectType=ClearanceCalculationSubset**, then **ModelId**, **ClearanceCalculationName**, and **PartitionSchemes** arguments are interpreted as those of a clearance calculation subset. The **plmxml_export** utility will be invoked to extract the product structure from Teamcenter, rather than the BOMwriter utility.

ModelId, **ClearanceCalculationName**, and **PartitionSchemes** are Teamcenter variables used only for the script **analyze_managed_product.pl**, which runs clearance analysis on the managed product (clearance calculation subset) in Teamcenter.

> Example:
>
> **ModelId=JCB-Fastrac_id ClearanceCalculationName=CC_JCB-Fastrac PartitionSchemes= "Ptn0SchemePhysicalForJCB-Fastrac;Ptn0SchemeFunctionalForJCB-Fastrac"**
>
> If no Partition schemes are specified, no 4GD partition data will be exported.

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 indeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

Yes — If indeterminable variant conditions exist, evaluate them lexicographically.

> Note:
>
> Variant conditions that cannot be evaluated based on textual representations are indeterminable.

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.
EXCUDEDUIDPAIRS	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 **EXCUDEDUIDPAIRS** 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 .plmxml 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:
>
> **TeamcenterWebServerPath=http://tcserver:80/tc8_2008/**

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 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 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 configure your environment for Teamcenter utilities in *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) [4BYKHklYy$_loP]"

- 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 .plmxml 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:
   >
   > For more information, see the PLM XML/TC XML Export Import Administration application documentation.

2. Select the **4GDPIEDataExportTcVisClearanceDB** transfer mode from the **PLM XML 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 .plmxml file

When the property is added to the property set, it is exported to the PLM XML 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.

```
########################################
# Cpd0DesignElement
########################################
Cpd0DesignElement|ProductInstance
Cpd0DesignElement.object_name|ProductInstance.name
Cpd0DesignElement<fnd0affectedObject.Mdl0VariantExprBlock.fnd0express
ions>
      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($TARGET_END_OBJECT
Cpd0DesignElement|Occurrence.instanceRefs>ProductInstance
Cpd0DesignElement<cpd0presented_parent.Cpd0DesignElement|
      Occurrence.occurrenceRefs>Occurrence
Cpd0DesignElement|Occurrence.partRef>ProductRevisionView
Cpd0DesignElement.cpd0UG_ENTITY_HANDLE|Occurrence/UserData|
      add_user_value($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.fnd0i
s_
      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.

```
!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'

##########################################
# BOMWindow
##########################################
```

Caching managed data for analysis

You can configure ClearanceDB to use the **load_fcccache** 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_fcccache** utility are located in the *Clearance.cfgglobal* file.

> Note:
>
> 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 Teamcenter Administration in the Teamcenter help.

1. Using a text editor, open *Clearance.cfgglobal* 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 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 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.

Set the default owner for each clearance result

To show the default owner for clearance results, you must set the default owner attribute in the BOMwriter, which is then included in the generated .plmxml file.

1. In the *Clearance.cfgglobal* file, add the following attribute under the BomWriterUserAttributes section:

```
BomWriterUserAttributes=target:Instance,key:"Item Revision
Owning User",prop:bl_rev_owning_user,key:"Last Modified
Date",prop:awb0RevisionLastModifiedDate
```

2. In the *configuration.csvcldb* file, add the following attributes:

```
<product name>,EAICL_RELEVANT_PART_ATTRIBUTE,Last Modified Date

<product name>,EAICL_RELEVANT_PART_ATTRIBUTE,Item Revision Owning User

<product name>,EAICL_INITIAL_CLEARANCE_OWNER_ASSIGNMENT,ON

<product name>,EAICL_INITIAL_CLEARANCE_OWNER_CHOICE,Item Revision Owning
User

<product name>,EAICL_INITIAL_CLEARANCE_OWNER_CHOICE_CRITERION,Last
Modified Date
```

The configuration file changes can be added in the default product's *configuration.csvcldb* and then copied to all new products when they are created by using either the -uc (copy default product's *configuration.csvcldb* file) or -ua (copy all default product's configuration files) flag in the create_product.pl command line call. Alternatively, the configuration file changes can be added to the individual product's *configuration.csvcldb* file.

Analyzing managed ClearanceDB products

Analyze managed ClearanceDB products

Analysis of managed products is performed via the *analyze_managed_product.pl* script, which triggers the following events in sequence:

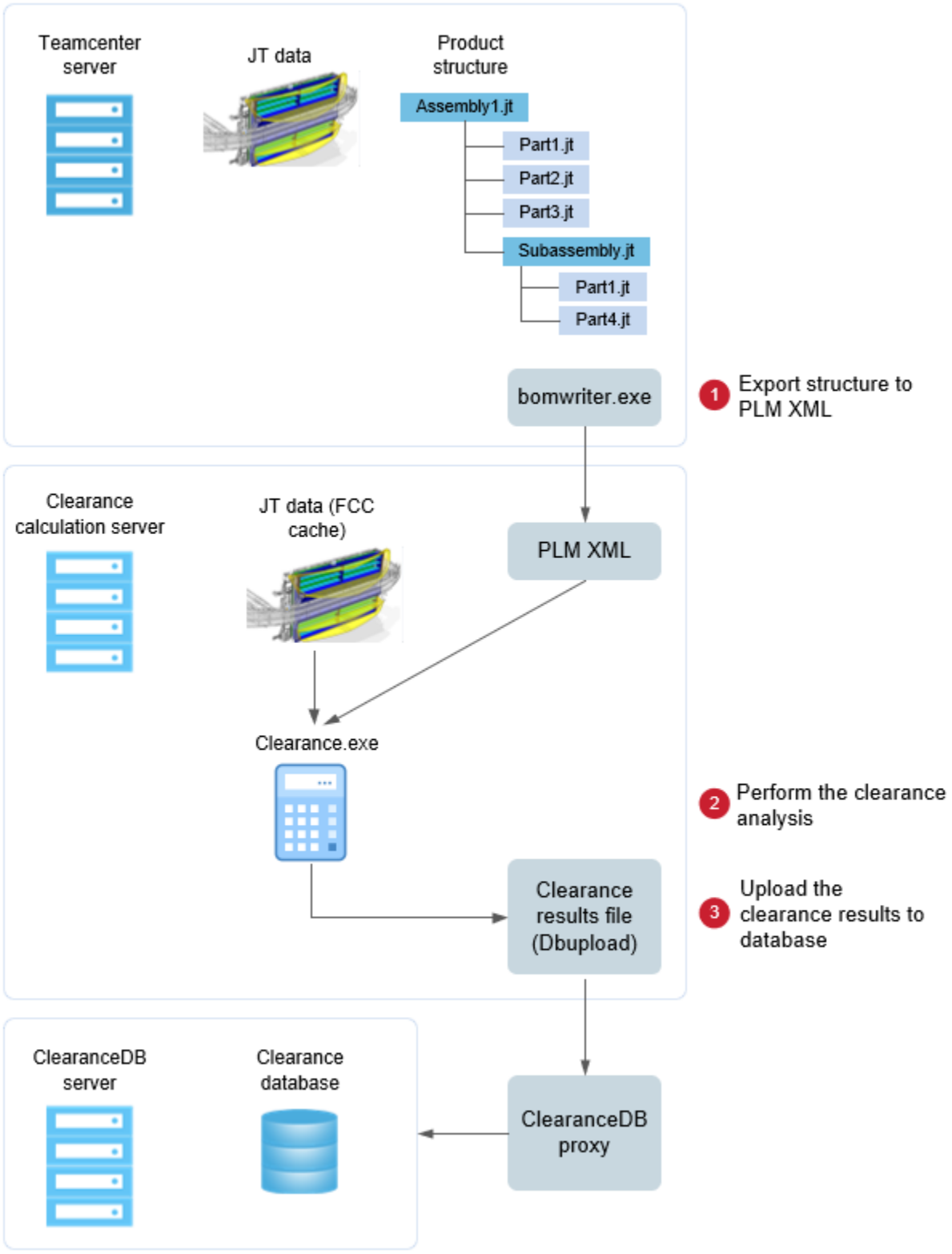
1. The BOMwriter utility generates a .plmxml file referencing the Teamcenter product data.

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.csvcldb* 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, as described in the Configure the Clearance Calculator section in *ClearanceDB — Administration*.

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

3. From the command prompt, type

```
analyze_managed_product.pl <product>
```

> Example:
>
> `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:

This RunLevel setting	Does this
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.

The analysis runs, and when it is complete the clearance results file, *ClearanceResultsDbUpload.csvcldb*, 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
#----------------------------------------

#----------------------------------------
# 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= -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
#----------------------------------------

#----------------------------------------
# Step 1b of 3: Run Load_fcccache.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_/141-f10002_A_1-Front_Suspension__view__orig.plmxml
Warning: Invalid "CopyOutLocation" in the configuration file. Use product dir by default.

Running:
Y:\\bin\load_fcccache -u=tcuser -p=xxxxxxxx
-config=C:/ClearanceDB_Work_Area/Clearance.cfgglobal
-f=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_/141-f10002_A_1-
Front_Suspension__view_.plmxml
-copy_out=C:/ClearanceDB_Work_Area/Products/141-f10002_A_1-Front_Suspension__view_/
StagingProcessDownloads -log_filename=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
DEBUG 2023-02-09T19:21:16.462Z: Setting logTypes to: -1
DEBUG 2023-02-09T19:21:16.462Z: Setting minor error: FILECOPY_FAILED, error code: 20
DEBUG 2023-02-09T19:21:16.478Z: Setting minor error: INVALID_DAKID_FORMAT, error code: 21
DEBUG 2023-02-09T19:21:16.478Z: Setting minor error: FCC_OPENFILE_FAILED, error code: 22
DEBUG 2023-02-09T19:21:16.478Z: Setting minor error: FCC_DOWNLOAD_FAILED, error code: 23
DEBUG 2023-02-09T19:21:16.478Z: Setting
minor error: GET_READ_TICKET_FAILED, error code: 24
DEBUG 2023-02-09T19:21:16.493Z: Setting minor error: DATASET_READ_FAILED, error code: 25
DEBUG 2023-02-09T19:21:16.493Z: Setting minor error: PLMXML_MISSING_JT, error code: 26
DEBUG 2023-02-09T19:21:16.493Z: Setting minor error: CHMOD_FAILED, error code: 27
DEBUG 2023-02-09T19:21:16.493Z: Setting
minor error: COPYOUT_CLEANUP_FAILED, error code: 28
DEBUG 2023-02-09T19:21:16.509Z: Setting BucketCount to: 30
DEBUG 2023-02-09T19:21:16.509Z: Setting UseAbsoluteLocation to: No
DEBUG 2023-02-09T19:21:16.603Z: Setting DirAccessMode to: 0750 (488)
DEBUG 2023-02-09T19:21:16.618Z: Setting FileAccessMode to: 0640 (416)
DEBUG 2023-02-09T19:21:16.618Z: Setting BucketPrefix to: RW
DEBUG 2023-02-09T19:21:16.618Z: Setting FilenamePrefix to: fmsr_
DEBUG 2023-02-09T19:21:16.618Z: Setting CopyOutLifetime to: 1209600
DEBUG 2023-02-09T19:21:16.618Z: Setting LifetimeCheck to: 0
DEBUG 2023-02-09T19:21:16.618Z: Setting the LifetimeCheckInterval to: 10
DEBUG 2023-02-09T19:21:16.634Z: Setting the LifetimeProcessLimit to: 300
FCC Interface Implementation fms.8.2.0.20091029.01(fms.8.2.0.20091029.01) initialized.

#----------------------------------------
# Load_fcccache.exe execution SUCCEEDED
# Execution time = 00:13:05
#----------------------------------------

#----------------------------------------
# Step 2 of 3: Run Clearance.exe
#----------------------------------------

Running:
C:/ClearanceDB_Work_Area/Products/141-f10002_A_1-Front_Suspension__view_/
clearance_exec.bat
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension__view_>set
VP_AUTO_TESTING=True

C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension__view_>set
EAI_CL_DATASOURCE=
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;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 -d 5 -c
C:/ClearanceDB_Work_Area/Products/141-f10002_A_1-Front_Suspension__view_/
141-f10002_A_1-Front_Suspension__view_.vvi

Teamcenter Visualization Clearance Testing.
Launching worker process.
Creating clearance components.
Loading input file.
Performing clearance test.
Generating pairs.
Product = 141-f10002/A;1-Front Suspension (view)(Latest Working)
2023/02/09-14:30:41: Connected to ClearanceDB Proxy on machine cldbmachine, at port 7206.
2023/02/09-14:30:44: Reconnected to
ClearanceDB Proxy on machine cldbmachine, at port 7206.
2023/02/09-14:30:45: Reconnected to
ClearanceDB Proxy on machine cldbmachine, at port 7206.
Number of unique rules = 1
2023/02/09-14:30:48: Reconnected to
ClearanceDB Proxy on machine cldbmachine, at port 7206.
2023/02/09-14:30:49: Reconnected to
ClearanceDB Proxy on machine cldbmachine, at port 7206.
The statistics of 141-f10002/A;1-Front Suspension (view):
   Number of element occurrences           =        55
   Number of unchanged element occurrences =         0
   Number of excluded element occurrences  =         0

The requirement engine called total of 1461 times.
Clearance test bed has been created with
CADID matching keyed by PART names.
All pairs have been generated.
Processing pairs.
Launching 2 processes for analysis.
Process 1: All pairs have been processed.
Process 2: All pairs have been processed.
Generating results.
Testing finished.  Results written to 'ClearanceResultsDbUpload.csvcldb'.
Process ended normally.

#----------------------------------------
# Clearance.exe execution SUCCEEDED
# Execution time = 00:01:12
#----------------------------------------

#----------------------------------------
# Step 3 of 3: Run Update_product.pl
#----------------------------------------

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 .....

2023/02/09-14:31:09: Connecting to Clearance Database...

2023/02/09-14:31:09: Connected to ClearanceDB Proxy on machine cldbmachine, at port 7206.
2023/02/09-14:31:09: Successfully connected to Clearance Database.
```

```
2023/02/09-14:31:09: Lock the Clearance Database for results update...

2023/02/09-14:31:11: Starting upload for C:/ClearanceDB_Work_Area/Products/
141-f10002_A_1-Front_Suspension__view_/ClearanceResultsDbUpload.csvcldb ...

2023/02/09-14:31:11: 100% complete.

2023/02/09-14:31:11: Starting merge results.
 This could take quite a while, please be patient...

2023/02/09-14:31:17: Merge successful.

2023/02/09-14:31:17: Upload successful.


#----------------------------------------
# Update_product.pl execution SUCCEEDED
# Execution time = 00:00:08
#----------------------------------------

#-----------------------------------------------------------------------
# analyze_managed_product.pl complete
#
# Step 1a: Bomwriter.exe.       Result = SUCCEEDED.     Time=00:15:03
# Step 1b: Load_fcccache.exe.   Result = SUCCEEDED.     Time=00:13:05
# Step 2: Clearance.exe.        Result = SUCCEEDED.     Time=00:01:12
# Step 3: Update_product.pl.    Result = SUCCEEDED.     Time=00:00:08
# Total Time = 00:29:31
#-----------------------------------------------------------------------
Completed: Tue Feb  9 14:31:17
C:\ClearanceDB_Work_Area\Products\141-f10002_A_1-Front_Suspension__view_>
```

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 Design Context

Work with Design Context

You can combine ClearanceDB with Teamcenter's Design Context application to create an Integrated Clearance Management (ICM) system. Design Context 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 Design Context, 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 Design Context.

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 Design Context 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 Design Context, 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 Design Context **Issues** panes.

  A SCO (Structure Context Object) containing the clearance results can also be created in the *Newstuff* folder, provided the **RDVCreateSCOForClearance** 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 Design Context, see Design Context in the Teamcenter help.

Design Context clearance analysis modes

Database query mode

The ClearanceDB administrator configures the scope of the analysis and the Design Context end user can only view the results. The end user can use the Design Context 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 Design Context 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 Design Context 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.

> Note:
>
> You can also view additional results information such as issue **Status**, **Priority**, and **Date First Found**. To update the selection of columns shown in the **Issues** panes, modify the **ClearanceDBShownColumns** preference in the rich client.

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 Design Context to see the clearance analysis results.

To perform real-time clearance analysis, the end user manually selects BOM lines of interest in Design Context. BOM lines involved in the analysis process are Design Context target parts. The user selects these parts from the target and background BOM lines that are configured in the Design Context 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 analyzes the BOM lines selected as Design Context target parts against all other BOM lines in the Design Context third window. It presents the issues in the following categories:

- Target parts against all the appearances in the Design Context target appearances table in the third window.
- Target parts against all the appearances in the Design Context 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.

> Note:
>
> Creation of an SCO is optional and occurs only if the **RDVCreateSCOForClearance** preference is set to `true` in the rich client.

Specify connection information for Design Context

For Design Context to connect with the ClearanceDB database, you must use the Teamcenter **RDVClearanceProxyServers** preference to specify the ClearanceDB connection information.

> Note:
>
> You must have Teamcenter administrator rights to modify the **RDVClearanceProxyServers** 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 `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

For this parameter	Type this
`machine_name`	The name of the system running the ClearanceDB Proxy or Oracle Client.
`port`	The port number used for the ClearanceDB connection.
`connection_name`	The name of the Oracle connection.

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 Design Context **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 Design Context

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 Design Context** button.

   Teamcenter opens the relevant BOM lines in Design Context, 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 Design Context window. If you do not select any lines, Teamcenter includes all appearances in the third Design Context 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 Design Context window.

2. Click **Finish**.

   Teamcenter opens the third Design Context 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.

Working with variants

Understanding variant-based clearance analysis

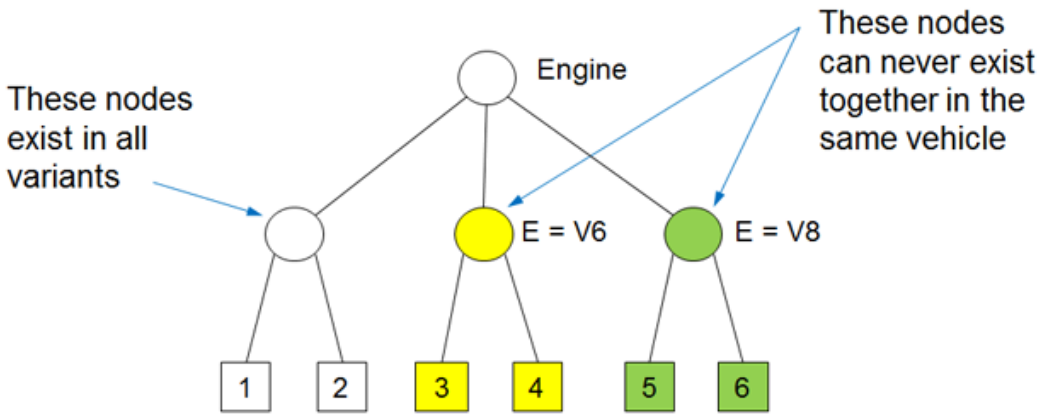
Variant-based clearance analysis uses product variant information stored in Teamcenter. Analyses can be performed on a 150% BOM structure, which is the union of two or more product variants. Non-buildable combinations of product structure are detected and excluded from the clearance analysis.

The benefits of removing non-buildable occurrence relationships from the ClearanceDB analysis include the following:

- Eliminate results between non-buildable product configurations.
- Faster clearance analysis.
- Fewer clearance incidents for engineers and designers to evaluate.

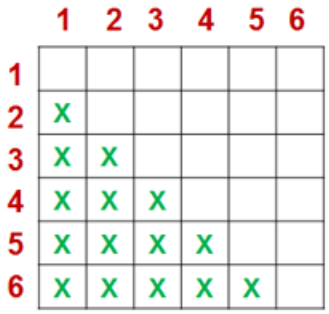
A simple example with variant conditions (150% BOM with variants)

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.



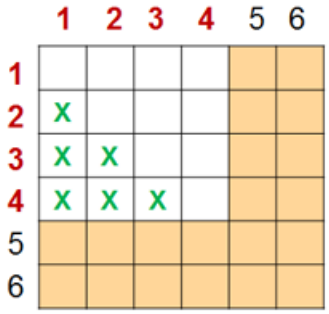
With no variants, all nodes are analyzed (15 part pairs).

No nodes are excluded.



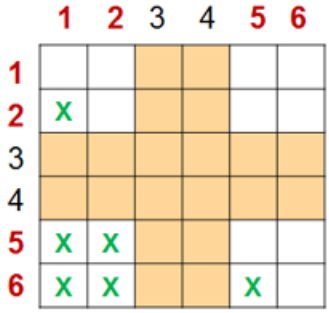
With the E=V6 variant, nodes 1, 2, 3, and 4 are analyzed (6 part pairs).

Nodes 5 and 6 are excluded (9 part pairs).



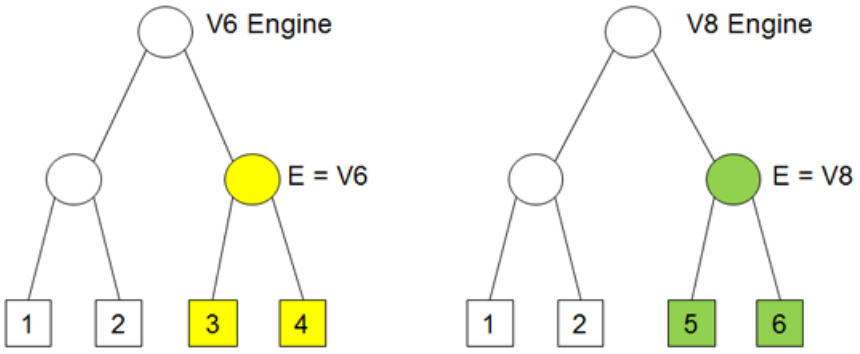
With the E=V8 variant, nodes 1, 2, 5, and 6 are analyzed (6 part pairs).

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.

	1	2	3	4
1				
2	X			
3	X	X		
4	X	X	X	

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.

	1	2	5	6
1				
2	X			
5	X	X		
6	X	X	X	

For more information about working with Teamcenter variant conditions, see Structure Management on Rich Client — Usage in the Teamcenter help.

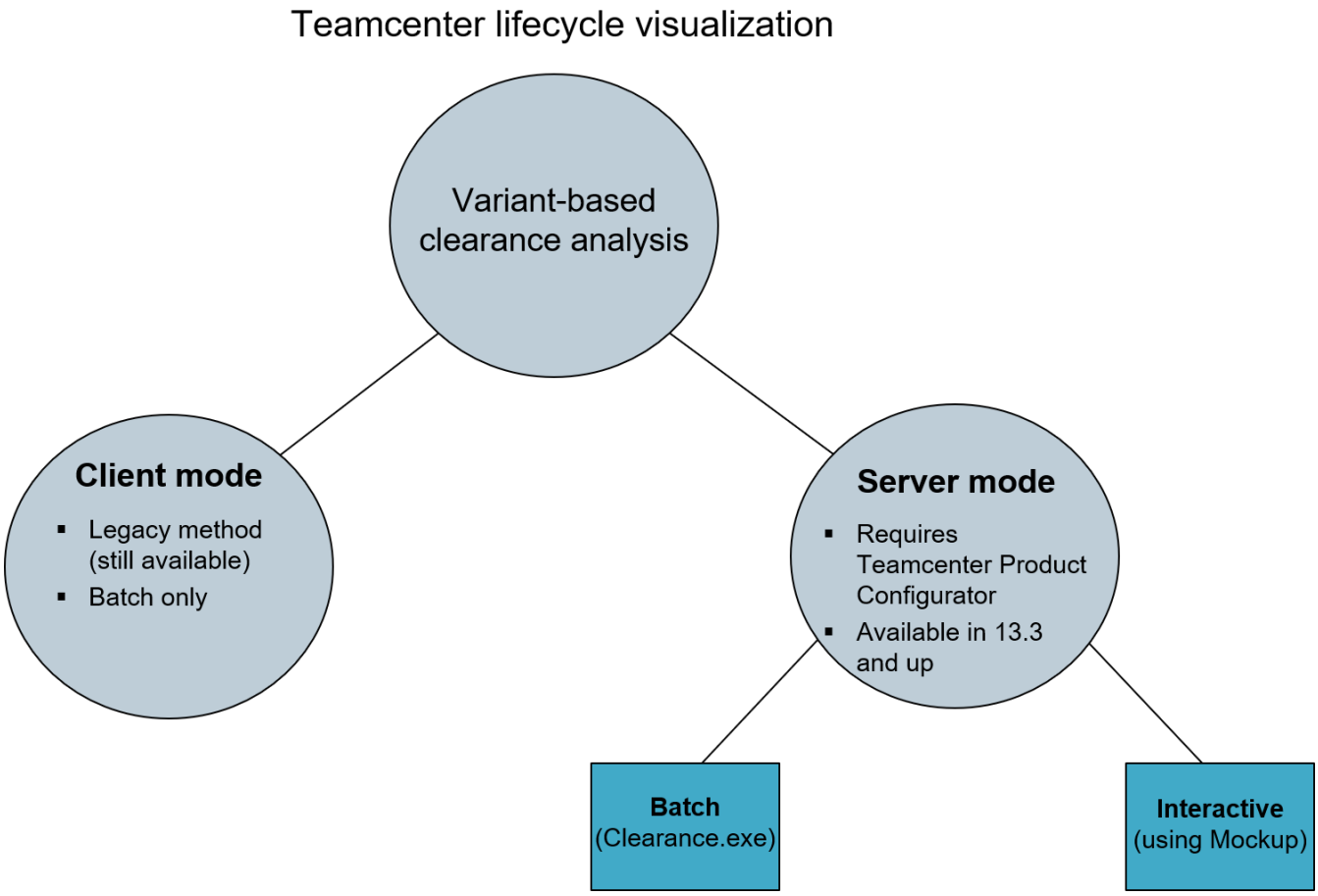
Variant-based clearance analysis methods

You can perform variant-based clearance analysis using the following methods:

- Client-side
  - No Teamcenter connection is required during the analysis.
  - Can only be run in batch using *Clearance.exe*.
- Server-side

- Requires a connection to Teamcenter Product Configurator during the analysis.
- Can be run either interactively in Teamcenter lifecycle visualization or in batch using *Clearance.exe*.

See Product Configurator on Rich Client to learn more about creating, modifying, and managing variants in Teamcenter.



Variant-based clearance analysis (client-side)

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 indeterminable variant conditions exist, the affected variant inferences are asked from the server. This is the default.

Yes — If indeterminable variant conditions exist, evaluate them lexicographically.

> Note:
>
> Variant conditions that cannot be evaluated based on textual representations are indeterminable.

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.
EXCUDEDUIDPAIRS	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 **EXCUDEDUIDPAIRS** 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 .plmxml file. This has the form of:

“Variant Condition”:bl_variant_condition

“Variant Formula”:bl_formula

> Example:
>
> **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:

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-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.

Variant-based clearance analysis (server-side)

Performing variant-based clearance analysis in Lifecycle Visualization

1. With an unconfigured .plmxml file open, choose **Menu→Clearance→Preferences→Product Configurator**.

2. In the **Product Configuration** dialog box, do the following.

   a. Select the **Enable Product Configurator** check box.

   b. From the **Variant Rule** dropdown, select the desired variant rule or criteria to apply against the loaded file during clearance analysis.

   c. Click **OK**.

3. Run a clearance analysis.

Performing variant-based clearance analysis in batch

To perform variant analysis in batch mode (*Clearance.exe*) using Teamcenter Product Configurator, complete the following steps.

1. Ensure that you are connected to Teamcenter and that the connection is maintained for the duration of the analysis process.

2. 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, which will override settings in the *Clearance.cfgglobal* file.

   EnableProductConfigurator

   Enables *Clearance.exe* to use the Teamcenter Product Configurator to perform variant analysis.

   > Example:
   >
   > **EnableProductConfigurator=True**

   ProductConfiguratorVariantRule

   Specifies the saved variant rule that is used for communicating with the Product Configurator.

   > Example:
   >
   > **ProductConfiguratorVariantRule=**<*valid variant rule*>

   VariantLogicalExpression

   Define the variant condition UserValue title and BOM line property pair in the .plmxml file. This has the form of:

“Variant Condition”:bl_variant_condition

“Variant Formula”:bl_formula

> Example:
>
> **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:

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

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

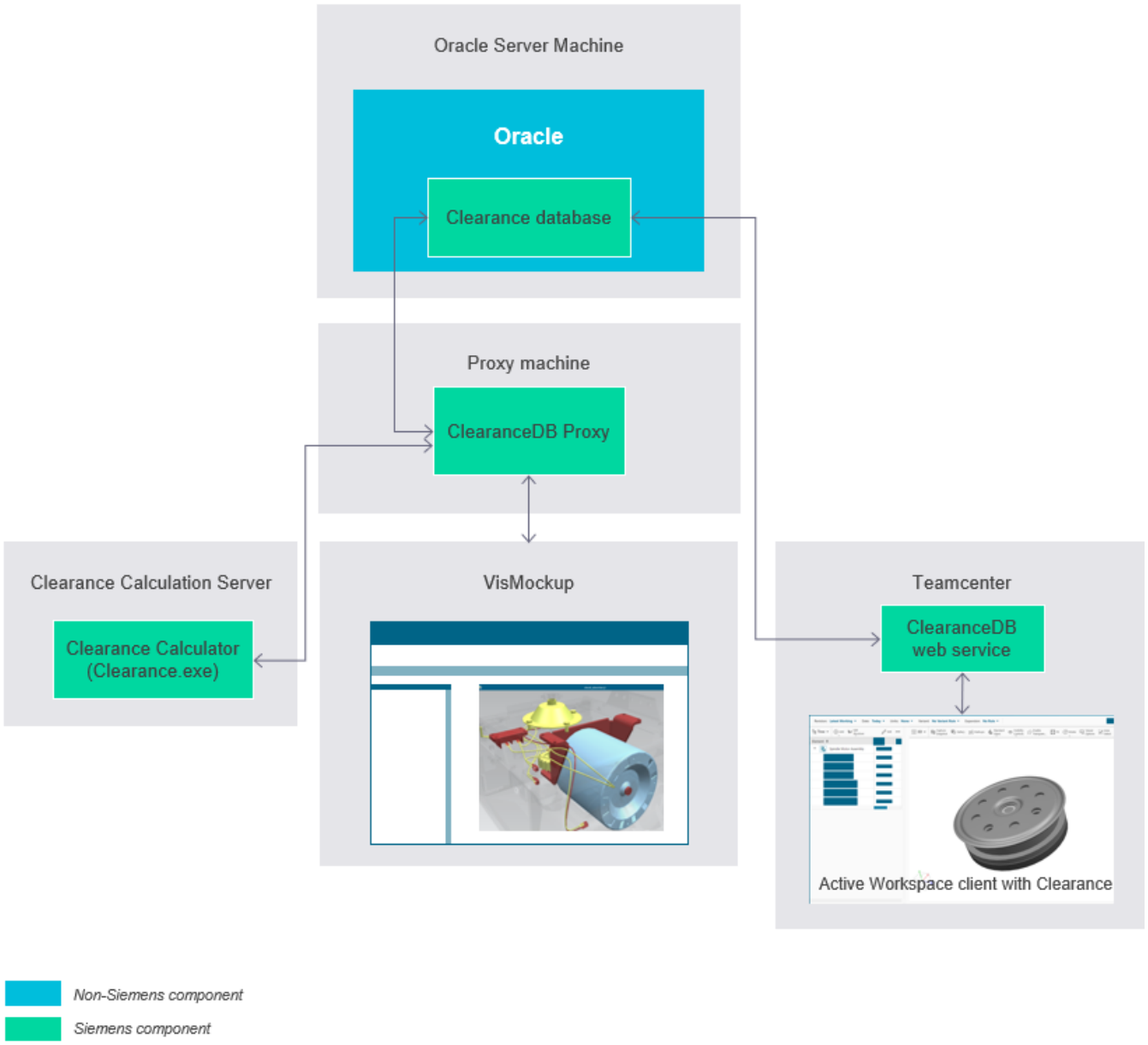
> Example:
>
> Use this option to include variant model related BOM line properties in the generated .plmxml file. For example:
>
> **BomWriterUserAttributes=target:Instance,key: VC-V6,prop: bl_variant_condition**

3. Save the file.

Using ClearanceDB with Active Workspace

Using ClearanceDB with Active Workspace

The ClearanceDB with Active Workspace client uses a web service to connect to the ClearanceDB. For more information, see *Connect to an existing 3D Clearance database using Deployment Center*.



Clone stable IDs

To view clearance results in Active Workspace, you must specify the clone stable ID for the BOMwriter to include in the generated .plmxml file. You can specify this attribute in the *Clearance.cfgglobal* file if you do not have specific metadata for every product, or in the *Clearance.cfgproduct* file if you have product metadata.

Add the following attribute under the BomWriterUserAttributes section in either of the files identified above.

```
BomWriterUserAttributes=target:Instance,key:__PLM_CLONE_STABLE_INST_UID,prop
:bl_clone_stable_occurrence_id
```

Note:

After adding this attribute to either the *Clearance.cfgglobal* or *Clearance.cfgproduct* file, run the *update_ngids.pl* utility for each product to ensure that the clearance results for each product can be viewed in Active Workspace.

Run the clone stable IDs utility

After **adding the clone stable ID attribute** to either the *Clearance.cfgglobal* or *Clearance.cfgproduct* file, you must run the *update_ngids.pl* utility for each product to ensure that the clearance results for each product can be viewed in Active Workspace. If you add the attribute but do not run the utility, when the clearance results are updated your end users may not be able to view the updated results in Active Workspace.

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 configure your environment for Teamcenter utilities in *Utilities Reference*.

2. At the command prompt, type:

   update_ngids.pl

3. Run the utility for each product you want to view in Active Workspace.

15. Maintaining the database

Database maintenance overview

This section provides information on maintaining and optimizing the database.

The administrator may choose 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 (Linux) 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:

Type	To do this
exec eaicl_p_clean	Delete obsolete objects from the database. 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.
exec eaicl_p_reset_gc_attr (eaicl_p_product.product_id ('<product name>'))	Make subsequent applications absolute rather than incremental.

Type	To do this
exec eaicl_p_product.del_product (eaicl_p_product.product_id ('<product name>'),'<user_name>')	Delete all the data associated with a product.
exec eaicl_p_product.ren_product ('<old_product name>', '<new_product name>', '<user_name>')	Rename a product. Only the name of the product changes.
exec eaicl_p_product.cpy_product ('<old_product name>', '<new_product name>', '<user_name>')	Copy a product. All associated data is copied, except the issue modification/deletion history.

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 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 uploads 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 rulebase 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.

16. 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 **Tools→Clearance**.
2. Choose **Clearance→Preferences→Requirement Components**.
3. In the **Requirement Rule Components** dialog box, select **Requirement Rules from ClearanceDB Server**.
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**.
4. Select **Filter Operator = !=**.
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 you must have a license for the Mockup product configuration with the **Analysis** option, and you must specify the licensing for the Lifecycle Viewer and the Teamcenter embedded viewers.

Visualization service level can be specified in the **Visualization Licensing Level** option in the Teamcenter Organization application for the desired user. It is set to **Base** by default.

Licensing for ClearanceDB is specified in the *client_specific.properties* file, which is located in the *<RAC_install_location>\plugins\configuration_<version>* directory on client machines.

Example:

This example shows the visualization licensing parameters in the *client_specific.properties* file set with all available options:

```
PortalViewer_Optional_Licenses=Simplified Rendering,
ECAD,Analysis,Path_Planning,Concept_Desktop,Collaboration,
Visual_Report, STEP, JT_Inspect_Workflow
```

Note:

If the `PortalViewer_Optional_licenses` line does not exist, add the following line to the **client_specific.properties** file, including the names of applicable functionality options.

This line	Requires this information
`PortalViewer_Optional_Licenses`	The names of functionality options not included in the default product configurations. Type the values, separated by commas, listed below that match any additional visualization licenses you have purchased: • `Simplified_Rendering`

This line	Requires this information
	• ECAD • Analysis • Path_Planning • Concept_Desktop • Collaboration • Visual_Report • STEP • JT_Inspect_Workflow

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:

- **PLM XML Units** — Specifies the unit of measurement for .plmxml files loaded in the viewer. This option is located on the **File→Preferences→PLM XML** menu.

- **Calculator Settings** — Specifies clearance options such as the element type. These settings are located on the **Clearance→Preferences→Calculator Settings** menu.

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. 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.

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.

Loading clearance results after an Oracle upgrade

You will need to perform the following procedure if the following scenarios occur:

- After upgrading your Oracle version, if you are unable to load the clearance results from the Clearance database.

- You encounter errors when uploading to the database using the update_product.pl script. For example, if you run update_product -ur to update rules and conditions and receive the following error:

  Example:

  ```
  2021/02/03-11:48:25: Privilege CREATE JOB has not been granted to
  CLOWNER.
  2021/02/03-11:48:25: The ClearanceDB Oracle user is lacking system
  privileges.
  ```

1. From a command shell, log in to SQL with SYSDBA privileges.

2. Enter the following at the SQL prompt, where, <clearancedb user name> is the clearancedb user name.

   ```
   GRANT CREATE JOB TO <clearancedb user name>;
   ```

17. Other resources

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

For documentation on working with clearance results in standalone 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 Design Context, refer to Performing clearance analysis and proximity filtering in *Design Context* in the Teamcenter help.

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.