

PTC®

Using Creo Parametric® with Windchill®

Windchill 10.1 M040

April 2013

Copyright

Copyright © 2013 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION. PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

Important Copyright, Trademark, Patent, and Licensing Information: See the About Box, or copyright notice, of your PTC software.

UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 01282013

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

Copyright	2
About This Guide	7
Getting to Work with Creo.....	11
Some Quick Basics	12
Collecting Objects for PDM Operations	14
Setting an Object Location	19
Saving and Uploading Objects	20
Checking In Objects	23
Checking Out Objects	26
Adding Objects to the Workspace.....	31
Removing Objects from the Workspace	32
Keeping Workspace Objects Up-to-Date	32
Refreshing the Cache.....	33
Importing Objects to the Workspace	36
Exporting Objects from the Workspace	37
Revising Workspace Objects	38
Using the Event Management Utility	39
Advanced Object Handling in Windchill	45
Modifying Object Attributes (Properties)	46
Renaming Objects	48
Deriving New Designs Using Save As	49
Opening Objects in Creo Parametric.....	53
Working with Family Tables.....	55
CAD Document Templates and Creo Parametric Start Parts	66
Using Library Parts	68
Managing Incomplete Dependent Objects.....	69
Simplified Representations	72
Managing Model Items	74
Managing Part-CAD Document Relationships	74
Verifying Windchill Editing Instructions	88
Heterogeneous Design.....	88
Working with Configurable CAD Documents.....	90
Customizing and Administering Creo Parametric with Windchill	95
Configuration Settings in Creo Parametric.....	97
Configuring Windchill for Interoperation with Creo Parametric	109
System Configuration Recommendations	154
Performance Tuning.....	155
Other Recommendations.....	158
Windchill Preferences That Control Interaction with Creo Parametric	160

Quick Reference for Menus, Icons, and Symbols	225
Using OIRs for Naming and Numbering	229
Setting Name and Number to the Same, Non-editable Autogenerated Value	231
Turning Off All Autonumbering	233
Setting Editable Autogenerated Values	235
Setting Non-editable Autogenerated Values	238
Setting Editable, Identical Value for Name and Number	240
Setting Editable, Non-autogenerated Values	242
Setting Autogenerated, Non-editable Values for Number	244
Setting Pre-generated, Editable Values	246
Setting Pre-generated, Non-editable Values	249

About This Guide

Using Creo with Windchill is an introduction to product data management (PDM), using Creo Parametric to manage product data in Windchill, both for basic and for more advanced functions. If you follow the content of this manual, you will see how Creo Parametric interacts with Windchill products and how you can use it to manage your product development cycle.

Intended Audience

The intended audience for this guide is broad and includes:

- New and experienced Creo Parametric users with little or no PDM or Windchill software experience.
- Users who have some experience in working with product data management software, but who would like to review the basics or learn more about how best to use Windchill solutions with Creo Parametric.
- Users new to Windchill but who have worked with other product data management applications. The philosophy behind Windchill closely follows the PDM system paradigm. These users will learn the specific procedures of using Creo Parametric with Windchill.
- Seasoned Windchill users who would like to learn about how Creo Parametric interacts with Windchill.
- Administrative users responsible for administering and configuring the interoperation of Creo Parametric with Windchill.

Scope and Purpose

This guide is not intended to be a complete summary of Windchill functionality. The goal of this manual is demonstrate to you how to use Creo Parametric with Windchill to achieve effective product data management.

Related Documentation

The following documentation may be helpful:

- User's guides for Windchill PDMLink, Windchill ProjectLink, or Pro/INTRA-LINK 9.0 and Windchill PDMLink available at the following link:

<http://www.ptc.com/appserver/cs/doc/refdoc.jsp>

- Creo Parametric online help
- Windchill online help

Technical Support

Contact PTC Technical Support via the PTC Web site, phone, fax, or e-mail if you encounter problems using Creo Parametric, Windchill PDMLink, Windchill ProjectLink, or the product documentation.

For complete details, refer to Contacting Technical Support in the *PTC Customer Service Guide*. This guide can be found under the Related Links section of the PTC Web site at:

<http://www.ptc.com/support/index.htm>

The PTC Web site also provides a search facility for technical documentation of particular interest. To access this page, use the following URL:

<http://www.ptc.com/support/support.htm>

You must have a Service Contract Number (SCN) before you can receive technical support. If you do not have an SCN, contact PTC Maintenance Department using the instructions found in your *PTC Customer Service Guide* under Contacting Your Maintenance Support Representative.

Documentation for PTC Products

You can access PTC documentation using the following resources:

- **Windchill Help Center**—The Windchill Help Center is an online knowledge base that includes a comprehensive index of all Windchill documentation. You can browse the entire Windchill documentation set, or use the search capability to perform a keyword search. To access the help center, you can:
 - Click any help icon  in Windchill
 - Select **Help** ► **Windchill Help Center** from the **Quick Links** menu at the top right of any Windchill page
 - Use the following link to access all PTC help centers:
<https://www.ptc.com/appserver/cs/help/help.jsp>
- **Reference Documents Website**—The Reference Documents website is a library of all PTC guides:
<http://www.ptc.com/appserver/cs/doc/refdoc.jsp>

A Service Contract Number (SCN) is required to access the PTC documentation from the Reference Documents website. For more information on SCNs, see the PTC Technical Support page:

<http://www.ptc.com/support/index.htm>

Comments

PTC welcomes your suggestions and comments on its documentation. To submit your feedback, you can:

- Send an email to documentation@ptc.com. Include the name of the application and its release number with your comments. If your comments are about a specific help topic or book, include the title.
- Click the PTC help center feedback icon  in the upper right of a Windchill Help Center topic and complete the feedback form. The help topic title is automatically included with your feedback.

Getting to Work with Creo

Some Quick Basics	12
Collecting Objects for PDM Operations.....	14
Setting an Object Location.....	19
Saving and Uploading Objects.....	20
Checking In Objects.....	23
Checking Out Objects	26
Adding Objects to the Workspace	31
Removing Objects from the Workspace	32
Keeping Workspace Objects Up-to-Date	32
Refreshing the Cache	33
Importing Objects to the Workspace.....	36
Exporting Objects from the Workspace.....	37
Revising Workspace Objects	38
Using the Event Management Utility.....	39

This manual provides detailed instructions on using Creo Parametric with Windchill to enhance Data Management. In this chapter you are introduced to the primary concepts and functions that will comprise your Creo Parametric session. The chapter begins by outlining the most frequently used functionality of Creo Parametric, with links to more detailed information on each topic. Subsequent chapters explain functionality of interest to advanced users and administrative configuration information and recommendations. The first appendix of this guide provides a [Quick Reference for Menus, Icons, and Symbols on page 225](#).

Some Quick Basics

The following sections outline key operations for using Windchill PDM with Creo Parametric.

Connection to a Windchill Server

Server registration enables connection and interaction with a Windchill server from Creo Parametric.

PURPOSE

Registering a Windchill server allows you to work in a collaborative design environment. Instead of a local working directory, you manage your designs in a project-related workspace (essentially a private folder on the Windchill server). For more information, see the Windchill Help Center, “Workspace Page Functionality. In a connected session, your work can be saved and uploaded to a secure server location. For more information, see [Saving and Uploading Objects on page 20](#). When you’re ready to share your work, you check it in to the Windchill commonspace. For more information, see the Windchill Help Center, “Introduction to the Workspace.”

Note

The preference Tables > Size Limit (which specifies the maximum number of objects to be displayed in a Windchill table) does not apply to the workspace Object List table.

WHAT IS INVOLVED?

Use the Server Management utility to register a PDM server:

- Click **Tools** ► **Server Management** to access the **Server Management** window.
- Enter a name and valid URL for the server.
- Select a workspace.
- Set the workspace to be your active workspace and make the server your primary server.
- Your server and workspace will then appear in the Folder Navigator.

Note

It is possible that different Windchill users might share the same client machine. If each user doesn't use his or her own special startup script pointing to their specified cache location, it is possible for a user to inadvertently access the client side cache belonging to another user. To avoid this situation, the system will pop up a warning message indicating that the current cache location specified by the client application does not match the user's ID and continuing could cause unwanted results or cache corruption. The user has the ability to either continue or cancel, with Cancel being the default selection.

For more information on server registration, see the Windchill Help Center, “About the Server Management Utility.”

Note

Connecting to a server involves authentication. For more information on authentication, see the topic, “Authentication” in the Windchill Help Center.

Storing New Designs

Proper storage of and access to design files is a key aspect of Windchill PDM.

PURPOSE

Once you have created or modified your design files you can store them appropriately on the Windchill server. The benefits that the Creo Parametric interaction with Windchill brings include:

- Product- or project-centered design environment (pre-defined templates, parameters)
- Access to other designs both within your project and in enterprise libraries.
- Workspace preferences can be leveraged to automatically store your designs as you have specified, minimizing the amount of required user input.

WHAT IS INVOLVED?

Check in—Saves, uploads, and creates a new iteration of the object on the server, making your design available to others with server access. For more information on checking in design files, see [Checking In Objects on page 23](#).

Developing Existing Designs

You can access existing designs on the PDM server.

PURPOSE

Collaborating with other designers or developing your own designs is made easier because Creo Parametric has access to your Windchill database. Windchill automatically checks for proper access to design files and enables you to view files and (upon checkout) reserve them for modification.

WHAT IS INVOLVED?

- Browse or Search in Windchill—Allows you to find files by browsing server locations or by criteria-based search.
- Add to Workspace—Places design metadata and (optionally) content into your workspace.
- Check Out—Adds data to your workspace and locks the server copy, giving you sole modification rights. For more information about searching in Windchill, see online help.

For more information about Add to Workspace and Check Out, see [Adding Objects to the Workspace on page 31](#) and [Checking Out Objects on page 26](#).

Keeping Designs Current

Working with appropriate versions of objects is essential.

PURPOSE

In any collaborative environment, it's important to know that you're working on the appropriate version of your design, as design work may be shared among different teams. By default, your workspace preferences will specify that you work with the latest versions of design files; or you can set custom preferences to ensure that a particular configuration is specified. Status symbols in your workspace listing table indicate whether a current object version meets your specifications. For more information on object status see the Windchill Help Center, "About Object Status."

Explicit commands available in the workspace allow you to update your files if a new iteration becomes available in the commonspace.

WHAT IS INVOLVED?

- Update—Brings workspace files current with latest iterations available on server.
- Synchronize—Freshens workspace files with latest information from the server (for example, attribute modifications done by you in a standalone Windchill session).

For more information on updating and synchronizing workspace objects see [Keeping Workspace Objects Up-to-Date on page 32](#).

Additional PDM Activities

The previous sections have outlined some basic actions that are part of using Windchill PDM with Creo Parametric. The following sections of this chapter and the chapter, [Advanced Object Handling in Windchill on page 45](#), discuss these and additional PDM topics in detail.

Collecting Objects for PDM Operations

A basic practice for many PDM operations (also referred to as actions) is specifying the set of objects upon which you want to perform the operation. Though the set might only consist of a single object in your workspace, more often it is made up of one or more assemblies, with or without other dependent objects and associated enterprise parts, and referencing a particular configuration (either latest or as-stored, or based on a particular baseline or part effectivity). This section discusses how you can collect the exact objects that you want for your PDM activities.

The workspace is your typical starting point for a PDM action, although actions can be initiated in many places throughout Windchill. Initially, you select an object (or several) and then click the button corresponding to the desired action. An action page appears, with your initially selected objects listed in the **Object List**. Arriving at your final collection typically follows this sequence:

1. Initial Selection – Generally you select a top-level object
2. Adding or removing related objects – Collection controls let you add or remove related objects by setting rules
3. Specifying a configuration for the objects – Embedded or pop-up fields allow you to set the desired configuration
4. Excluding unwanted objects – Objects collected by the foregoing steps can be selectively excluded.
5. Setting options (not part of all actions) – Specifying how objects will be treated upon the execution of the action

About Dependency Processing

Dependency processing refers to the tracing of the object to object relationships among the objects considered for inclusion in the configuration you specify for any of the various PDM actions. A key distinction is between part-centric and document-centric dependency processing.

Essentially, part-centric processing traces a product structure for dependencies while document-centric processing traces a CAD document model structure. Either method can be selected, regardless of whether the initially-selected object is a part or a CAD document, so long as there is an active association between the part and CAD document.

For example, for part-centric processing of an initially-selected CAD document :

- The CAD document remains in the object list
- The associated part object is included in the object list
- The associated part becomes the "root" object for:
 - Setting configuration (latest, managed baseline names, view, effectivity)
 - Collecting dependents (all, none)

For example, if:

configuration = managed baseline

dependents = all

-
- Then only the part dependents that are also part of the managed baseline are included in the object list.
- The possible related objects to be included (CAD documents, documents)
 - CAD documents will be the version (revision/iteration) that is peer to the gathered part version
 - If a managed baseline for parts is selected, the CAD documents do not need to be part of the baseline.

When a workspace specification has dependency processing set to part centric, a CAD document is considered out-of-date if the version in the workspace does not have an active associate link to the part version included in the specified baseline for parts.

Configuration

Most PDM actions involve collecting groups of objects for you to act upon. Typically, you initially select a one or more key objects and then gather a larger set of dependent objects that you want to include in the action based upon their relationship to the initially selected object. The system chooses a default group of objects that you can then modify to suit your intended task. The particular set of object types and iterations you gather is the configuration applicable for the action.

The configuration controls need to be accessed whenever you want to change the configuration rule for collecting objects. A configuration may be identified for each object selected for a particular action.

When specifying a configuration, you first identify whether the configuration is based on an enterprise part, end object, product, or serial numbered part (a Product Structure, specifying part-centric processing); or on a CAD document (a Model Structure, specifying document-centric processing). This choice determines the set of configuration options (latest, managed baselines; as stored configurations; per workspace configuration specification; or based on part effectivity) available for selection, as follows:

- For Model Structure – Latest, as stored configurations, promotion requests, per workspace configuration, and managed baselines that include the iteration of the single initially selected CAD documents or the CAD documents actively associated to the initially selected parts in the object list.
- For Product Structure – Latest, per effective date, managed baselines, per workspace configuration, and promotion requests that include the iteration of the initially selected parts or the parts actively associated to the initially selected CAD documents in the object list.

If the initially selected object has an active association to a peer object (for example, a CAD document has an active association to a part, or vice versa) you can collect iterations of dependents based on a configuration specification for the peer object.

You may change the rule for collecting dependents.

You may change the configuration of the objects to be collected.

If the action is initiated from a workspace, or results in objects being added to a workspace, configuration changes may be written to the workspace configuration specification (when the configuration is committed).

When you commit the selections you make in the configuration tool, your changes are applied to the object list.

Note

Configuration changes have effects if you have already collected related drawings, parts, CAD documents, or instances, or edited the list in any other way (for example, using exclude). Upon changing a configuration, the collected objects that are related to the new configuration remain in the list. However, if the property `core.collection.collectall` is set to `false` in `wt.properties`, all of the collected related objects are removed from the list upon changing the configuration.

Changes to configuration settings never remove initially selected objects from the object list.

Changes to configuration settings can change the iterations of dependent objects added to the action list; configuration settings never change the iteration of initially selected objects in the object list.

For the Update action, the iteration of the initially selected object is the up-to-date iteration, not the iteration that was initially selected. Also, when using effectivity for Product Structures, the iteration of the initially selected objects will be the effective one per the date specified, which may be different from the iteration that was initially selected.

Using the Collection Tools

The collection tools available for PDM actions include configuration specification and are present in any action page where collecting additional object for the action is supported. Two modes of collection, termed basic and advanced, are provided to perform the collection. The basic mode is rule-based and can be preset by administrators with default rules to simplify the collection process (though users may be able to override the rules). The advanced mode allows users to see the initially selected and subsequently collected objects in a table view, and to act on objects individually, using menu options provided for the table. For many actions,

only the table (advanced) mode of collection is available. Where the basic mode is available, users can toggle between the two modes (with some restrictions, as described in the following sections) by selecting the appropriate tab.

Collecting in Basic Mode

The Basic mode of collection is designed for rapidly accomplishing collection and configuration activity for an action. You can specify a simple set of rules and continue with the action. You can choose whether to collect just those objects related to the Initially Selected object, All objects, or None for each rule that you define.

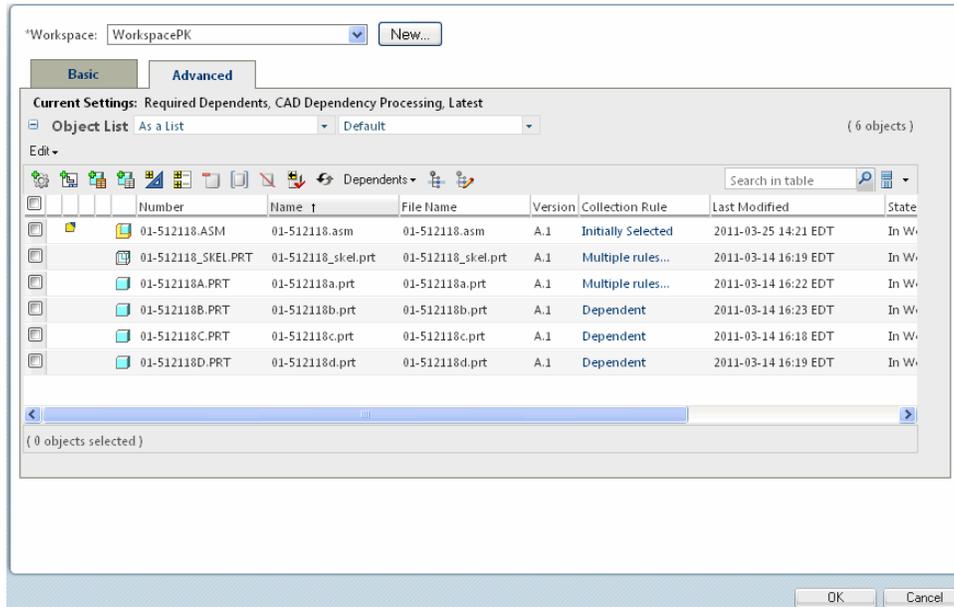
Although you are collecting objects using this mode, the set of collected objects is never displayed on this window, simplifying the collection activity. For more information, refer to the Windchill Help Center, “Collecting in Basic Mode.”

The screenshot shows a dialog box for collecting objects in Basic Mode. At the top, there is a workspace selector set to "WorkspacePK" and a "New..." button. Below this are two tabs: "Basic" (selected) and "Advanced". A "Check Out:" dropdown is set to "None". The main area is titled "Current Settings: Required Dependents, CAD Dependency Processing, Latest" and contains a "Dependencies" section with a "Dependents:" dropdown set to "Required" and a "Limit Expansion Level to:" checkbox. Below this is a "Collect Related Business Objects" section with several dropdown menus: "Parts", "Drawings", "Family Table", and "Generics" (all set to "None"); "CAD / Dynamic Documents" (set to "None"); "Image" (set to "None"); "Source" (set to "None"); and "Published Content" (set to "None"). A small note at the bottom left states "* Indicates Required Fields." At the bottom right are "OK" and "Cancel" buttons.

Collecting in Advanced Mode

The **Advanced** mode of collection is designed for the majority of process scenarios. It allows you to specify collection rules and look for related objects. When you use **Advanced** mode, you will see collected objects displayed in a table. You also use these subsequently collected objects to create new collections.

If you switch from **Basic** mode to **Advanced** mode, the initially selected objects are displayed along with any objects gathered from the collection rules specified in the **Basic** mode.



Note

If both the **Basic** and **Advanced** tabs appear, you can move from **Basic** to **Advanced**, but if you change the configuration criteria or rule criteria, data may be lost. Data can also be lost if you change one or more rules and then move from **Advanced** to **Basic**. A window appears, allowing you to confirm that you want to move to the **Basic** mode or to cancel the action.

For more information, refer to the Windchill Help Center, “Collecting in Advanced Mode.”

Setting an Object Location

In Windchill, a location is a folder (or subfolder) with a context. Objects are assigned their storage locations when first checked in. While a default location is assigned by the workspace configuration specification, during initial check-in you can use the **Set Location**  command to invoke the **Set Location** window to specify a different location. You are not allowed to set a location during subsequent check-ins; however, the **Set Location** window is available during other operations (such as, Move and Save As).

Note

To change the context of an object after it has been checked in to Windchill, you must use the **Move** action.

For more information, refer to the Windchill Help Center, “Setting an Object Location.”

Saving and Uploading Objects

Many CAD data management actions are accessible from both the Creo Parametric and the workspace user interfaces. A newly-created object, however, must first be saved in order to appear in the workspace.

You can save Creo Parametric files using the **Save**, or **Save and Upload** commands. A **Save** command creates a file in a specified directory. Traditionally, this has been a directory on the local file system (your working directory) but with Creo Parametric your working directory can be your workspace in the PDM system. An **Upload** command, which operates in the background, places a previously saved object in your private space on the PDM server, but does not check it in. The **Save and Upload** command performs both operations simultaneously.

Creating CAD Documents with Creo Parametric CAD Data

Saving a model to the workspace creates a CAD document that contains the model file. You can also initiate the CAD document creation from the workspace itself, specifying the CAD document's attributes during the process. For more information, refer to the Windchill Help Center, "Creating a new CAD Document."

Creating Part Structures for CAD Data

Once a CAD document structure has been created, a product structure can be created in Windchill by creating and associating an enterprise part to each CAD document in the CAD document structure, and then checking all the objects into Windchill. Upon check in, the Windchill build rule uses the relationships among the CAD documents to build a product structure relating all the enterprise parts. The term owner association (or owner link) indicates that the association is a primary kind that is recognized by the build rule in creating a structure, passing attributes, and defining a representation. The term content association (or content link) refers to a secondary association (for example, that of a drawing to a part) wherein the CAD document describes the part, but does not need to be included in the product structure. In between these two strengths of association are several 'flavors' of association to help you define the relationships between in the CAD and part structures as best suits your business purposes. For more information, see [Managing Part-CAD Document Relationships on page 74](#).

Associating CAD Documents to Parts

It was mentioned in the section on creating CAD documents that a recommended practice is to associate Windchill parts with CAD documents at the time of creation. However, there are at least two reasons why that practice might not be followed:

-
- The CAD documents were created in the workspace by the **Save and Upload** command in Creo Parametric, not using the **New CAD Document** window in the HTML UI.

In this case, the workspace provides the **Auto Associate** command that allows you to select multiple CAD documents, and then create and associate enterprise parts for those documents with a single click. You can also associate CAD documents to existing parts.

Note

The preference, Operation > Auto Associate > Store New Parts with CAD Documents, when set to "Yes," specifies that the storage location of new part created during Auto Associate be the same as its associated CAD document. By default, the preference is set to "No."

- Enterprise parts intended to correspond to the CAD documents have already been created in Windchill.

In this case, the **Edit Association** command allows you to select a CAD document and then search or browse for the appropriate enterprise part to which to associate it. The **Edit Association** command also allows you to start with a part and find an appropriate CAD document. The use of both commands is detailed in the following sections.

Automatically Associating Parts

The **Auto Associate** command allows you to automatically find and associate an existing part to a CAD document or, if no matching part currently exists, create a new part and associate it to the CAD document. This functionality operates according to several conditions, and is accomplished using the **Auto Associate** page.

Note

The exact manner in which the part is searched for, created, named, and numbered depends on preferences set by a site administrator. For more information, see the section [Customizing Auto Associate](#) on page 140.

Auto Associate Conditions

The autoassociate parts functionality works in accordance with the following conditions:

- For a successful association, the document must be checked out and have no existing associations, and the part must be checked out (autoassociate automatically checks out the found or created part to the workspace).

Documents and associated parts remain checked out to the workspace after association.

- While searching parts, if more than one part per document is returned, then the **Auto Associate** command ignores the document and an error message is shown in the event console.
- In multi-selection, if you select parts and checked in documents along with qualified CAD documents, the parts and checked in CAD documents are ignored by the **Auto Associate** command. If none of the selected objects are valid candidates for the command, then a status message is shown:

```
None of the selected objects is eligible for the 'Auto Associate Parts' action.
```
- If you select a newly created drawing document, the system searches for a model for the drawing in the database. If the model is found, the system creates a Described By link between the part and the drawing, and an owner association between the part and the model. If the model is not found, a message stating this is reported in the event console.

For more information, see the Windchill Help Center, “Automatically Associating Parts and CAD Documents.”

Editing the Associations of CAD Documents and Parts

Initiating the **Edit ► Edit Association** action causes an automatic checkout of the selected CAD document and part objects, if they are not already checked out. In the case of a CAD document, checkout is for meta data only; no content is downloaded during this checkout action.

Note

Any object checked out implicitly remains checked out even if the association fails to check out the other object.

After you associate part with all CAD documents in an assembly, you can see the association; however, Uses links between the parts are not visible until after check in, when the Windchill build rule constructs the Uses links.

It is recommended that you associate a part with a CAD document at the time of object creation, not when you create a structure. A CAD document is said to describe a part to be included in the bill of materials, and in turn a part is described by a CAD document.

To verify the association, view the details page for either object. Selecting the **Related Objects** tab on the information page for a part shows the associated CAD document in the **CAD/Dynamic Documents** table. Selecting the **Related Objects** tab on the information page for a CAD document shows the associated part in the **Parts** table.

Once checked in, the parts you associated to CAD documents have a product structure, which is visible on the **Product Structure** page for the top-level part.

For more information, see the Windchill Help Center, “Editing the Association of CAD Documents and Parts.”

Uploading Objects

You can use the **Upload** command when you want to store your object securely on the PDM server, but still keep it invisible to other users. Once an object has been saved, you can upload it to the server using the Creo Parametric **File > Save and Upload** command, or the **Upload** action from the workspace (if the object has been saved to the workspace).

Performing an Upload from Creo Parametric

The procedure for uploading an object from Creo Parametric is initiated by selecting **File > Save and Upload**. The rest of the procedure is identical to that for a simple save, as described in a previous section. At the end of the procedure, the system uploads the object to your personal area on the PDM server and notifies you that the upload has been successful.

Performing an Upload from the Workspace

Consider the following information about an upload operation:

- Upload is only valid for files that are new or checked out to the workspace and modified. A workspace file which is identical to the server file is not uploaded.
- Selecting a workspace object for upload also selects its new or modified dependent objects to be uploaded. Selecting a family table object also selects other family members, including the generic.

For more information, see the Windchill Help Center, “Uploading Objects from a Workspace Cache.”

Checking In Objects

When you are ready to place a new object into the Windchill database, or you have completed making modifications to the working copy of a checked-out object and are ready to remove your lock on the object, you check the object in to the database.

When an object is checked in, the system assigns it the next iteration. In the case of a newly created object, the system creates the first iteration. The modified object information becomes available to other Windchill users, and the object is available for checkout by others (unless you specify to keep the object checked out after the check-in).

The check-in process can be accomplished in several ways:

- From the Creo Parametric user interface using either auto or custom Check In
- Using the **Check In** page that is accessible from the workspace in the Creo Parametric browser. These different check-in options are explained in the following sections.

Checking In Objects from Creo Parametric

After you have finished working on objects in your workspace, you can share the design changes with other users. The check-in operation copies the information and files associated with all changed objects from the workspace to the Windchill server.

Check In serves several purposes:

- Check In enables other users to access the latest version of the object and to check it out to their workspace.
- If you created a new object in your workspace or opened an object from disk into Creo Parametric and saved it to your workspace, Check In adds the object to the commonspace database for the first time and makes it accessible to other users.

There are two ways to check in an object:

- **Auto Check In**—Checks in objects from your current Creo Parametric session to the Windchill server using default values that you can set in your workspace's preferences. This method of checkin is only available from the Creo Parametric **File > Check In > Auto Check In** menu.
- **Custom Check In**—Enables you to check or change default settings and also provides additional options during the checkin. The custom method is available from both the Creo Parametric **File > Check In > Custom Check In** menu and the workspace user interface.

Performing an Automatic Check In

1. In an active Creo Parametric session, select **File > Check In > Auto Check In**. The name of the file appears in the **Model Name** field of the **Save Object** window.

Alternatively, in assembly mode you can select the object's name in the Model Tree and right-click. A shortcut menu appears. Select **Check In > Auto**. The system uses default settings to check in the object.

2. Accept the default object or enter the name of another object in session. Any file name entered must be unique.

-
3. Click the checkmark button, or press ENTER.

Note

To reject the save operation, click the X.

4. The system uploads the file to the PDM server. The modified file can now be referenced and modified by other PDM users.

Note

After a checkin, if you want to continue to modify the file you need to check it out again; however, if the file is still in your local cache it does not need to be downloaded again.

Note

During Auto Check In, if creation of managed baselines upon checkin has been enabled, by default the name of the baseline is generated in the format "user_ yyyy-dd-mm hh:mm:ss". However, if the property "com.ptc.windchill.uwgm.cadx.checkin.DetailedBaselineName" is set to "true" in wt.properties, the name of the baseline would be generated using the format "user_filename_version.iteration_day_dd_mmm_yyyy_hh_mm_ss" where "filename", "version" and "iteration" are respectively the filename, version identifier and iteration identifier of the top level seed object being checked in.

Performing a Custom Check In

1. In an active Creo Parametric session, click **File** ► **Check In** ► **Custom Check In**. The name of the file appears in the **Model Name** field of the **Save Object** window.

Alternatively, you can select the object's name in the Model Tree and right-click. A shortcut menu appears. Select **Check In** ► **Custom** (if you choose this method, skip directly to step 4).

2. In the **Save Object** window, accept the default object or enter the name of another object in session. Any file name entered must be unique.
3. Click **OK**. The **Custom Check In** window opens.
4. In the **Object Types** area, select one of the following options:
 - **Models**—Checks in the model files.
 - **Viewables**—Checks in files as ProductView viewable files.
 - **Models and Viewables**—Checks in objects as both model files and viewables
5. Click **Ok** in the **Custom Check In** window. The **Check In** page opens in the Creo Parametric browser. See the following section which explains using the **Check In** page.

Checking In from the Workspace User Interface

The **Check In** action presents the **Check In** page on which you can select options specifying which objects and which dependents of the objects to use for actions. The **Check In** page also lets you specify the location in which to store the objects.

Note

Activity in the CAD application session will be blocked until the check-in activity is complete.

For more information, see the Windchill Help Center, “Checking In Objects to Windchill.”

Checking Out Objects

To modify an object, you must perform a checkout operation on the object. The process of checking out communicates your intention to modify a design to the PDM server. The checkout operation ensures that access to objects is appropriate for a multiple user environment. A lock is placed on the object in the database, so that other users can obtain read-only copies of the object but are prevented from modifying the object while you have it checked out. In addition, the checkout process enables you to determine the configuration of the desired objects as well as the workspace in which to modify the objects.

Tip

*When attempting to retrieve baseline configurations during **Check Out** or **Add to Workspace** actions, it is important to keep in mind that checkout places a modifiable copy of a locked commonspace object into the workspace – not the commonspace object itself – and the copy is not a member of the baseline.*

Therefore, to successfully retrieve the baseline configuration of an object, you should select the commonspace version of the object (which you have added, but not checked out to your workspace) as the initially selected object for the action.

During the checkout process, all of the data that defines an object and its relationships is copied to the local workspace. You can specify if the physical files should be copied from the PDM server to the workspace or whether the files should be accessed through a link. Linked files are only retrieved from the PDM server when requested by Creo Parametric. By using linked files, you maintain local copies of only those objects that you have retrieved into a Creo Parametric session after check out. The benefit of checking out using links is quicker check-out transactions because the content files are not downloaded to your workstation. The content files will only be transferred to your workstation when you need them.

Note

If you have multiple workspaces, a checked-out object can only be modified from the workspace where it was originally checked out. The object is inaccessible from any other workspace until it is checked in.

Checking Out Objects from Creo Parametric

When working with a downloaded object in Creo Parametric, the object is strictly read-only. To modify the object, you must first perform a checkout. There are three ways to check out an object in Creo Parametric: from the Creo Parametric menu, from the model tree, and "on-the-fly."

From The Menu

This method uses the **File ► Check Out** command.

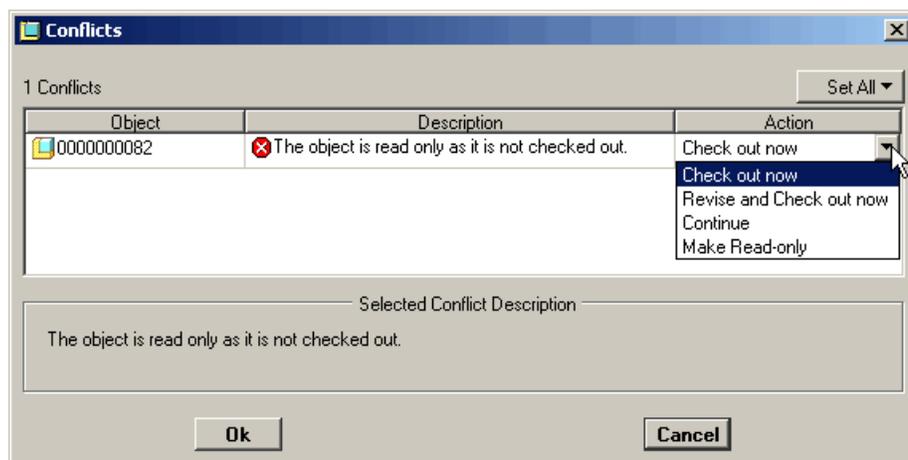
1. In Creo Parametric, click **File > Check Out**. The system prompts you to enter the name of the object that you want to check out.
2. Click the checkmark button to accept the default value. The downloaded object is checked out to your workspace.

From the Model Tree

1. Right-click on the object in the model tree, then select **Check Out**.
2. The object will be checked out as long as you have the proper permission and it is not checked out by someone else (or yourself in another workspace)

Check Out On-The-Fly

The third method is called "checkout on-the-fly". Checkout on-the-fly prompts you to check out a read-only object whenever you attempt to modify it. When you attempt to modify a read-only object, Creo Parametric displays a **Conflicts** window, indicating that "The object is read only as it is not checked out". The suggested action is "Check out now."



Choose **Ok** and the object is checked out (if you have the proper permission and it is not checked out by someone else, or by yourself in another workspace).

Note

Checkout on-the-fly only works with objects downloaded from a primary server.

Checking Out Objects from the Workspace

Use the **Check Out** action to add objects from Windchill to your workspace for modification. Generally, a working copy of the file is transferred to your local disk, and a lock is placed on the object in the database to prevent simultaneous modification in another workspace.

Refer to the section, [Comparison of Download, Link, and Reuse on page 28](#), for an explanation of how content can be handled during a checkout.

A direct checkout (no user interface involved, only initially selected objects checked out) occurs when checkout is initiated from the following places:

- Workspace toolbar check out icon 
- Workspace row level action
- Edit Attributes from the workspace
- Check out row level actions
- Right-mouse-button menu actions
- Save As in workspace toolbar
- Workspace CAD Document Structure Report toolbar
- Creo Parametric **File** menu

For more information on the Check Out page, see the Windchill Help Center, “Checking Out Objects from Windchill.”

Comparison of Download, Link, and Reuse

When checking out or adding objects to your workspace, your choice of download options (download, link, or reuse) should be guided by the current workspace situation and your intent. If your intent is to open the current server-stored model in Creo Parametric at some point in time, the **Download** option should be selected, as it will download the content in a much more efficient manner with fewer performance concerns. Selecting **Download** does, however, overwrite any locally cached modifications, if they exist.

If you select the **Link** option during checkout, file data for that object is not downloaded during the checkout, but it will be downloaded at some subsequent time if it is requested by Creo Parametric. Whether a subsequent download is triggered or not, the server content is identified as the content of the object. Therefore, the link option can be used in scenarios where you want to overwrite (or

have no) locally cached modifications of the CAD document, and have no intention to open the model in Creo Parametric (for example, you might want to modify the model parameters through the Windchill **Edit Attributes** page).

If the file already exists, is modified in your local cache, and you select **Reuse**, content for that object is not downloaded. Instead, the locally modified content becomes the content of the "added" object iteration or working copy. In other words, the reuse option is only applicable if you already have modified object content in your local cache.

Checking Out an Earlier Iteration

It may become desirable to revert to an earlier design for a CAD document. You can use the **Iteration History** of the latest iteration of the object to identify an earlier iteration, navigate to its information page, and initiate a checkout. In general, you can initiate the checkout of a non-latest iteration from the same places you would initiate the checkout of the latest iteration. Exceptions are the row-level check-out actions available in places such as the workspace page, the workspace **Edit Attributes** page, and the **Check In** page.

Note

The checkout of an earlier iteration is not supported for Windchill part objects.

Check out of an earlier iteration of a CAD document is supported, subject to the following conditions:

- You receive a warning that the iteration you are attempting to check out is not the latest.
- If you perform the checkout using the Check Out/Add to Workspace page, a conflict message saying an iteration other than the latest cannot be checked out will be displayed in the **Event Management** utility. This is an overridable conflict that can be overridden using the **Conflict Management** utility.
- No other iteration of the object can be currently checked out. Also, when you check out a non-latest iteration, all other iterations of the object are adorned with a "checked-out" symbol to indicate that no other iterations can be simultaneously checked-out. On the **Iteration History** report, to avoid confusion, only the iteration actually checked out displays the yellow checked-out-by-you symbol (🟡). The iteration which was latest before the non-latest checkout shows a checkmark on a gray background to indicate that it is unavailable for checkout.
- When gathering related objects of an initially selected object that is not the latest iteration, the default configuration is the As Stored configuration for the initially selected object (not Latest). This can be modified by the Windchill preference, Set Configuration for Check Out.

-
- Upon checking the object back in, it becomes the latest iteration. The **Iteration History** report records the earlier iteration it was derived from.
 - When checked out, the object has both the content (only if a CAD document) and meta data of the earlier iteration. When checked in, however, the earlier iteration is assigned the life cycle state of the iteration that had previously been the latest.
 - When the check out of an earlier iteration requires overriding an overridable conflict, you need to explicitly refresh the workspace in order to properly view the earlier iteration.
 - Checking out an earlier iteration of a CAD document to a project is not allowed.
 - Checking out an earlier iteration from a project to a workspace is allowed, so long as:
 - The version is native to project, or is a one-off version checked out to the project.
 - The object is not checked out by any individual.
 - The version is not shared to the project from Windchill PDMLink.

Undoing Check Out

There are multiple reasons for deciding to undo a checkout. You may want to discard the latest changes to an object and return to the version of the object stored in Windchill, or you may simply want to remove the lock that your checkout has placed on the object.

When you undo a checkout, changes you have made to the content and meta-data of the object are discarded and the content as stored in Windchill is downloaded to the workspace. You can choose not to have the Windchill content downloaded.

Note

A Reuse option for the Undo Check Out action is only available with Pro/ENGINEER 4.0 M070 and later releases and Creo Parametric. This option allows you to retain local modifications that have been saved to the workspace upon undoing the checkout. The Reuse option is only available when the Undo Check Out action is initiated from a primary active workspace viewed in the embedded browser. It is not available in standalone mode. Beginning with Pro/ENGINEER 4.0 M110 and later releases and also with Creo Parametric M010 and later releases, the Reuse option applies to objects whose modifications have been uploaded. At earlier release levels, the reuse option is not applicable to objects whose modifications have been uploaded.

Adding Objects to the Workspace

The download action in Creo Parametric and the **Add to Workspace** action in Windchill enable you to bring read-only copies of objects into your workspace. This allows you to examine the object without placing a lock on it. If you attempt to modify such an object, the system will prompt you to check it out.

When using the **Add to Workspace** action in Windchill, you have options as to how content files are handled. Refer to the section, [Comparison of Download, Link, and Reuse on page 28](#), for an explanation of how content can be handled when adding objects to a workspace.

Note

Access to the Add to Workspace action can be managed using Windchill profiles. For more information, see the Windchill Help Center topic, “Managing Profiles.”

Initiating a Download from Creo Parametric

In Creo Parametric, you can use the **File Open** window to browse your workspace and download objects from the PDM server to your session of Creo Parametric.

1. In Creo Parametric, click **File ► Open**. The **File Open** window opens.
2. Use the **Look In** list to select the workspace or commonspace area that you want to browse. Once selected, a workspace’s contents are visible in the file area.
3. Select the object that you want to download and click **Open**. The selected object downloads to Creo Parametric and opens.

Note

Downloaded objects are read-only. To modify a downloaded object, you must first perform a checkout.

Initiating Add to Workspace from the Workspace

Use the **Add** action (**Add to Workspace** when accessed from the commonspace) to update workspace objects, or to add objects from the commonspace to your workspace. The content files associated with the objects can optionally be downloaded to your local disk for read access by you. In the Windchill **Preference Management** utility, you can set preferences for Add to Workspace and Check Out behavior.

Note

You need "Download" access permission to add objects to the workspace. Creo Parametric dependency rules and an object's status (if it is already existing in the workspace) determine which objects are selectable or deselectable for download. In addition, you may be able to download some objects, check out others, or check out only meta data for yet other objects.

For more information, see the Windchill Help Center, “Adding Objects to the Workspace.”

Removing Objects from the Workspace

In the process of using your workspace, you will create, open, and download many objects. Each object that you create, open, or attempt to modify is added to your workspace. After a while, you may find that your workspace has become cluttered with old or unused objects that you want to remove. To remove the unnecessary objects from the workspace, you must initiate the action.

Note

If you remove a checked-out object from your workspace, the check-out is undone, and any local modifications made to the object are lost.

For more information, see the Windchill Help Center, “Removing Objects from the Workspace.”

Keeping Workspace Objects Up-to-Date

If you work in a multi-user environment, you are likely to encounter the situation where changes to product data are made by others while you have the objects in your workspace. In order to manage this dynamic situation, Creo Parametric can notify you of changes in object status and allows you to update selected or any changed workspace objects to be sure that you remain current with the latest server information. Status information regarding whether a workspace object is out-of-date is communicated via three optional status columns, which you can add to your workspace table view. The status columns are the following:

- Out of Date status
- Out of Date with Workspace Configuration status
- Compare status

For more information on these status columns, see the Windchill Help Center, “About Object Status”.

The workspace offers two actions, Update and Synchronize Workspace, that you can use to keep your workspace current. The Update action is used when modifications to an object made by other users or by you in another workspace may cause your current workspace object to become out-of-date as compared to the default CAD document and part configuration specification (defined in the Workspace Preferences). The default workspace configuration specification is the latest iteration on the latest revision.

You may select one or more objects to update (for example, when their status column symbols indicate they are out-of-date). In addition you can select the Update action without preselecting an object (action-object). In this case, the

Update page is automatically populated with any out-of-date objects from the workspace. After you specify how you would like the content handled (for example, whether or not to update, download, or link for download as needed), executing the update adds the latest iteration of an out-of-date object to the workspace. For more information, please see the Windchill Help Center, “Updating Workspace Objects”.

Tip

Use the Creo Parametric configuration option `overwrite_contents_on_update` to control behavior during the Update action from the Creo Parametric user interface. Note that "yes" is the default setting.

- If set to "no," the system does not overwrite the locally modified contents for out-of-date objects, but updates their metadata only
- If set to "yes," the system overwrites the locally modified or out-of-date objects with the ones in the server in addition to updating their metadata

The Synchronize (workspace) action updates the local cache with the latest information for objects already in the workspace (for example, modifications made from a standalone browser) and is described in the section [Refreshing the Cache on page 33](#).

Refreshing the Cache

You can use the **Synchronize** (workspace) action to explicitly refresh all workspace objects that have become stale in the cache due to more recent changes made on the server by another user or by you in a standalone workspace. One example would be if you used a standalone browser (no cache awareness) to modify attributes on a workspace object. An explicit synchronization with the server would be required to communicate those changes to the local cache.

Note

The cache location is defined by the environment variable `PTC_WF_ROOT`, which is the client connector cache. This is where all Windchill Workgroup Manager-related client side information is stored. If the environment variable is not explicitly defined, the default location will be your home directory. By default, cache is not shared between Windchill Workgroup Manager and Creo Parametric. If you are planning to run the Windchill Workgroup Manager and Creo Parametric on the same system and also want to have the cache residing in a non-default location, you must explicitly set the cache (`PTC_WF_ROOT`) for both these applications to different locations. For example, one method of achieving this is to create a startup batch file for each application that defines the `PTC_WF_ROOT` environment variable (cache location) and then start the program.

There are two types of synchronization of cached information with the server: implicit and explicit synchronization.

- Implicit synchronization occurs whenever the client makes a request to the server (Check Out, Upload, and so on), and refreshes information in the following areas:
 - Change to the status "Checked out by you" in this workspace
 - Addition of an object into this workspace
 - Removal of object from this workspace
- Explicit synchronization occurs when you select **Tools ► Synchronize**. Explicit synchronization refreshes information in the following areas:
 - Everything that implicit synchronization synchronizes (mentioned above)
 - Update of File Name in the local cache
 - Update of attributes on the object's master version in the database
 - Change to the status "Checked out by another user", "Checked out by you in another workspace"

For performance considerations, not all statuses are updated upon every invocation of the workspace listing. The following table explains which statuses require explicit synchronization (**Tools ► Synchronize**) of the workspace with server information:

Status Category	Status	Requires Explicit Synchronization?
Share Status	 Shared to a project	Yes
	 Shared from PDM	Yes
	 Checked-out from PDM	Yes
General Status	 Locked (embedded browser only)	N/A (cache-only status)
	 Checked out by you	No
	 Checked out by you in another workspace	Yes
	 New	No
	 Checked Out by another user	Yes
	 Checked-out to a project	Yes
	 Another Iteration is checked out by you	Yes
	 Another Iteration is checked out by another user	Yes

Status Category	Status		Requires Explicit Synchronization?
Modified Status		Modifications Need to be Uploaded (Embedded browser only)	N/A (cache-only status)
		Modifications Uploaded	No (derived from server)
		Modifications Not Eligible for Upload (Embedded browser only)	N/A (cache-only status)
Out of Date Status		Out of Date modified by you	No (dynamically computed on each workspace listing invocation)
		Out of Date modified by other (user name)	No (dynamically computed on each workspace listing invocation)
Out of Date with Workspace Configuration		Out of date with Workspace configuration - Modified by you	No (dynamically computed on each workspace listing invocation)
		Out of date with Workspace configuration - Modified by another user (user name)	No (dynamically computed on each workspace listing invocation)
Compare Status		Critical Error: File name conflict	No
		Error: File name changed	No
		Warning: Removed from Workspace	Yes

To explicitly synchronize the workspace with the server, perform the following procedure:

-
- Select **Tools** ► **Synchronize**.
 - The system updates all of the information about the workspace and the objects in the workspace with the latest changes made on the server by another user or by you in another workspace.

Importing Objects to the Workspace

You can use your active workspace to load CAD objects into your workspace without explicitly retrieving them into Creo Parametric. This functionality is supported only for Creo Parametric CAD documents; which can include file types that are supported by Creo Parametric but cannot be opened directly, such as material files or texture files, which become CAD documents upon import.

Overview of Importing Objects

Import involves invoking the import user interface, initially selecting objects in a source location for import, using collection rules to gather related objects, and, optionally, specifying additional options for how the objects should be handled by the system upon committing the import.

The following are important aspects of import:

- While objects are being imported, Creo Parametric session is frozen.
- Setting a preference allows you to specify that secondary (attached) content is transferred along with the primary file.
- One object can cause the failure of import. Conflicts are reported in the **Event Management** utility.
- If the object already exists in the commonspace or workspace, its status is shown in status columns.
- If a component of an assembly stored in the local file system has been moved to a location other than where the rest of the assembly is stored, the import of the assembly from the original location will show the moved component as an incomplete dependent object (ghost), and the component's file path will be displayed as its former location.

Import creates CAD documents in the target workspace with primary content. An imported object's file name becomes its CAD name. The system tries to establish appropriate dependencies between new and existing CAD documents, if there is an object with the same name already in the target system.

Import supports the following functionality:

-
- Display of the object status and conflicts (provided the object already exists in the commonspace or workspace)
 - Electing either to reuse objects that are already in the commonspace or workspace, or to overwrite with an imported object from a source directory. The system provides default object handling settings by checking object database status or user privileges.
 - Check out of objects upon import if you are importing objects that are already on the server.
 - Reuse of a server version of an object. If that version is not in the workspace, you have an option not to add it to the workspace upon import (by setting a preference).
 - Attaching secondary content from the local directory

Note

If any dependents of an imported object are not imported and not existing in the database, they appear in the workspace as incomplete objects. (regardless of the nature of dependency, required or optional). Any resolution of incomplete objects should be done during a subsequent upload or checkin.

See the Customizing and Administering Creo Parametric chapter of this guide for information about preferences for search paths, automatic download, or allowing attachment of secondary content. For more information on using the workspace import user interface see the Windchill Help Center, “Importing to the Workspace.”

Exporting Objects from the Workspace

You can use your active workspace to load CAD objects out of your workspace without explicitly retrieving them into Creo Parametric. This functionality is supported only for Creo Parametric CAD documents; which can include file types that are supported by Creo Parametric but cannot be opened directly, such as material files or texture files, which become CAD documents upon import.

Overview of Exporting Objects

Export is accomplished by initially selecting CAD objects in your active workspace (embedded browser only), using collection rules to add related objects to the set, defining a target directory for the exported objects, and specifying additional options for how the objects should be handled by the system upon committing the export.

The following are important aspects of export:

-
- While objects are being exported, Creo Parametric session is frozen.
 - Setting a preference allows you to specify that secondary (attached) content is transferred along with the primary file.
 - One object can cause the failure of entire export. Conflicts are reported in the **Event Management** utility.
 - If the object already exists in the commonspace or workspace, its status is shown in status columns.

Export supports the following functionalities:

- Dependency processing.
- Target location setting
- Optionally confirming the list of objects to be exported
- Ability to also attach secondary content from the local directory (by setting a preference)
- Automatic download of objects not currently in the workspace (by setting a preference).
- Electing whether to overwrite or reuse objects that already exist in the target location

To export objects from a workspace, you can use either the **Basic** mode or the **Advanced** mode of collection. In the **Basic** mode you can specify rules that determine what set of objects are to be included in the export in addition to the initially selected objects. In the **Advanced** mode, you can use collection and configuration tools to incrementally add or remove objects to or from a table listing of objects, prior to committing the action. You can move from one tab to the other; however, returning to the **Basic** tab may remove objects you have collected while using the **Advanced** tab.

For more information, see the Windchill Help Center, “Exporting from the Workspace.”

Revising Workspace Objects

You can create a new revision for an object by assigning the next revision level available in a revision scheme. The revision scheme represents a sequence of characters identifying subsequent versions of a revisable object. Creating a new revision of an object results in the object, and all objects you choose to associate with it, being incremented to the set revision level when the revision operation completes. When you revise an object, the latest version of that object is used as the content for the new revision.

Use the **Revise** action to create a new revision of an object. Typically, this is done to initiate a new branch of the design, based on the current object. You can revise objects that are checked in to the database, or checked out by others.

Note

Checked-out objects cannot be revised. Revise is not available for objects in a project context.

For more information, see the Windchill Help Center, “Revising Workspace Objects.”

Setting a Revision

If enabled by a server-side preference, the **Set Revision** window allows you to select a specific revision level for the object(s) you are revising. Your ability to select a revision level can depend on the relative revision levels of parts and CAD documents, the nature of any associations, and the revision scheme for the object. Details on the factors affecting your ability to set a revision level are explained in the chapter, Customizing and Administering Creo Parametric with Windchill, in this guide.

For more information, see the Windchill Help Center, “Setting a Revision.”

Using the Event Management Utility

Many transactions between Creo Parametric and a Windchill server happen asynchronously, allowing you to initiate an operation (such as Check In or Upload) and continue working in Creo Parametric while the operation is processed.

The Event Management utility provides a way for you to check and act on log messages generated in your Creo Parametric or Windchill sessions. It can be accessed from Creo Parametric by clicking on the console status icon in the status bar. In addition, by selecting **Tools ► Console** in the Creo Parametric user interface, you can access the console of any server to which you are connected. From a workspace, the Event Management utility is accessed by selecting **Event Management** from the workspace **Pick an Action** menu. In Windchill PDMLink and Windchill ProjectLink you can also access it by selecting **Quick Links ► Event Management**.

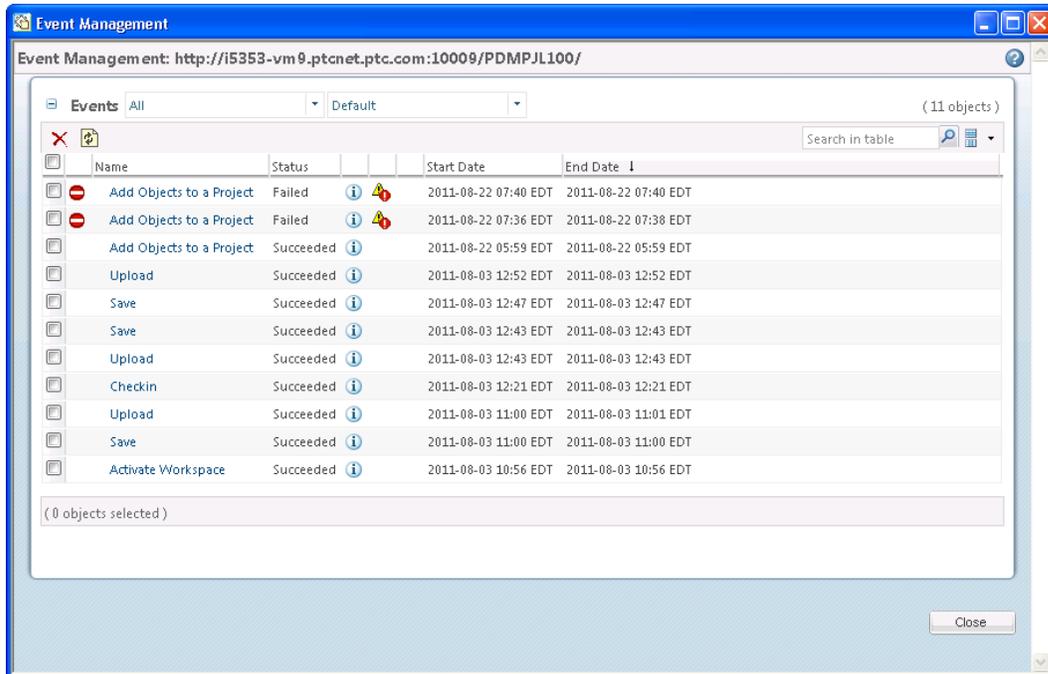
The **Event Management** utility launches automatically in the case of a failed transaction attempt that is initiated from the workspace or PDM system. In the case of a failed transaction that is initiated from the Creo Parametric user interface, an event manager status icon appears in the Creo Parametric status bar.

Using the Event Management utility is particularly useful when performing transactions with a large number of objects or when working in a multi-server environment, because you can access information specific to a server.

The Event Management utility consists of three interlinked windows that allow you view, get information on, and resolve conflicts arising from PDM transactions. The three pages are the **Event Management** page, the **Event Information** page, and the **Conflict Management** page.

Event Management Page

In the **Event Management: <server>** page you can see the PDM events for the named server listed chronologically in rows in the **Events** table. You can access detailed information about a particular event in the **Event Information** page by clicking  in the event's Actions column, and access the **Conflict Management** page by clicking icons displayed in the Actions column for viewing or resolving conflicts.



The **Events** table also has a tool bar with commands for the following actions:

- **Delete**  – Deletes selected row(s) from the table
- **Refresh**  – Refreshes the event listing in the table

Event Types

The type column in the **Event Management** page contains icons that identify the particular type of event listed. The event type icons are described as follows:

-  – Overridable type conflict
-  – Non-overridable type conflict (Failed)
-  – Warning
-  – In Progress
-  – Pending

-  – Retried
-  – On Hold

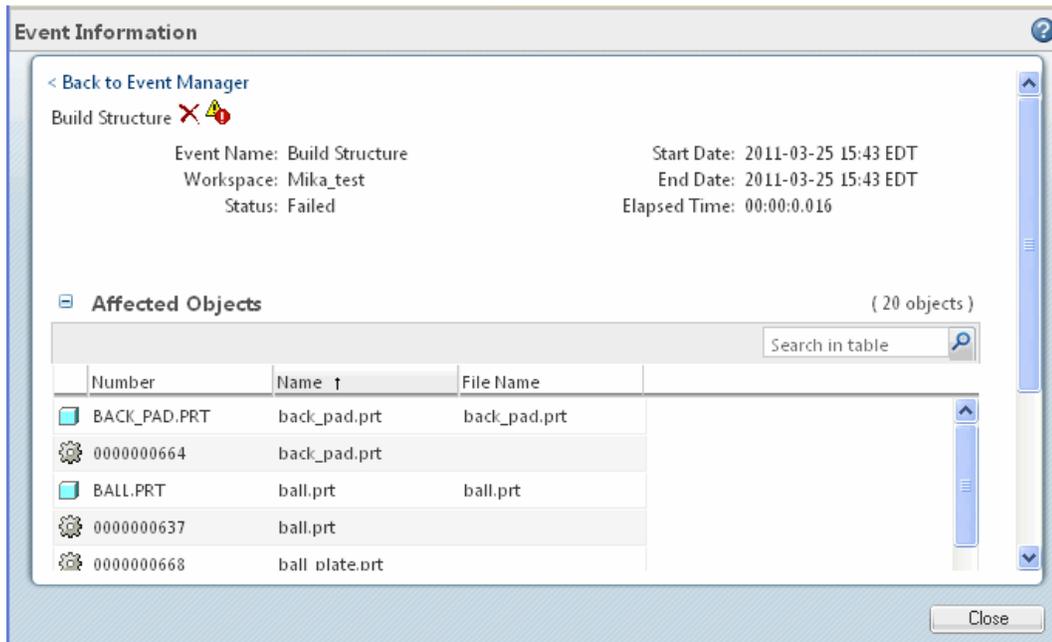
Event Actions

The Actions column in the **Event Management** page contains icons that can call either the **Event Information** page or the **Conflict Management** page, and are as follows:

-  – View event information in the **Event Information** page
-  – View warnings or errors in the **Conflict Management** page
-  – Resolve Conflicts in the **Conflict Management** page

Event Information Page

The **Event Information** page is accessed from the **Event Management** utility by clicking  in the Actions column for the event in the **Events** list, or by selecting the link in the event name.



Event Information

[< Back to Event Manager](#)

Build Structure  

Event Name: Build Structure Start Date: 2011-03-25 15:43 EDT
 Workspace: Mika_test End Date: 2011-03-25 15:43 EDT
 Status: Failed Elapsed Time: 00:00:0.016

Affected Objects (20 objects)

Search in table 

Number	Name ↑	File Name
 BACK_PAD.PRT	back_pad.prt	back_pad.prt
 0000000664	back_pad.prt	
 BALL.PRT	ball.prt	ball.prt
 0000000637	ball.prt	
 0000000668	ball_plate.prt	

Close

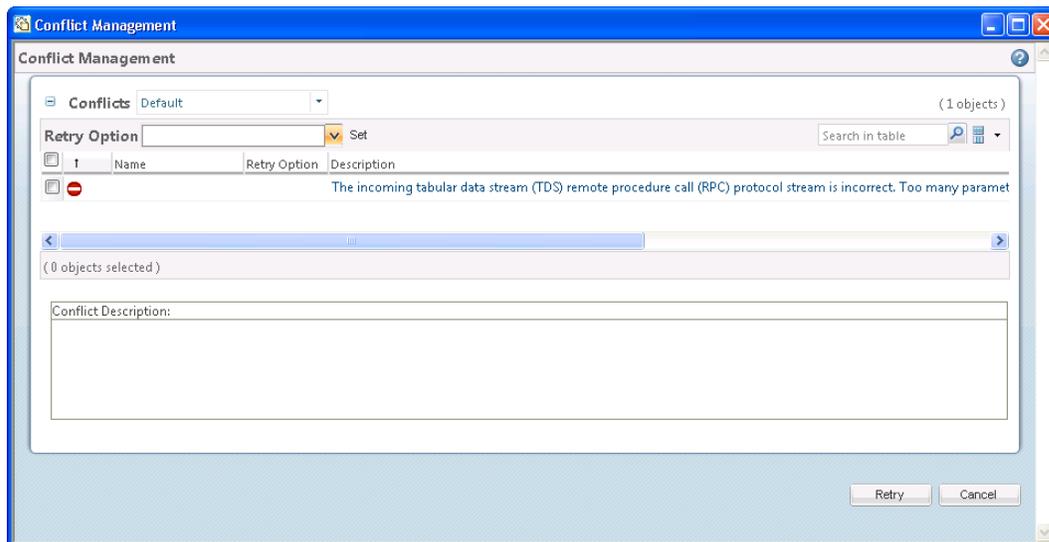
At the top of the page is a hyperlink that returns you to the **Event List** in the **Event Management** utility. The area immediately below the hyperlink lists the event attributes, as follows:

- **Event Name** – The name of the event. Next to the event name are icons for the Delete action () and, if warnings or conflicts occurred, one of the following actions, as applicable to the type of event:

- **View Warning**  – If the event type is a warning
- **View Conflicts**  – If the event type is a non-overridable conflict
- **Resolve Conflicts**  – If the event type is an overridable conflict
- **Workspace** – The workspace from which the event originated
- **Status** – The status of the event, if completed, or a progress bar indicating how close the event is to completion
- **Start Date** – The date and time the event began
- **End Date** – The date and time the event was completed
- **Elapsed Time** – The total time required to complete the event (if completed)

Conflict Management Page

The **Conflict Management** page assists you in viewing and resolving conflicts that arise from PDM events. It is accessed from the **Event Management** page or the **Event Information** page by clicking the view conflicts icon  or the resolve conflicts icon .



In the toolbar of the **Conflicts** table is a menu that lists the **Retry Options** common to all of the selected conflicts (if any are available). The **Description** column in the **Conflicts** table describes the conflict. If the description is truncated, the full text can be displayed in the **Conflict Description** area below the table by clicking on the ellipsis (...) in the **Description** column.

To resolve an overridable conflict, perform the following procedure:

1. Select one or more rows containing an overridable conflict.
2. Select an option from the **Retry Options** menu in the table toolbar.

3. Click **Set**.

The **Conflicts** table refreshes to display the new value in the **Retry Option** column for the selected row(s).

4. Click **OK**. The action is retried and a new entry is created in the **Events** list. The original event's status changes to **Retried**. Further access to the original event in the **Conflicts Management** page is read-only.

2

Advanced Object Handling in Windchill

Modifying Object Attributes (Properties).....	46
Renaming Objects	48
Deriving New Designs Using Save As	49
Opening Objects in Creo Parametric	53
Working with Family Tables	55
CAD Document Templates and Creo Parametric Start Parts.....	66
Using Library Parts	68
Managing Incomplete Dependent Objects	69
Simplified Representations.....	72
Managing Model Items.....	74
Managing Part-CAD Document Relationships.....	74
Verifying Windchill Editing Instructions.....	88
Heterogeneous Design	88
Working with Configurable CAD Documents	90

This chapter describes how to perform more advanced PDM activities and explains how Windchill handles some Creo Parametric objects, for example, family tables and simplified representations.

Modifying Object Attributes (Properties)

Your design work can sometimes call for the addition or removal of attributes (also referred to as properties) on objects, or the modification of attribute values. The **Edit Attributes** window provides a means for you to:

- Edit an attribute shared by multiple objects, all at the same time
- Edit multiple attributes on a single object

If you edit attributes of workspace objects, then those modifications apply only to the workspace version of the objects. Alternatively, you can initiate attribute modification on the commonspace version of an object (initiating **Edit Attributes** from the commonspace view of the object's information page). In this case the attribute modification applies to all versions of the object.

Attributes on Family Table objects can be modified, subject to certain restrictions. For more information on modifying Family Table attributes, see [Modifying Family Table Attributes in Windchill on page 62](#).

Editing Attributes from the Workspace

User-defined attributes, referred to in some authoring applications as parameters or properties, can be edited on one or more objects in the workspace. You can also edit attributes from an object's information page.

For more information, see the Windchill Help Center, "Editing Attributes."

Note

*If you cannot see the attribute you wish to edit, add a column to the **Object List** table that displays the attribute, using the **Customize** option in the **Current View** drop-down list.*

Note

Location is only editable for objects that have never been checked in.

Note

*If you have entered a location value that is in a context in which you do not have authorization to create CAD documents, then you are returned to the **Edit Attributes** window. The incorrect values are removed, and you see a warning icon in the **Status** column. Additionally, the message area displays a message indicating that you are not authorized to check in to the specified context.*

Editing Attributes from the Information Page

When you want to edit attributes from the information page of an object, select **Edit Attributes** from the action drop-down list.

Attributes edited through the commonspace view of an object's information page apply to all versions of the object. Editing attributes for all versions of an object is only available via the commonspace view of the information page.

Setting Attribute Values

The **Edit Attribute Value** window, called by the **Set Value** action on the **Edit Attributes** page, allows you to modify any of the editable displayed attributes, except location, of objects selected or the action. Attribute values are only editable for checked out objects. If you select both checked out and checked in objects and then click **Set Value**, the **Edit Attribute Value** window is opened; but changes made in it will only apply to the selected objects that have been checked out.

Similarly, if you select a family table instance, and the instance has a generic that is not checked out in the current workspace, then the **Edit Attribute Value** window is launched; but changes made in it will only apply to the valid set of objects.

In the **Edit Attribute Value** window, the **Set** menu contains a list of object attributes currently displayed in the columns of the **Edit Attributes** page. Depending on the attribute selected in the **Set** drop-down menu, the **To** field presents a menu (for list-specified values) or input panel (for range-delimited values). Any default value for the attribute is displayed when the **Set** selection is first made.

For more information, see the Windchill Help Center, “Setting Attribute Values.”

Updating Attribute Values in Creo Parametric and Windchill

If you have an owner-associated CAD document and part, with the same attribute assigned to both, there can be differences in how the attribute value is updated, depending on the application from which the modification of the value is made.

If you have a checked-in part with a unique value for the attribute, and you subsequently check out the CAD document, modify (in Creo Parametric) the parameter that drives the attribute and execute auto-checkin, the part's attribute value is modified to match the CAD document's parameter value.

However, the opposite is not true. If you check out a part to the workspace and modify its attribute value using the **Edit Attributes** page and then check it back in, the value of the CAD document's attribute (the CAD document is never checked out) is not updated to reflect the the part attribute's new value.

Renaming Objects

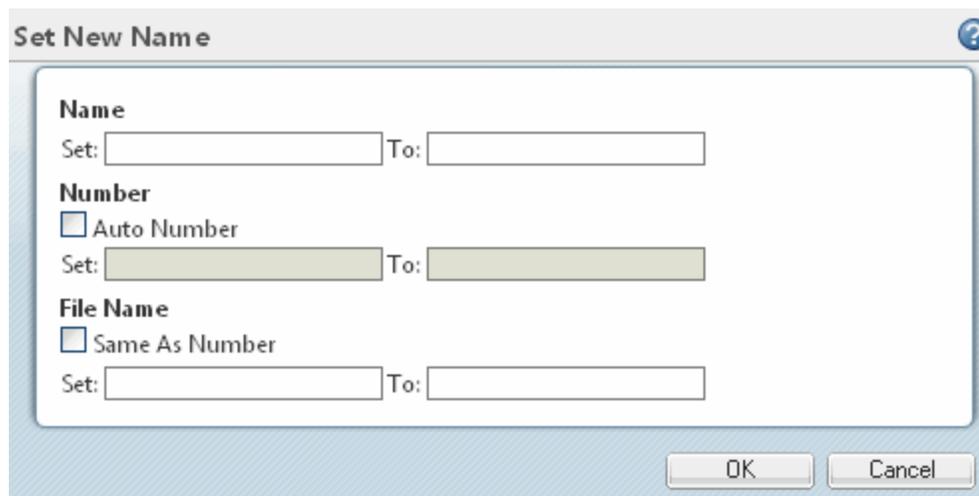
You can rename an object, changing the values of Number, Name, and Model Name attributes, if you have Modify Identity access permission. You can even rename objects that are in another's workspace; the system notifies the other user that they need to synchronize their workspace with updated information on the server.

Only objects that have never been checked in (new) can be renamed from the workspace. Once objects have been checked into the commonspace, they can no longer be renamed from the workspace and Rename must be accessed from the commonspace.

For more information, see the Windchill Help Center, “Renaming Objects.”

Setting a New Name

The **Set New Name** window is invoked by clicking **Set New Name** on the **Rename** page (or on the **Save As** page). It allows you to set conventions for naming and numbering objects (or new copies of objects).



The screenshot shows a dialog box titled "Set New Name". It has a title bar with a question mark icon. The dialog is divided into three sections: "Name", "Number", and "File Name". Each section has a "Set:" and a "To:" text box. In the "Number" section, there is a checkbox for "Auto Number". In the "File Name" section, there is a checkbox for "Same As Number". At the bottom right of the dialog are "OK" and "Cancel" buttons.

The **Set New Name** window displays the original object attribute value in the **Set** field and the default attribute value for the new object in the **To** field. Depending on site settings you may be able to modify default values, which are as follows:

- In the **Number** area:
 - If auto-numbering is set, the **Auto Number** check box is selected by default, and the **Number Set** and **To** fields, as well as the **New Number** field on the **Save As** page are inactive
 - If you clear the **Auto Number** check box, both controls are activated.

- If auto-numbering is not set, the check box is clear, and the **Number Set** and **To** fields, as well as the **New Number** field on the **Save As** page are active.
- In the **File Name** area:
 - The **Same as Number** check box is checked by default, and the **New File Name Set** and **To** fields, as well as the **New File Name** field on the **Save As** page are inactive. The new file name is identical to the new number.
 - If you clear the **Same As Number** check box, both controls are activated.

To specify your own conventions for naming and numbering objects, use the following procedure:

1. Enter the object name or number in the appropriate **Set** field (you can use wild card matching).
2. Edit the **To** field to your specification (See the following table for examples).
3. Click **OK** to close the **Set New Name** window and have your naming conventions applied to the objects listed on the **Rename** (or **Save As**) page.

Rename Objective	Original Value	Target Value	Format Set	Format To
Apply Prefix	Object.prt	Prefix_Object.prt	*	Prefix_*
Apply Suffix	Object.prt	Object.prt_Suffix	*	*_Suffix
Apply Suffix	Object.prt	Object_Suffix.prt	*.*	*_Suffix.*
Replace	Object.prt	New.prt	*Object*	*New*

Deriving New Designs Using Save As

When you create a new CAD part or product structure, you can save time by using an existing part structure as the starting point from which you derive a new design. The **Save As** action provides a way copy single objects or multiple objects, or an entire structure, and rename them as new objects.

About Using Save As

The **Save As** action applies to parts and end items, CAD parts and assemblies, drawings, and Creo Parametric family table objects.

When you select CAD documents and parts to copy within a product structure, you create new objects (copies with new names). You also have the ability to not copy some of the members of the structure, instead creating references to the original existing objects (reusing original objects in the new structure).

When using the **Save As** action, remember these guidelines:

-
- If you copy a drawing without copying the part or assembly referenced by the drawing, the new drawing references the original part or assembly.
 - You can use Save As to copy an entire or partial family table or any individual member. If you copy an instance without its generic, the new instance copy becomes a new member of the family table

The Save As command is available in the following places:

- Where workspace objects are accessed, including:
 - The workspace **File** menu (**File ▶ Save As**)
 - The workspace information page **Actions** list
 - The toolbar of the workspace version **Model Structure Report**

Note

As opposed to using Save As from the commonspace, workspace Save As allows you to manipulate and modify your newly saved-as objects before committing them to the commonspace with a checkin.

- In the commonspace, including:
 - The actions list on the information page of a checked-in (commspace) CAD document or part object
 - The toolbar on the **Product Structure** table (on the information page of a checked-in part or end item)
 - The toolbar on the **Model Structure Report** table (on the information page of a checked-in CAD document)
 - In addition, the commonspace Save As action is available from search results and from the **Folders** page for products and libraries – from the toolbar and from the object action menus for objects that can be copied.
 - The Save As action is not on the **Folders** toolbar when the current view is set to any of the following:
 - ◆ Folders Only
 - ◆ Links Only
 - ◆ Documents Only

Overview of Save As

Objects selected for **Save As** are initially shown in the **Save As** table as intended to be copied.

Note

The preference, Save As > Save Selected Objects Only, is set to false by default. When this preference is set to true, only selected objects are copied and rest are reused by default.

Some general considerations for (workspace) **Save As** are as follows:

- If CAD documents and their associated parts (either by Owner or Content links) are copied together, the new objects have the same types of associations as the originals.
- If a CAD document is copied without its associated part(s), the new CAD document does not have any associations to parts.
- If a part is copied without its associated CAD document(s), the new part does not have associations to any CAD document.
- Incomplete objects or objects modified in the local cache are not eligible for collection. Objects with circular dependencies must be either included or excluded together.

The following are other important points about how Save As works:

- Selected objects display a default **Number**. CAD objects also have a default **Name**.
 - If the selected object is auto-numbered by default by a site preference, then by default the **New Number** field displays the text (Generated) and the **New File Name** field displays the text <Same As Number> and both fields are inactive. This is to indicate that the object is being copied, and that auto-numbering is applicable for that object type.
 - Selecting an object and clicking **Reuse** indicates that you intend to reuse the existing object in the new structure, rather than creating a new copy. If you toggle between saving a new copy and reusing the existing object, then any text, whether generated by the system or entered by you, is removed from the **New Number**, **New Name**, and **New File Name** (if applicable) fields.
 - An underscore (_) is added to the default object **New Number** when auto-numbering is not the default mechanism for naming for this object type.
 - An underscore (_) is added to the default object **Name**.
 - Windchill generates the default **New File Name** when the file name is tied to the CAD document **New Number**. The file name is the <number> plus the appropriate CAD document extension. For example, if auto-numbering specifies a number for a CAD document to be "1234567", and you are copying a Creo Parametric part, then the resulting **New File Name** is "1234567.prt."

-
- An underscore (_) is added to the base of the default **New File Name** when auto-numbering is not invoked. That is, if the file name was bolt.prt, then when auto-numbering is not used, the **New File Name** by default would be bolt_.prt.
 - Related objects may be added to the object list by using the collection tools and **Configuration** menu.

If you wish to change the default name, number or file name of these added objects, you must do one of the following:

- Enter a **New Number**, **New Name**, or **New File Name** in the appropriate field of the table.
- or
- Select the object and click **Set New Name** in the tool bar. The object Name is changed based on the options set in the **Set New Name** window.

Note

*Changing context, location, or organization ID does not generate a new object. These changes are ignored when exiting the window if the object **Name**, **Number**, or **File Name** has not been changed.*

- You can also use the **Set View** command to specify a different view for a part object.

The default behavior of **Save As** is to preserve CAD document and Windchill part structures to the maximum extent possible.

If you save a CAD document or Windchill part structure, a new, complete, and parallel structure is created only if both of the following conditions are met:

- The top-level object is saved as a new object.
- All the parents of a lower-level object are also saved as new objects.

If these conditions are not met, the explicitly selected objects are copied, but each change is discrete.

Note

*You can use **Save As** to copy an entire or partial family table. If you copy an instance without its generic, the new instance copy becomes a new member of the family table. Also, if you copy a drawing without copying its referent model, the new drawing refers to the original model.*

Using Workspace Save As

The workspace **Save As** action allows you to save objects in the workspace as new objects. These newly created objects are not committed to the commonspace until you perform a checkin; so copies can be modified, evaluated, and, if best, discarded before checkin, to avoid unwanted or redundant database iterations.

Workspace Save As also allows you to update a dependency from an existing parent object to the newly-duplicated object, letting you replace an old component with a newly-duplicated component without involving the authoring application. In addition, circular dependencies can be detected and managed.

For more information, see the Windchill Help Center, “Using Workspace Save As.”

Using Commonsense Save As

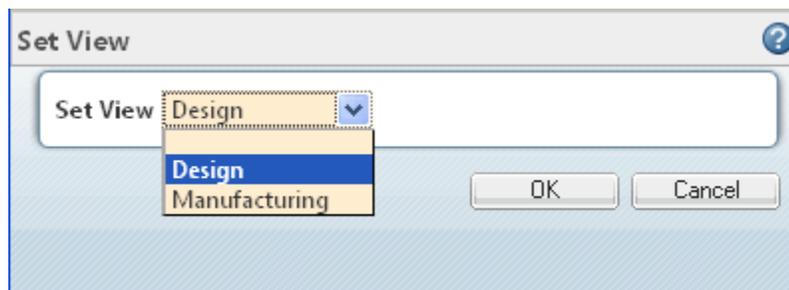
The commonsense **Save As** action allows you to copy a checked-in CAD document or CAD document structure (with or without associated part(s)), or a part object or product structure, and store it as new object(s). This functionality applies to models, drawings, and family table objects in the commonsense, and is capable of preserving CAD document/part associations (that is, the saved as CAD documents and parts are associated in the same way that their originals were).

Commonsense **Save As** is only available from the commonsense (for example, the **Model Structure Report** or the **Product Structure** report on the commonsense view of the object's information page, as well as the part or CAD document **Actions** menu on the commonsense view of an object's information page, and the search results page).

For more information, see the Windchill Help Center, “Saving Commonsense Objects as New Objects.”

Setting a View

The **Set View** window is used to specify a view of a part object during the Save As action.



To set a view for the part object, select a named view from the **Set View** menu.

Opening Objects in Creo Parametric

You can open CAD documents from a workspace, or from various places in the commonsense while working in the embedded browser of a Creo Parametric session. In addition, you can open CAD documents from a standalone browser if an appropriate installation of Creo Parametric is installed on your machine.

Opening Workspace Objects from the Embedded Browser

To open a listed workspace CAD document in Creo Parametric, select **File ► Open inCreo Parametric** or click the open in Creo Parametric icon  in the **Actions** column for the object. The object is opened in your current Creo Parametric session. You can also access the Open in Creo Parametric action from most places where CAD documents are exposed in Windchill, as explained in the following section.

Note

You cannot open CAD documents for UDFs, incomplete objects, or any file type for which direct retrieval is not allowed in Creo Parametric.

Opening Objects from a Standalone Browser

When you are working in a standalone browser and an appropriate version of Creo Parametric is installed on your machine, you can open CAD documents in Creo Parametric. If a Creo Parametric session is already running, the object is opened in your existing session. If no session is running, the action to open an object automatically launches Creo Parametric.

The action for opening CAD documents is available in Windchill generally where CAD documents are exposed. In addition to the workspace, these places include the Folder page (when CAD documents are displayed), the search results page, the CAD document Structure tab, the Product Structure tab (when associated CAD documents are displayed), and the CAD document information page. The action can be initiated either by clicking the open in Creo Parametric icon  in the **Actions** column for the object, or selecting **Open in Creo Parametric** from an actions menu.

System Responses in a Standalone Server Environment

When you initiate the **Open in Creo Parametric** action, the system is either able to directly open the objects in an existing session (for example if you already have a primary server registered with an active workspace), or helps you register and activate the server/workspace required. The following are some general characteristics of the process:

- The specific iteration you select is opened in Creo Parametric.
- Other objects that are required to open your selected object are also collected. The collected objects represent the latest configuration, unless:

-
- You have downloaded a specific configuration to your workspace or the workspaces own configuration identifies a specific configuration (for example, a baseline)
 - You are downloading an object and launch Creo Parametric from the control in the **Check Out** or **Add to Workspace** page.
 - If the registered server you initiate the action from is not primary you are asked if you want to make it primary.
 - If you initiate the action from a workspace that is not active, you are asked to activate the workspace (and warned that any objects in session are erased by the change of workspace)
 - If no server is registered, you are presented with a server registration window.
 - If you have existing workspaces on the server, you are asked to select one.
 - If you have no existing workspaces on the server, a default workspace is created for you.
 - If you have more than one available startup configuration (.psf file) for launching Creo Parametric, you are asked to select one.
 - If the action requires a different workspace to be activated, you are warned that objects currently in session will be erased.

Working with Family Tables

The following sections describe Family Tables and how to work with them in the Windchill PDM system.

Family Table Overview

A Family Table is a means to define a collection of CAD parts (or assemblies, or user defined features) that share the same generic properties, but deviate slightly in one or two aspects, such as size or detail features. Each member of the family (for example, a particular size of a family of similarly designed wood screws) occupies a row in the table, while attributes that are either shared or differentiated among the members of the family appear as the table column headings. Table field cells contain the object values for each member of the family.

CAD parts (or assemblies) in Family Tables are also known as table-driven parts.

Using Family Tables, you can:

- Create and store large numbers of objects simply and compactly
- Save time and effort by standardizing model generation
- Generate variations of a CAD part or assembly from one file without having to re-create and generate each one

-
- Automatically create variations and configurations of the design
 - Create a table of CAD parts that can be saved to a print file and included in CAD part catalogs

Family Tables promote the use of standardized components. They let you represent your actual part inventory in Creo Parametric. Moreover, families make it easy to interchange CAD parts and subassemblies in an assembly, because instances from the same family are automatically interchangeable with each other.

Tip

Because modifications to generics are inherited by instances, recommended practice is to avoid using generics as assembly components.

Family Table Structure

Family Tables are essentially spreadsheets, consisting of columns and rows. It is possible for any instance of a Family Table to also have its own Family Table, thus creating Family Table trees, also referred to as nested Family Tables.

Family Tables consist of the following three components:

1. The base object (generic object or generic) upon which definitions of all other members (instances) of the family are based.
2. Any attributes that may vary between the instances and the generic – dimension and parameter values, features and assembly components to be table-driven, user-defined feature and pattern table names, geometry tolerance and other types.
3. Names of all family members (instances) created by the table and the corresponding values for each of the table-driven objects.

Each row contains the instance name and corresponding values of the attributes in it; columns are used for attributes.

The column headings include the names of all of the dimensions, parameters, features, members, and groups that were selected for the table. Dimensions are listed by name (for example, d9) with the associated symbol name (if any) on the line below it (for example, depth). Parameters are listed by name. Features are listed by feature number (for example F107) with the associated feature type (for example [cut]) or feature name on the line below it.

The generic model is in the first row in the table. Only modifying the actual CAD part, or suppressing or resuming features, can change the generic's table entries – you cannot change the generic model by editing its entries in the Family Tables.

Note

Family Table names are not case-sensitive.

For each instance, you can define whether a feature, parameter, or assembly name is used in the instance either by indicating whether it is present in the instance (Y or N) or by providing a numeric value (in the case of a dimension). All dimension cells must have a value, either a number or an asterisk (*) which indicates that the generic's value is to be used.

All the family members automatically share all aspects of the generic model that are not included in the Family Table. For example, if the generic model has a parameter called Material with a value Steel, all instances will have the same parameter and value.

You can scroll horizontally through a Family Table to see additional information. The **Instance Name** column remains visible as you scroll.

PDM Activities with Family Tables

The following sections explain how to perform PDM operations with Family Tables.

Viewing Family Tables in Windchill

Windchill offers enhanced display options when viewing family tables. The enhanced view allows you to see:

- Family table hierarchy
- Verification status
- Name and Number
- Attributes

Note

Family tables in the latest release use a richer internal data set than in earlier releases. Therefore, migrated family table objects need to be saved & uploaded in Windchill 9.0 in order to view their internal data and verification status correctly.

To display a family table in Windchill:

1. In the workspace, select a family table object (either an instance or the generic).
2. Click the information action. The information page for the selected object opens.
3. On the information page, select **New Tab**. When the new tab appears, select **Customize ► Related Objects ► Family**. This places the Family Tree table on your new tab, displaying the object and its related family members. A selected object is indicated in the tree by a check in the first column of its row.

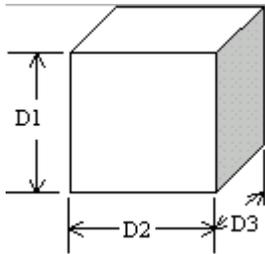
Family Tree Default (3 objects)

Model Name	Verification Status	Number	Name	Version	DESCRIPTION	COLOR
000006080....		000006080.PRT	box	A.1	My Box	Blue
000006081....	Verified	000006081.PRT	box_INST	A.1	My Box	Red
000006082....	Verified	000006082.PRT	box_INST1	A.1	My Box	Yellow

Uploading and Checking In a Simple Family Table

Consider a simple Family Table, based on the generic CAD part, box.prt. It has three major components, as follows:

- Geometric parameters:



- Non-geometric parameters

Name	Type	Value	Designate	Access	Source
DESC	String	My Box	Y	Full	User-Defined
COLOR	String	BLUE	Y	Restricted	User-Defined
WEIGHT	Real	27.000	N	Locked	Relation

- The family table that defines the members of the family

Type	Inst. Name	D1	D2	COLOR
	BOX	3.0	3.0	BLUE
	BOX_1	1.0	3.0	RED
	BOX_2	2.0	2.0	YELLOW

Uploading the Box family table to Windchill results in the creation of the following CAD document objects in the PDM database:

Icon	Model Name	Content File	Attributes	
			COLOR	DESCRIPTION
	Box.prt	box.prt	BLUE	My Box
	Box_1.prt	box_1.prt	RED	My Box
	Box_2.prt	box_2.prt	YELLOW	My Box

Note

In the Icon column of the preceding table of CAD documents, the symbols with the highlighted table row indicate that the document is an instance (no highlighting in the symbol for a generic). During the upload, designated Creo Parametric parameters generate analogous attributes of the CAD document objects created in Windchill, only if:

- An attribute definition exists for an attribute of the same name as a designated parameter
- or
- An explicit mapping between a designated parameter and an existing attribute definition exists on the server.

Note

The upload process includes non-verified (non-regenerated) instances. The following series of graphics show the information for the Box Family Table members after upload:

The screenshot shows the Windchill interface for a CAD Part document. The title bar indicates the document is 'CAD Part - 000006080.prt'. The 'Details' tab is active, showing 'Visualization and Attributes' and 'More Attributes' sections. A 3D model of a box is displayed on the left. The 'Visualization and Attributes' section lists the following information:

File Name:	000006080.prt
Number:	000006080.PRT
Name:	box
Status:	
Modified By:	
Last Modified:	2011-03-22 15:39 EDT

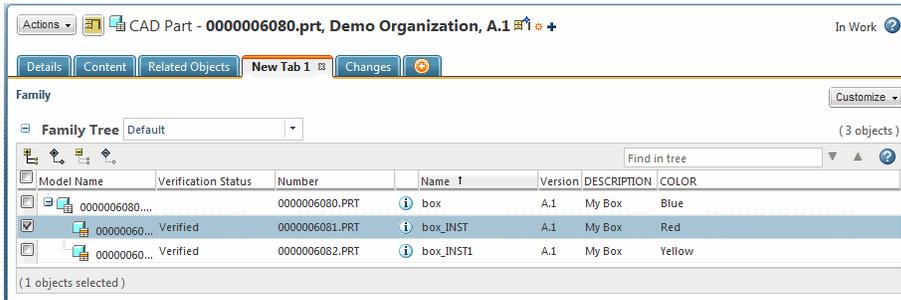
The 'General' section contains the following information:

Version:		Generic:	Yes
Authoring Application:	Creo	Instance:	No
Type:		Missing Dependents:	No
Document Category:	CAD Part	Incomplete Object:	No
Document Subcategory:		Description:	
		Checkin Comments:	

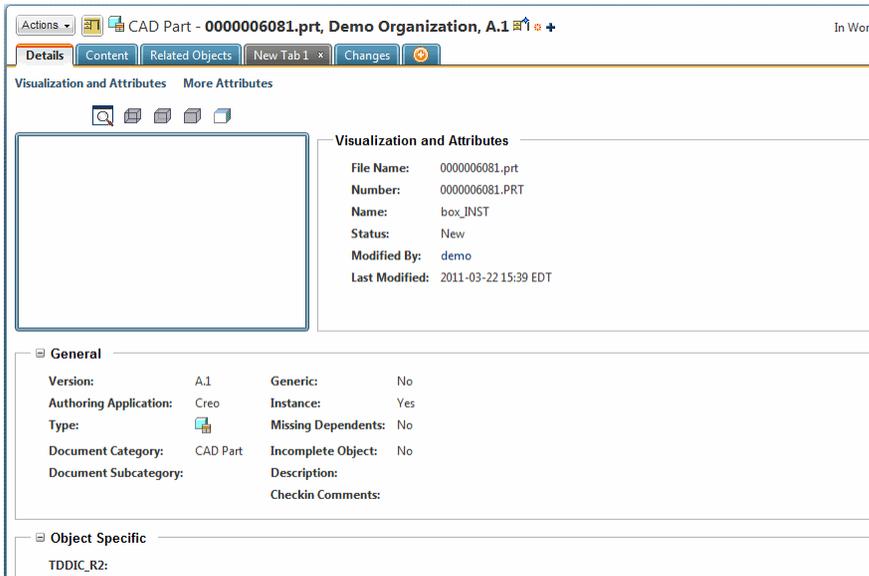
The 'Object Specific' section lists the following parameters:

- TDDIC_R2:
- TDDIC_B:
- TDDIC_R1:

Selecting the **Related Objects > Family** on the information page of any family member refreshes the information page to display the other members of the Family Table:



Each instance is an independent document in its own right, and can be searched for, downloaded, and included in assemblies independently from other members of the family.



Checking Out and Adding Family Tables to the Workspace

The following rules apply to the checkout and download of Family Tables:

- If you check out an instance, you do not need checkout the generic

Note

Default behavior is to checkout the generic also, but you can override this by deselecting the generic on the checkout page; however, in the case of an earlier iteration of a family table, you must check out the entire family table together.

- You can checkout the generic without checking out any instances.
- You can download the generic without downloading any instances.

Modifying Family Tables

You can modify Family Tables using Creo Parametric and then check the modified table into Windchill. You can also edit attributes of Family Tables in Windchill

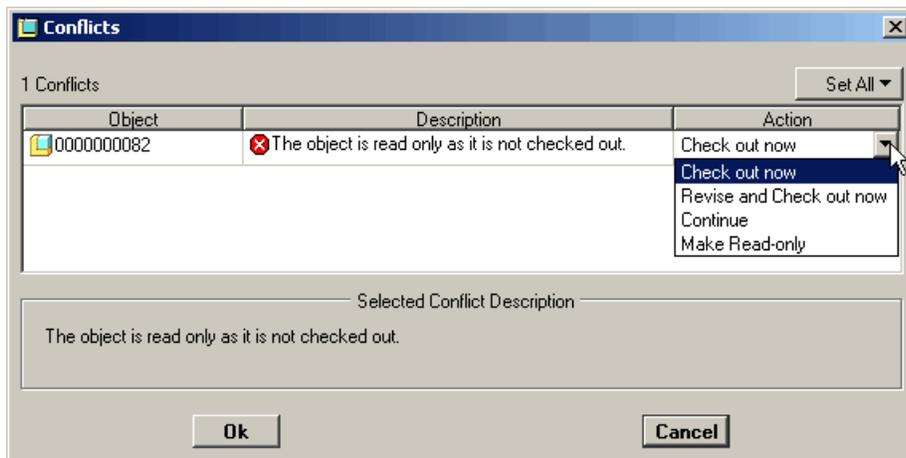
Modifying Family Tables in Creo Parametric

The following procedure describes how to use the Family Table editor in Creo Parametric to modify a Family Table:

1. Open the generic in Creo Parametric.

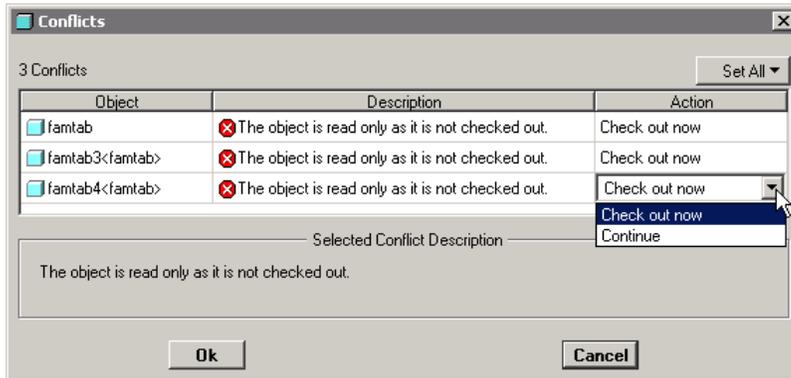
Starting with an empty workspace and opening the generic in Creo Parametric downloads (no checkout) the generic to the workspace. No instances are added to the workspace at this time.

Open the Family Table Editor (**Tools** ► **Family Table**). If an object is not checked out, when you attempt to edit it the **Conflicts** window appears, informing you that the object is read-only. The **Actions** column displays the recommended action (typically, to check out the object).



Note

When you click the cell in the **Action** column, the system presents a list of choices for that object: Check out (this object) now, Check out entire family table, or Continue. If you choose to continue (no checkout) you will be able to save, but not to upload your modifications.



If you have multiple objects in the **Conflicts** window, you can use the **Set All** list to select an action for all listed objects.



2. Modify a Family Table instance.
 - a. Select a row (instance) in the Family Table.
 - b. Edit the existing cells of the Family Table.
3. Verify the instances (**Tools** ► **Verify**), and exit the Family Table Editor.
4. Save and check the generic into Windchill (You can check in from the Model Tree, using the workspace **Check In** action, or using **File** ► **Check In** in Creo Parametric).

Modifying Family Table Attributes in Windchill

The attributes of generic or instance Creo Parametric Family Table members can be added or modified using the **Edit Attributes** page; however, because of their interdependency, additional members of the Family Table may need to be checked out in addition to any whose attributes you want to modify.

The following information is provided to review the classification of Family Table attributes, and describes the requirements for, and results of, attribute modification.

There are three classifications of attributes for Family Table objects, as described in the following table:

Attribute Class	Description
Generic-driven	An attribute (parameter) that has been designated in the generic model, but not added as a column to the Family Table. All instances reflect the same value as the generic for this attribute.

Inherited Table-driven	An attribute (parameter) that has been designated in the generic and has been added as a column to the family table, without assigning a unique value to the instance. The Family Table editor in Creo Parametric displays an asterisk (*) for the value for the instance(s). In Windchill, the value displayed for the instance is the same as is displayed for the generic
Independent Table-driven	An attribute (parameter) that has been added to the Family Table and a unique value has been given to the attribute either via the family table editor or the Windchill Edit Attributes page

When adding or modifying the value of generic-driven attributes, all Family Table members must be checked-out. If your initial selection of objects did not include all members, upon clicking **OK** on the **Edit Attributes** page, the **Confirm** window opens, listing the additional instances that need to be checked out. Clicking **OK** checks out the additional instances to allow the Family Table to be modified. Clicking **Cancel** returns you to the **Edit Attributes** page without modifying the Family Table.

When you modify an inherited table-driven attribute at the instance level, it becomes independent. That is, its value is no longer driven by the value of the generic. Therefore, only an instance whose value is being modified requires a check out.

Modifying an independent table-driven attribute requires only that the instance is checked-out.

Saving Family Tables Objects as New Objects

You can create a copy of any family table member (generic or instance) in your workspace by using the Save As command. Moreover, depending on how you select the family members, you can use the Save As command to do the following:

- Copy the entire set of instances.
- Copy the generic object only (as a standalone object).
- Copy a partial set of instances
- Copy a single instance object (as a new instance of existing table, not as a standalone object).

The Save as action can be initiated from several places in Windchill, for example:

- An actions menu on the Folders page that contains the object you want to copy
- An actions menu on the information page of the object
- The toolbar of the a CAD document's **Structure** tab

The following sections provide details on the various ways to copy Family Table members using Save As.

Copying the Generic Object Only

1. You can use Save as to copy only the generic  of a family table. When you have entered the Save As action on a generic object, use the following procedure to copy the generic object only:
2. In the **Save As** table, enter a name for the new generic in the **New Name** field.

Note

The system automatically generates a name for the new object. Enter a name for the new object only if you want to override the automatic naming.

3. Enter a folder location in the **Location** field or accept the default value.
4. Click **Ok**, the system creates a copy of the generic as a standalone object and saves it with the specified name in the specified location.

Copying the Entire Set of Instances

To copy an entire set of instances, you can initiate Save As for a single instance  of a Family Table and proceed as follows:

1. On the **Save As** page, select the instance and collect family  CAD documents.

Note

The generic is included in the family collection.

2. The **Save As** table refreshes to include all family table members.
3. For each family table member, enter a name in the **New Name** field.

Note

The system automatically generates a name for each new object. Enter a name for an object only if you want to override the automatic naming.

4. For each family table member, enter a folder location in the **Location** field or accept the default value.
5. Click **Ok**. The system creates a copy of the entire family table and saves it with the specified name in the specified location.

Copying a Partial Set of Instances

You can copy a partial set of instances of a Family Table by initiating Save As on a single instance and proceeding as follows:

1. On the **Save As** page, select the instance and collect family  CAD documents.

Note

The generic is included in the family collection.

2. For each family table member, enter a name in the **New Name** field.

Note

The system automatically generates a name for each new object. Enter a name for an object only if you want to override the automatic naming.

3. For each family table member, enter a folder location in the **Location** field or accept the default value.
4. Exclude any instance that you don't want to copy from the **Save As** list by selecting its row and clicking the reuse icon .
5. Click **Ok**, the system creates a copy of the selected instances along with required generics as a family table and saves it with the specified name in the specified location.

Note

Objects saved to Windchill in releases prior to Wildfire 2.0 and Windchill 8.0 have a more restricted set of metadata in the database. Partial copy of a family table requires the richer meta data that is created by saving the objects to Windchill 8.0 (and above) with Wildfire 2.0 (and above). Once the meta-data has been upgraded, partial copy is then available on these objects.

Copying a Single Instance Only

You can copy a single instance of a Family Table and save it as a new instance of that table. Initiate Save As on the instance and proceed as follows:

1. The generic is also brought into the **Save As** table. Select it, and click the reuse icon  in order to not copy the generic.
2. Enter a name for the instance copy in the **New Name** field.

Note

The system automatically generates a name for the new object. Enter a name for an object only if you want to override the automatic naming.

3. Enter a folder location in the **Location** field or accept the default value.
4. Click **Ok**. The system creates a copy instance as a new instance of the existing family and saves it with the specified name in the specified location.

CAD Document Templates and Creo Parametric Start Parts

Both Windchill and Creo Parametric use default template files when creating a new object. In Creo Parametric these objects are called start parts and in Windchill they are referred to as CAD document templates. The Creo Parametric start parts traditionally reside on your local file system, while the CAD document templates are stored in the Windchill database. In a concurrent engineering environment, you may find it challenging to keep all of your CAD document templates up-to-date with your Creo Parametric start parts.

To remedy this, you may find it useful to manage your Creo Parametric start parts in the Windchill database. Additionally you can also create new CAD document templates that reference the same start part files. The end result is that regardless of whether a designer uses Creo Parametric or Windchill to create a new object, both applications will use the same set of template files.

Other advantages to managing your start parts in Windchill PDMLink are:

- They are easily updated.
- They are easily distributed.
- They are version controlled.
- Same objects are used for all new Creo Parametric CAD documents.

Managing Creo Parametric Start Parts In Windchill

1. In Windchill, create a new library which you want to store your start part files.
2. In Creo Parametric:
 - a. Create a new workspace.
 - b. Open the start part files for each Creo Parametric object type (for example, by clicking **File** ► **Open**, and then navigating to the start part and opening it in Creo Parametric).
 - c. Save the start part to the workspace.
 - d. Repeat steps b and c until all of the desired start part files have been saved to the workspace.
 - e. In the workspace, check in the start part files to the library that you created in step 1.
3. In Creo Parametric, you must set a configuration option for each of the start parts that you want to manage with Windchill:
 - a. Click **Tools** ► **Options**. The **Options** window opens.

- b. In the **Options** field, type the name of the configuration option associated with the desired start part (a full table containing each of the configuration options is provided below) and press **Enter**.
 - c. Click **Browse** and navigate to the corresponding start part file in the library that you created in step 1.
 - d. Repeat steps b and c until all of the desired object types are pointing to the correct start part files in Windchill.
4. In Windchill, create new CAD document templates for each of the desired object types that point to the appropriate start part files.

Tip

*The name that you chose to assign to the server when you registered the Windchill server in the **Server Management** window (see the Windchill Help Center topic, “About the Server Management Utility.”) is used in the path created in the steps above. If you change the name of the server, you will also need to update the values of these config.options.*

Setting any of these options will cause Creo Parametric to validate that it has access to the templates on startup. This will cause an authentication window to appear when you start Creo Parametric, requiring you to log into the Windchill server. This is normal behavior, and once authenticated you will not be required to authenticate again in the same session.

Creo Parametric Configuration Options

The following table lists configuration options relevant for start parts:

Configuration Option	Description
template_designasm	Specify the assembly to use as the default assembly template.
template_drawing	Specify the drawing to use as the default drawing template.
template_ecadasm	Specify the model to use as the default ECAD assembly template.
template_ecadpart	Specify the model to use as the default ECAD part template.
template_mfgcast	Specify the model to use as the default manufacturing cast template.
template_mfgcmm	Specify the model to use as the default manufacturing cmm template.
template_mfgemo	Specify the model to use as the default manufacturing expert machinist template.
template_mfgmold	Specify the model to use as the default

Configuration Option	Description
	manufacturing mold template.
template_mfgnc	Specify the model to use as the default manufacturing assembly template.
template_mold_layout	Specify the model to use as the default mold layout template.
template_sheetmetalpart	Specify the model to use as the default sheetmetal part template.
template_solidpart	Specify the model to use as the default part template.
start_model_dir	Specify the complete path to a folder containing start parts and assemblies

Using Library Parts

To avoid "reinventing the wheel," designers incorporate already-designed components into their assemblies. Online catalogs can offer a wide selection of such components; however, retrieving them into your Creo Parametric session, with proper references is not always straightforward.

Windchill PDMLink libraries provide a means for storing and controlling CAD parts that have been approved for use within your company. CAD parts contained in Windchill PDMLink libraries can be searched, browsed, and access controlled, and they work well with standard Creo Parametric commands such as **Insert > Component** (Creo Parametric can automatically resolve component references when using CAD parts stored in Windchill PDMLink libraries.). Common parts, such as fasteners, that are used in many end objects can be stored in a common CAD parts library, allowing all product designers to access and use those parts. Company-approved libraries also help enterprises by reducing CAD part proliferation (wherein the same CAD part may be redundantly gathered and assigned different part numbers by different engineers working independently).

Creating Libraries

The administrator for your site can set up libraries that correspond to your company's organization and processes. In fact, Windchill PDMLink comes with an out-of-the-box template for organizing a library specifically for CAD parts. If you are designated as a library manager by your organization, you can create a library and use it to store CAD parts of particular interest to your project. The first steps are to:

1. Create a library in Windchill PDMLink. Within your library, you can set up a folder structure to organize hierarchies of CAD parts and to group related CAD parts together. For more information on creating libraries, see the online help

for the Windchill PDMLink Library tab. For more information on administering a library, see the *Windchill Enterprise Administration Guide*.

2. Set the configuration option in Creo Parametric (`pro_library_dir`) to point to the library in Windchill PDMLink. This will speed access to the primary CAD parts resource.

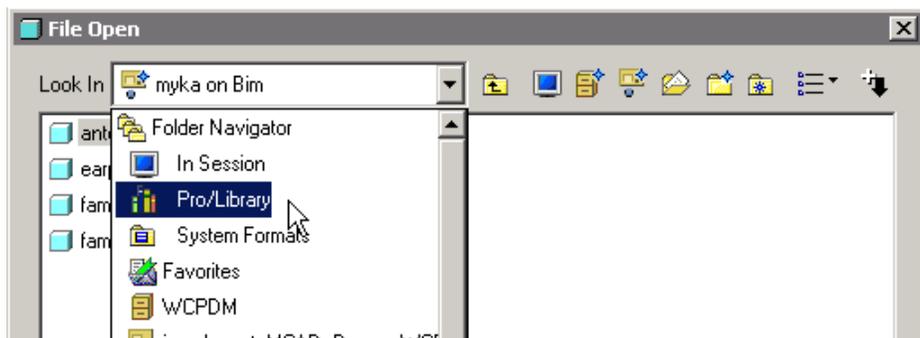
Retrieving Components from a Library

If you have a library in Windchill PDMLink that is a frequent source for components, a best practice is to set the configuration option `pro_library_dir` to point to that library as your default directory. Do this by setting the value of the config option to the path to the library, for example: `pro_library_dir = <Windchill PDMLink server URL>\...\<library folder name>`.

Tip

The server URL is a "wt.pub://..." URL, not a codebase URL. An easy way to specify the config option value is to click **Browse** in the **Options** window to present the **Select Directory** window. When you navigate to and select your library, its path is entered in the **Value** field of the **Options** window. Then, clicking **Add/Change** assigns your selected path to the value of `pro_library_dir`.

Once you have assigned the path of your chosen library to `pro_library_dir`, you can navigate directly to your library when you choose the **Pro/Library** option from the **Look In** list in the **File > Open** window.



Managing Incomplete Dependent Objects

An incomplete dependent is a CAD Document based on incomplete information known about a missing Creo Parametric file. For example, if you import or save an assembly file to the workspace, the saved file may have name references to a file that cannot be saved to the workspace (perhaps because it was suppressed in the assembly file). The issue is that all the required information to make the object complete is absent.

Typically, the information known includes:

-
- The model name
 - The CAD document type (inferred by the model name extension)
 - Whether or not the object is an instance or generic, and, if an instance, its relationship to the generic.

Information not known about the object that is usually defined by Creo Parametric includes:

- CAD document subtype
- Children or dependents of the object
- Designated parameters
- Other family table members. The generic will be known if it is an instance; but not intermediate generics or other members of the table.

Because good PDM practice does not allow checking incomplete data sets into the database, Creo Parametric provides several strategies for helping you to identify and resolve incomplete objects, including:

- Making users aware of incomplete objects in the workspace
- Providing tools for users to resolve incomplete objects by either removing the Creo Parametric reference or converting them to fully defined CAD documents.
- Providing tools for Administrators to prohibit or manage the creation of incomplete objects.

Identifying Incomplete Dependents

If you save a Creo Parametric file that references an incomplete dependent object to the workspace, the object is listed with this icon  to represent its object type.

Incomplete dependents are also listed appropriately in information page listings such as **Where Used** reports.

Resolving Incomplete Dependents

Creo Parametric allows you to deal with incomplete objects in two ways:

- Resolving an incomplete dependent from its information page
- Using the auto-resolve functionality available from the **Upload** or **Checkin** page.

Resolving Incomplete Dependents from the Information Page

You can resolve incomplete dependents using the Replace command or using the Add Placeholder command.

Using the Replace Command

The **Replace** command allows you to resolve an incomplete dependency by replacing a missing object with an object from the Windchill database. The command is only available from the information page of an incomplete object.

To replace an incomplete object, perform the following steps:

1. From the information page of an incomplete object, select **Replace** from the actions menu. The **Replace missing dependent <Object name>** page appears.
2. Select either **Replace with existing file from active workspace** or **Replace with file from commonspace** and enter or browse to (if commonspace file) the file name.

Clicking **Cancel** returns you to the information page without replacing the incomplete object.

3. Click **Ok** to replace the incomplete object with the selected document. All the parents of the incomplete object in the workspace are updated to reflect the selected object and the incomplete object is deleted.

If the parents are in Creo Parametric session, you are prompted to replace the objects in session. To persist the changes after replacing the incomplete object, upload the documents.

Using the Add Placeholder Command

Alternatively, the **Add Placeholder** command allows you to attach a CAD document template as a temporary file to allow the system to treat the incomplete dependent as fully defined.

To add a placeholder to an incomplete object, perform the following steps.

1. From the information page menu of the incomplete object, select **Add Placeholder** and click **Go**.
2. The system adds the template for the file extension of the Creo Parametric file to the incomplete object.

Note

The incomplete object must first be uploaded in order that:

- The **View Information** action is available from the object's workspace **Actions** column (Windchill ProjectLink only).
- The **Add Placeholder** action is available on the object's information page.

Resolving Incomplete Dependents from the Check In or Upload User Interface

When objects selected for check in or upload, or objects added to the list based on dependencies include incomplete dependent objects, the **Auto resolve incomplete objects** check box is also available. When selected, the auto-resolve functionality offers two options:

- **Update with object on server, then ignore**—The system searches on the server for an object with the same file name. If one is found, the incomplete object is updated by the found file. The object is no longer incomplete, and is therefore available for upload.

If no object is found to update the incomplete object, the system ignores the incomplete dependent (which is removed from the upload list).
- **Always ignore**—The system will simply remove any incomplete objects from the upload list.

Note

Site administrative policies may not allow the ignore option, or may only allow certain object types to be ignored. Required dependents cannot be ignored.

Simplified Representations

You can use simplified representation (or simp rep) tools to simplify an assembly by excluding components in a particular representation or substituting one component (CAD part or assembly) for another. Additionally, simp reps allow you to control the amount of data retrieved for a component. So for any given component, you can retrieve all data, just the geometry (no feature information by all surfaces and edges are represented) or just graphics (just a wireframe representation of the component). For example, when working with a very large assembly, you may find it useful to create a simplified representation that only contains the component in a small section of the assembly. This enables you to simplify your working environment significantly, and reduce the number of files that you need to download, while still including critical geometry.

Additionally, simplified representations improve the regeneration, retrieval, and display times of assemblies, enabling you to work more efficiently. You can use them to control which members of an assembly the system retrieves and displays. This lets you tailor your work environment to include only the information of current interest to you. For example, to speed the regeneration and display process, you can temporarily remove a complicated subassembly that is unrelated to the portion of the assembly on which you need to work.

Simplified representations are stored within the master assembly file, so if you are modifying a simplified representation, then you must have the master assembly file in session.

Tip

Bear in mind the following when using simplified representations with a Windchill system:

- Simplified representations only need to download the subset of files that are required by the simplified representation, thus speeding up operations that require file transfer (such as download and checkout)
- When several users are working concurrently on an assembly, any changes made to the simplified representation definition will require a checkout of the top level assembly. The consequence of this is that although many people can work on the assembly simultaneously, only one person at a time can be modifying the simplified representation definition. To overcome this restriction, consider using external simplified representations (external sim reps), described in the following section.

An external simplified representation is a presentation of a master assembly stored as a separate assembly model (whereas the simplified representation is stored in the master assembly file). An external simp rep contains particular components of the master assembly or their simplified representations. You can create multiple external simplified representations of a master assembly, each corresponding to a different area of the assembly and levels of detail. You can include low-level components without top and intermediate level assemblies and allow multiple users to work simultaneously. External simplified representations avoid the risk of accidental modifications to top level assemblies.

All the components included in an external simplified representation are the same as those in the master assembly. Therefore, it is not necessary to propagate modifications made to the external simplified representation or master assembly. All modifications to external simplified representations are automatically reflected in the master assembly.

External Simplified representations are supported by Windchill. This allows you to check in the external simp rep file to the Windchill database, retrieve an external simp rep assembly, and create (in Creo Parametric) an external simp rep "on-the-fly". The ability to use external simp reps allows multiple users to work on the same assembly without checking out the master. Users can work on their simp reps, which is particularly useful for large assemblies. Each user can check out only what's needed and download the rest.

In Windchill 10.0, the concept of an external simplified representation is supported by the Design Context object, a representative structure that can be derived from an existing CAD assembly or from a Configuration Context object. A configuration context is derived from a part structure. For more information, see the Windchill Help Center, "Creating a New Design Context from a CAD Structure."

Managing Model Items

You are able to select a feature—such as an annotation element, or a piece of geometry— within a given Creo Parametric model and designate the feature. At the same time as designating the feature, you have the ability of specifying whether this designated feature should be considered a key characteristic.

Alternatively, you are able to specify whether the model item to be created in Windchill should be considered a BOM item, which will convey to Windchill that a corresponding part be created and represented in the product structure after a build.

Managing Part-CAD Document Relationships

Since the first release of the Windchill Workgroup Managers, we have provided customers with the ability to drive part structures from CAD structures so that (where they are the same) the two will be kept in sync (CAD-driven product structure, sometimes referred to as “bottom up” design). In release 10.0, we provide customers with tools to drive CAD structure from part structure (top down design).

The following sections discuss how structures display and behave in Windchill. For more information, see the Windchill Help Center.

Managing a BOM with the Product Structure and CAD Document Structure

A Creo Parametric BOM (Bill of Materials) is a listing of the quantity and type of subcomponents that make up an assembly. When you are working in assembly mode you can generate a BOM (**Info ▶ Bill of Materials**) that you can inspect, print, or save to disk. In this form, the BOM serves principally as a convenient way to view a summary of the assembly.

By creating Windchill parts and associating them to the CAD documents that contain the Creo Parametric files as primary content, you can create a product structure, a hierarchy of enterprise parts in the PDM system that comprise an end item. The product structure mirrors the Creo Parametric model tree structure for the assembly.

The product structure in Windchill PDMLink (available from the **Structure** tab on the information page of a part with subcomponents) displays the product structure. The **Related Reports** menu on that page allows you generate the following types of reports:

- Single Level BOM
- Indented BOM

-
- Multilevel Where Used
 - Multilevel BOM compare

The Single Level BOM is similar to the Creo Parametric BOM report; however, in Windchill PDMLink you have the ability to annotate a BOM, which can be useful in spinning off new product configurations or new products themselves.

CAD-Driven Product Structure

Background

Driving the creation of product structure from the underlying CAD relationships has been a fundamental capability of our workgroup managers since the beginning. However, these capabilities were limited. Provided were two types of associations (build rules): Active (“Owner” starting in 9.0) and Passive (“Content” starting in 9.0).

- The owner association between a CAD model and a WTPart (part) allowed for participation in, and the driving of, product structure; the passing of attributes; and the passing of Creo View representations from the CAD document to the associated part
- The passive, or content, link between a CAD model and a part did not pass any information from the CAD document to the part, behaving in the same fashion as a Describes link between a standard Windchill document and part, passing no information from model to part.

Starting with Windchill 9.1, to provide greater flexibility, the characteristics of the Active association have been decomposed to its three basic building blocks:

- Structure—indicates that the CAD document "drives" the part structure
- Attribute—indicates that attributes will be "published" or "contributed" from the CAD document to the associated part
- Representation— indicates that when the corresponding CAD document is added to an assembly the build process will add its associated part to the generated product structure. It will also transfer the model's viewable from the CAD document to the associated part as an available representation.

The usage of these building blocks for creating relationships between CAD documents and parts must follow the following rules:

- Structure: Only ONE CAD document can drive the structure of a part
- Attribute: Many CAD documents can drive the attributes of a part
- Representation: A CAD document can only represent ONE part; but part can have many CAD documents representing it.

About Structure Build Rules

The Owner link assumes that you want to pass all children of this CAD document to its corresponding WTPart, provided that the child has a WTPart with at least a representation build rule associated to it.

The **AutoAssociate** and **Edit Association** pages will prevent the user from creating or associating a CAD document if it is:

- “Incorrect” type with an “R” type of build rule
- Manually flagged to not associate

When you define a Structure link, the user interface allows you to choose one or many particular EPMMember links to copy to the part structure. Only the immediate EPMMember links of the selected CAD document are presented in the user interface.

During the build, if the structure has changed, then the corresponding part is iterated; however, if the structure has not changed the build service will not iterate the part.

About Gathering Parts

Sometimes, CAD designs include subassemblies of parts for the purpose of organizing those parts as a group, rather than as an assembled structure (for example, a set of parts that are gathered as a group during the manufacture of a higher level assembly—sometimes referred to as a phantom assembly). The association and build actions create a Windchill part for this subassembly to parallel the CAD structure; however, as the subassembly is not a true assembly in and of itself, it has no place in the final BOM.

Windchill supports filtering these gathering parts from the BOM using a Gathering Part attribute on the Windchill part. When the Gathering Part flag is set to yes, it indicates an object that can be filtered from the BOM. There is no other impact resulting from setting this attribute, including no effect on the gathering part’s related CAD document.

A Windchill part can be assigned the Gathering Part attribute in the following way:

1. Set the value of the server-side preference, **Operations ▶ Auto Associate ▶ Phantom Assembly Override Attribute Name** to a BOOLEAN customer-defined soft attribute
2. In the CAD application, set the attribute to Yes, and then auto-associate will create a Gathering Part.

Setting Up Build Rules

Because the build is no longer performed as soon as the “build rules” or associations between CAD documents and parts are created, then you may find yourself setting up the associations (build rules) earlier in the process.

- There is still a preference to auto-associate parts to CAD documents upon every check in (**Operation ▶ Auto Associate ▶ Create Associate New Part**)
- You also have a preference to specify an attribute to prevent auto-association for a particular CAD document (**Operation ▶ Auto Associate ▶ Part Structure Override Attribute Name**)
- The **New CAD Document** page continues to allow to auto-associate upon creation
- In the workspace you continue to have an interface to auto associate outside of check in
- In the workspace you continue to have an interface to manually associate parts and CAD documents

About Representation Build Rules

Some representation build rules (Owner, Contributing Image, Contributing Content) require the part to be iterated during a build. That is, associating or disassociating a part and CAD document with those links requires that the part is checked out and back in.

Using these building blocks in a set of combinations has produced some alternative links providing a varying degree of behavior between what historically has been provided by the Owner and Content associations. The table below illustrates these new link types and their characteristics.

Link Name (Association Type)	Structure	Attribute	Representation	Related Part Added to Structure	Number of allowed links
Owner	True	True	True	True	1 (A CAD document can only be associated to one WTPart using this type)
Contributing Image	False	True	True	True	1 (A CAD document can only be associated to one WTPart using this

Link Name (Association Type)	Structure	Attribute	Representation	Related Part Added to Structure	Number of allowed links
					type)
Image	False	False	True	True	>1 (many CAD documents can be linked to the same WTPart, but a CAD document cannot link to multiple WTParts using this type)
Contributing Content	False	True	False	False	1 (Only one CAD document can be linked to a given WTPart using this type, but the same CAD document can be associated to multiple parts with this type)
Content	False	False	False	False	>1 (many CAD documents can be linked to the same WTPart)

About Attribute Build Rules

The Attributes passed to a WTPart are defined using the **Type and Attribute Management** utility.

- Attributes that can be “built” can be on:
 - EPMDocument (Iteration)
 - EPMDocument Master
 - EPMMember Link (Occurance)

An administrator is able to define:

- The complete set of global attributes that can be driven by the CAD document on the part

Note

The Name of the attribute on the CAD document drives the same-named attribute on the part

- Which attributes of the full set are only propagated from CAD document to part, as opposed to bidirectional propagation

For more information on mapping attributes, see the Windchill Help Center, “About Attribute Mapping.”

When you define a Contributing Image or Contributing Content link, a conflict is given if the target Part has a soft-attribute with a single value constraint and more than one CAD document is contributing a value for that attribute. During the build, if the attributes have changed, then the corresponding part is iterated; however, if none of the attribute values has changed the build service will not iterate the part.

The following table suggests some usage examples for the various types of association:

Association Type	Usage Examples
Owner	Multi-CAD secondary association, flexible components
Contributing Image	Multi-CAD secondary association, flexible components
Image	Multi-CAD secondary association, flexible components
Contributing Content	Multi-CAD where secondary attributes critical to BOM
Content	Inclusion of additional descriptive CAD content, for example, Model's Drawing to Part

In the first table above there is the column titled "Related Part Added to Structure". This might be the most important element to these new association types, at the same time the least understood. So, what does this mean for those link types listed in the table that have a value of True? This is best addressed by looking at a simple example, assuming the following:

- CAD document MODEL.PRT is associated to part MODEL using an Image link
- CAD document ASSEMBLY.ASM is associated to another part ASSEMBLY using an Owner link

Next, the CAD document MODEL.PRT is assembled into ASSEMBLY.ASM:

-  ASSEMBLY.PRT
 -  MODEL.PRT

The product structure for the ASSEMBLY part is updated as the build process is initiated at check-in. The result is that we now have a structure for ASSEMBLY as follows:

-  ASSEMBLY
 -  MODEL

This same behavior holds true for the Contributing Image link, as well. In summary, using these links means that the parts associated using the Image and Contributing Image links will have their associated parts contribute to the structure of its parent assembly. An Owner link would not only contribute to the structure but also the build structure for its node and its children, based on the CAD document that has been associated to it using an Owner link.

Consider the case where it is required to route the same cable multiple times within the same assembly. With some CAD tools, each routing represent unique geometry that must be represented by a unique file name in the assembly's model tree. However, from a product structure perspective, these routings represent a single part number. This example is the quintessential use case for these new link types. Let's assume that we have an assembly Top.asm with three routing of a cable cable.prt, cable_1.prt and cable_2.prt. Illustrated as a model structure:

-  Top.asm
 -  cable.prt
 -  cable_1.prt
 -  cable_2.prt

Now, using a combination of Owner and Image links we can associate cable.prt, cable_1.prt and cable_2.prt to the WTPart CABLE, as follows:

- cable.prt > Owner > CABLE
- cable_1.prt > Image > CABLE
- cable_2.prt > Image > CABLE

Also, assuming that Top.asm is associated to TOP with an Owner link, the following part structure is built:

-  TOP
 -  CABLE
 -  CABLE
 -  CABLE

And when a looking at the BOM Report you see:

-  TOP
 -  CABLE (qty. 3)

In the past, the additional parts for the non-built cables would have to be added to the product structure manually. Now with these link types a user can better leverage the automated CAD driven product structure capabilities of Windchill's CAD integrations. However, this is just one example. These link types can also be leveraged in multi-CAD scenarios, support for shrinkwraps, and the like.

Additional Behavioral Controls through Preferences

Additional controls are also provided through Windchill preferences (Operation > Auto Associate > ...) that allow further control over what CAD document types and sub-types can and cannot be associated using different flavors of link types.

For example, to not allow Owner link, only Contributing Image or lower level link:

- Operation > Auto Associate > Disallow Product Structure Links by Document Types
- Operation > Auto Associate > Disallow Product Structure Links by Document SubTypes

Or to not allow Owner, Image or Contributing Image links, only Contributing Content or below

- Operation > Auto Associate > Disallow Structure CAD Document Types
- Operation > Auto Associate > Disallow Structure CAD Document Sub-types

Overview of Top-Down Design

Consider the case where you have checked in a CAD structure actively associated with a part structure in Windchill. In top-down design (TDD) you are able to modify the part structure (on the part information page **Structure** tab) by adding to

or removing a child part from the part structure. Now user can update the CAD structure from the modified part structure. This is referred to as a reverse build: building CAD structure as per the part structure. At this stage the CAD assembly is said to be an annotated structure and in sync with part structure. To confirm the changes pushed to the CAD assembly, you must open the CAD structure in its CAD tool (Creo Parametric). As part of the retrieval process, you are asked to confirm that the child CAD document will be removed or added as per the changes done in the part structure. Checkin of the CAD structure completes the TDD cycle.

Characteristics of Part-CAD Document Relationships

Prior to Windchill 10.0, the association between a CAD document and a WTPart required the same (Modify) access control to exist on any state in order to modify the CAD document. Now it is possible to update (check in/out) the CAD document even if the user does not have sufficient access to the part. In addition, the action of associating or disassociating a part and CAD document requires an interaction of the part. For more information about comparing and editing CAD and product structures, see the Windchill Help Center.

Prior to release 10.0 the build did not populate occurrence information on a built product structure. Top-down design and flexible product structure editing made available in release 10.0 is dependent on the occurrence link information being instantiated properly. This is mostly applicable to the proper control over the deletion of parts from a CAD driven product structure.

Note

*Due to this change in Windchill 10.0, customers upgrading will see that the build status on the usage link will indicate that the build is 'Completed.' However, this is somewhat misleading. If one navigates to the **Occurrences** tab, it will show that the build status remains 'to be built'. Prior to leveraging the new capabilities in Windchill 10.0, by editing this product structure, a new build should be executed.*

Comparison User Interface

Windchill provides a dual structure browser that allows comparison of two part structures, or (starting from a part structure) comparing with an associated CAD structure, or (starting from a CAD structure) comparing with an associated part structure. With this comparison utility you can do the following:

- View part structure and CAD structure side by side
- Identify via the **Child Difference** column structure components with a difference in their subassemblies (the Build Status column indicates where any differences actually occur)
- Build part structure from CAD structure from the **Compare CAD Document to Part** utility
- Build CAD structure from part structure using the **Compare Part to CAD Document** utility

- Find or search objects in the structure.
- Assign **Set for Build** status to a part in part structure in the **Compare Part to CAD Document** utility

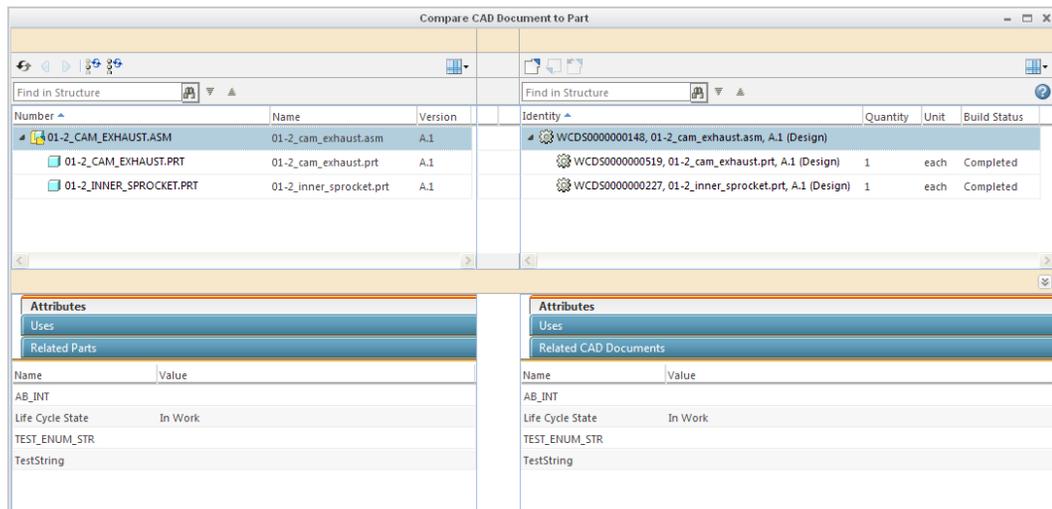
How to Determine the Structures Are in Sync

You are only truly be able to tell if structures are in sync or not by looking at the two structures (CAD and part) side by side in the new structure browser.

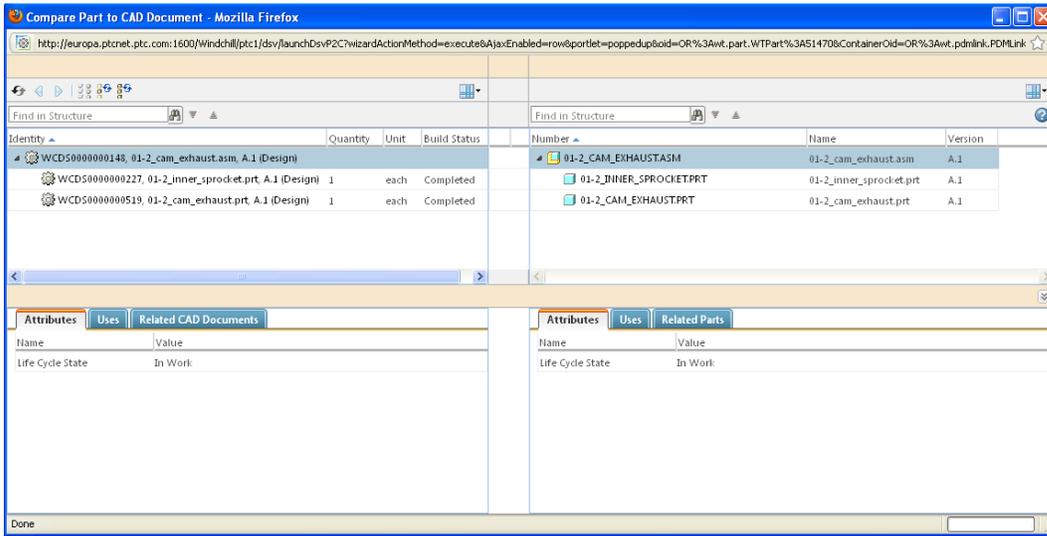
You can compare the part structure to the CAD document building its structure (possibly heterogeneous; that is, authored in more than one CAD application). Only looking at the CAD documents building the structure will show if the structure is up to date; however, it might be a configuration that is not retrievable.

When the user requests the software to perform a build (CAD structure drives part structure) or reverse build (part structure drives CAD structure), the build is performed and the updated structure is iterated. If a part builds CAD structure or vice versa, it creates an EPMBuildHistory link.

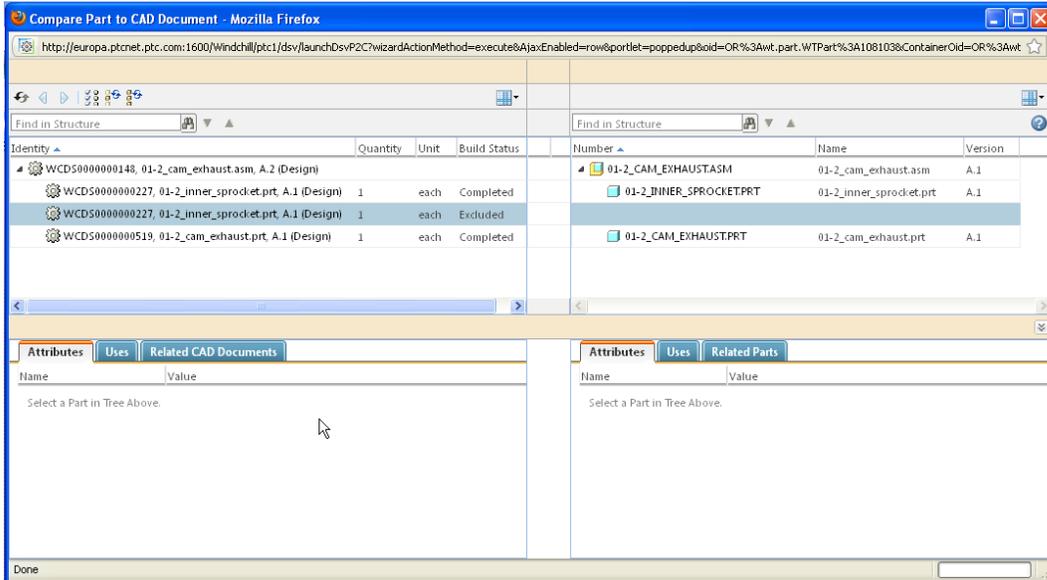
The following graphic shows the **Compare CAD Document to Part** utility (with assemblies in sync):



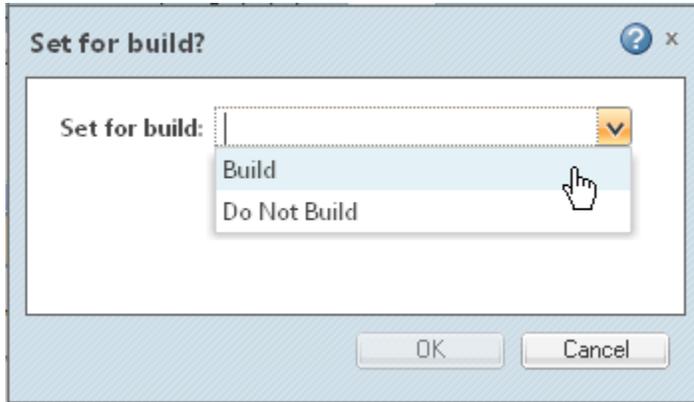
The following graphic shows the **Compare Part to CAD Document** utility (with associated assemblies in sync):



The following graphic shows the upper portion of the **Compare Part to CAD Document** utility (with structures potentially out of sync—a part has been added to the part structure, but its associated CAD document is not in the associated CAD structure. Currently, the new part is excluded from any new build:

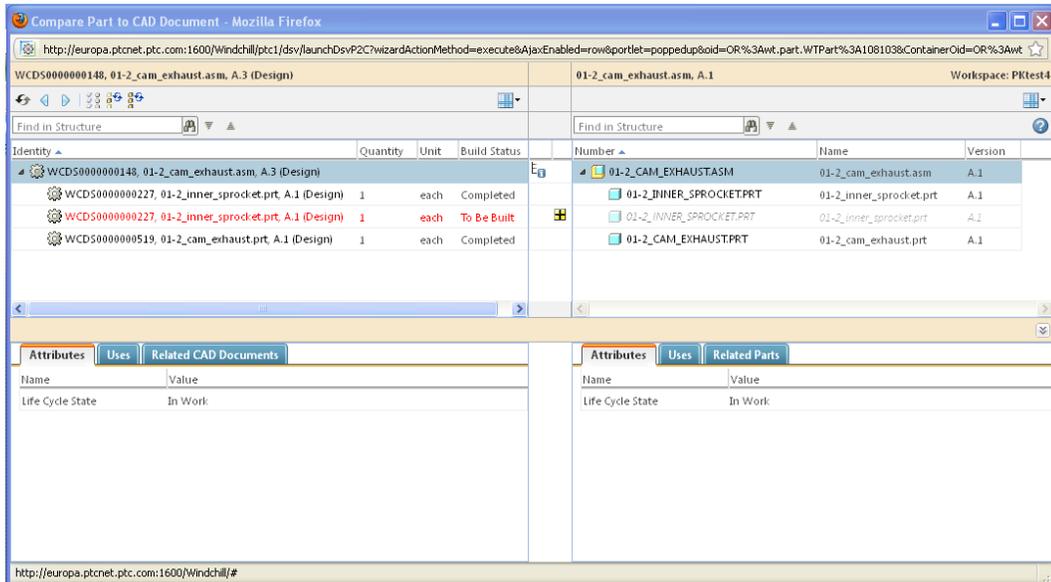


By selecting the new part and the **Set for build** icon  (also available in a right-click menu) you can call the **Set for build** window:



If you select **Do Not Build** and click **OK**, the build status of the new part remains “Excluded.”

However, if you select **Build**, then the part receives the status **To Be Built** (is set to be fully integrated into the part structure). At this point a **Child Difference** symbol is displayed between the structures indicating the asymmetry among the children. A tooltip appears if you hover over the status symbol next to the “missing” CAD part, explaining that the CAD part is not part of the current CAD structure, but will be built into the CAD structure if a build is initiated.

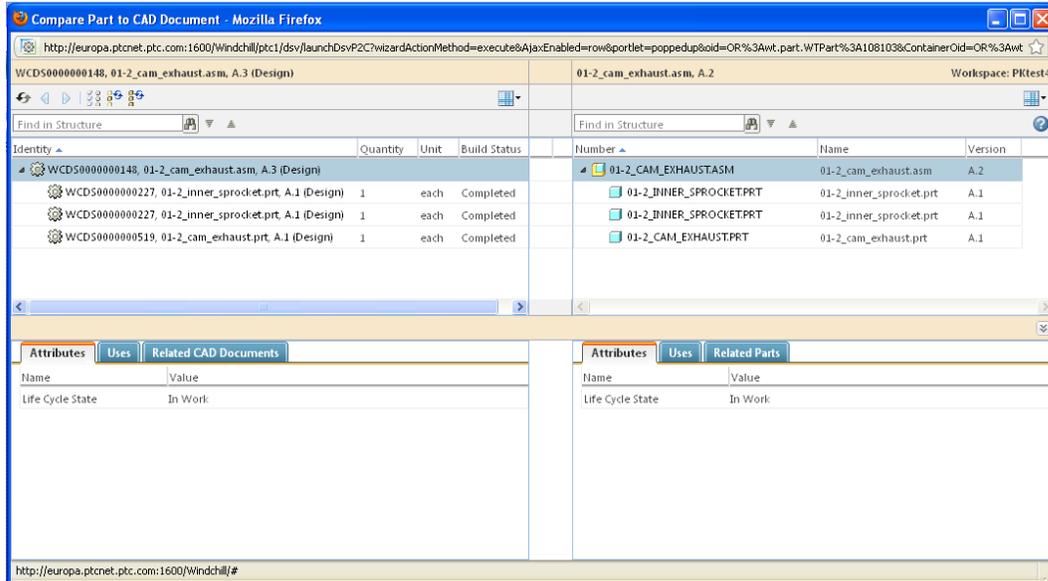


You can manually initiate a build of the entire structure  or of one selected level .

Note

The part was must be checked-in before a build can occur.

Once built, the new structures are again in sync and the CAD structure has been updated in the workspace, driven in a top-down fashion by updates to the part structure.



Note

The completion of the top-down design cycle requires that the newly edited CAD structure is able to be opened in its CAD application. Beginning with Creo Parametric 1.0, a CAD designer can view the edits performed in Windchill and accept or ignore them during the Open in CAD tool process. For more information, see the Windchill Help Center, “Verifying Windchill Editing Instructions.”

For more information on using the comparison utilities, see the Windchill Help Center, “Comparing CAD Document Structure to Part Structure” and “Comparing Part Structure to CAD Document Structure.”

A build can be executed from the **Compare CAD Document to Part** utility (CAD structure on the left, part structure on the right). This build will properly update the occurrence information.

Once the occurrence information is corrected, The data will be in a state to take advantage of Windchill 10.0 capabilities. For example, the system is capable of fully managing deleted occurrences. This means that you can delete parts in the product structure and control whether changes should or should not be propagated down to the CAD structure in a top-down fashion. If the occurrence information has not been updated, then a top down build will always remove any deleted parts from the CAD structure.

When to Build

A build can occur automatically (no user interface) after being triggered by a user-invoked action (for example, Check In or Promote). Alternatively, a build can occur based on a manual request (for example, from an explicit Build Structure action invoked on the CAD document information page)

Automatic Build—Retains the convenience of having the build occur without any user request or interaction (as in prior releases). You may configure the system to automatically build at the default values after the following events:

- Creation of a new CAD document iteration with a build rule to a part
-
- Change of state of a CAD document with a build rule to a part
- Modification of the CAD Document Master with a build rule to a Part
- Some combination of the conditions above
- Never occur automatically

An automatic build might also need to build more than just the object affected, but its children as well. The set of children affected by an automatic build is based on their “maturity” or lifecycle state.

The build :

- Can be executed only on checked-in CAD documents
- Will modify the latest iteration on the version of the part associated via the build rule
- Will automatically commit (check in) the changes to the parts
- (If the build is to build only the selected object) then only objects affected by the action are built.
- Otherwise, a recursive build builds all children that meet the “lifecycle state” to build.

You will receive success/failure messages in the **Event Management** utility. If the build fails, then the event that triggered it still continues, as we allow the two structures to be out-of-sync.

Manual Build—The command to build part structure and attributes from CAD documents is available from:

- CAD document commonspace information page:
 - In the main actions menu
 - In the structure browser
- Part information page (either checked-in or working copy version):
 - In the main actions menu
 - In the structure browser

-
- Structure Editor
 - Common implementation of the **See Actions** menu when the part is driven. For example, in:
 - Folder browser
 - Search results
 - Action user interface for the following actions:
 - Check In page
 - Promote page
 - Set State page

Verifying Windchill Editing Instructions

During top-down design a product structure can be edited in Windchill. The resultant modifications are pushed to its associated CAD structure during a “reverse” build, and the CAD structure is considered to be annotated. The completion of the top-down design cycle is to open the annotated CAD structure in Creo Parametric and then check in the structure as modified. Because modifications have been made to the structure, Creo Parametric typically presents the **Conflicts** (check out on-the-fly) window to alert the user that modification is in progress and a checkout is required.

With Creo Parametric 1.0, a configuration option (`enable_show_changes = yes`), allows a CAD designer opening an annotated CAD structure to see the **View Changes** window, which lists the components edited in Windchill and allows the designer to accept or reject the changes stipulated by Windchill. When all changes have been addressed, the accepted changes are regenerated with the model. The rejected changes will not be performed.

Heterogeneous Design

Heterogeneous design in context (HDIC) refers to the ability to incorporate design data that was authored in other CAD tools (for example, Creo Elements/Direct Modeling or CATIA V5) into a Creo Parametric design. This functionality must be enabled by settings for the Windchill server and in Creo:

On the Windchill server, do the following settings:

- Set the preference, **Workgroup Manager** ► **Client** ► **Open In CAD Tool For Non-native Objects**, to Yes (default) to allow a user of Creo Parametric to open supported non-native CAD documents into Creo Parametric session.
- Add the following property in the `site.xconf` file to enable the New Image Design Context’ action on the **Structure** tab for a Windchill-authored design context .
`com.ptc.windchill.uwgm.cadx.caddoc.enableImageDesignContextActions`

```
overridable=true
targetFile=codebase/wt.properties
value=true
```

Note

Run `'xconfmanager -p'` from Windchill shell to propagate this property.

In Creo Parametric, set the following config.pro options:

- Set the config.pro option `topobus_enable` to YES to allow you to directly import a non-native file into Creo Parametric session.
- Set `hdic_export_v5_to_ws_enable` to YES to enable HDIC export as Image for CATIA V5 to workspace

Note

In worker mode, this config option is not needed to be set on worker Creo client.

- Set `hdic_export_v5_secondary_cgr` to YES so that HDIC export for CATIA V5 allows the attachment of CGR as a secondary content.
- Set `intf3d_out_parameters` as appropriate (values are All*, designated and None) to enable attribute exchange (the option sets which parameters are exported with models).

These settings are designed to work with the Creo Parametric module ATB (Associative Topology Bus) which enables the direct import of a non-native file. With the Windchill preference set, opening of these non-native files is supported from a Windchill PDMLink search, from the CAD document information page, from the workspace, and from the Search Navigator, if you are using the embedded browser (not supported for a standalone browser). The advanced mode of the Windchill collection process enables the inclusion of source and image CAD documents for selected CAD documents to be included in the collection of objects for PDM actions.

Briefly, in heterogeneous design non-native files can become source files for Creo Parametric-created files that are derived from them, which are referred to as image (also, TIM or translated image model) files. Managing these files in a PDM environment is aided by additional information page reports such as the **Source and Image** table and the **Relationship Report**.

The **Source and Image** table displays the sources or images for the object whose information page you are viewing, and it indicates in a status column if the source and image are in sync. No icon displayed in the status column indicates that the source or image is up-to-date with regard to the object version of this info page.

When the row in the report shows an image, this icon  indicates that it is out-of-date with regard to the object version of this information page (which, in this case, is the source). This could happen, for example, when the source object has been

iterated since the image file was derived. When the report shows a row with a source object, this icon  indicates that there is a newer version of the source and the object version of this information page is out-of-date with regard to it.

It should be noted that both an original, non-native model and the TIM can be associated to the same Windchill part (though only one can have an owner association) and therefore both can contribute to a BOM.

For more detailed information about heterogeneous design, refer to Creo Parametric documentation.

Working with Configurable CAD Documents

Configurable products are special Windchill product structure types and CAD structure types designed to include one or more solutions of an options set defined for the structure. They are part of a Windchill strategy to allow companies to deliver products by adding optional features and components to a base model or a generic platform.

When configurable CAD assemblies are associated to configurable product structures, options and option sets can be passed to CAD documents from the associated part during the “reverse build” process (that is, when a CAD structure is built from the part structure.)

Depending on your case and your design process, you may start your work on a configurable structure in Creo Parametric or in Windchill. The following use case examples give a high level summary of the steps involved. For more detailed information on creating and modifying configurable assemblies in Creo Parametric, see the Creo Parametric Help Center. For more information on configurable product structures in Windchill, see the Windchill Help Center topic, “Setting up Configurable Product Structures.”

Example of a Top-Down Configuration Approach: From a Configurable Product Structure to a Configurable CAD Structure

In this scenario, you start with a configurable part structure in Windchill, which has an associated configurable CAD structure in Creo Parametric.

In Windchill:

1. Create an option pool to contain all required options, choices, and rules to configure the supported product variations.
2. Create an option set that contains the relevant options, choices, and rules for the configurable product.
3. Include configurable modules in the structure to capture variations in design solutions.
4. Assign choices to child parts of configurable modules in the part structure.

-
5. Update the CAD structure from the configurable product structure.
The choices assigned to parts in Windchill will be propagated to the corresponding CAD objects in Creo Parametric.
 6. Configure the part structure to generate a product variant and validate your selections. Save the variant specification and the product variant.
 7. Preview the configurable CAD structure.

In Creo Parametric:

1. Open and check out the configurable product (an overloaded assembly with all optional components) and review the placement of configurable modules.
2. If changes have been made, check in the configurable product
3. Configure the product using a variant specification.
4. Check in the variant product (variant top-level assembly).
5. Associate the product variant assembly (created in Creo Parametric) with the product variant part structure (created in Windchill).

Example of a Bottom-Up Configuration Approach: From a Configurable CAD Structure to a Configurable Product Structure

In this example, you start with a configurable product in Creo Parametric that will be used to generate an associated configurable product structure in Windchill.

In Creo Parametric:

1. Open a configurable product in Creo Parametric and review its configurable modules
2. Upload the configurable product, so that the enterprise options set will be available in Creo Parametric.
3. Assign the choices to the dependents of the configurable modules in Creo Parametric.
4. From Creo Parametric, check in the configurable product to Windchill to build the corresponding part structure in Windchill. During the checkin, the system generates associations between CAD documents and parts in Windchill.

In Windchill:

1. Define options and choices in an option pool.
2. Create and assign an option set to a top-level item in the configurable product structure in Windchill, or to the context of the configurable product.
3. If desired, designate the top-level part in the configurable product structure as the end item.
4. Assign or modify choices to parts where variability is required.

-
5. Configure the product to generate a product variant and the corresponding variant specification.
 6. Using the Compare tool, compare the two structures: the CAD structure versus the part structure. In the compare dual structure browser, build the structure to propagate the choice assignments on parts to the corresponding CAD documents.

Note

*When comparing configurable part structures to configurable CAD structures (**Compare to CAD Structure**), a **Choices** tab is displayed, showing the Choices available for a listed Option Name. (The **Choices** tab only appears in the standalone browser.) When comparing configurable CAD assemblies to configurable part structures (**Compare to Part Structure**), an **Options** tab is added to the window, displaying the Choices available for a listed Option Name. (The **Options** tab only appears in the standalone browser.)*

In Creo Parametric:

1. Open the configurable product in Creo Parametric.
2. Review the choice assignment and update as needed. If there are no changes to the choice assignment, apply the variant specification generated in Windchill.

Note

If you have made changes to a choice assignment, you need to check in the CAD structure to propagate the changes to Windchill. Once in Windchill, you can configure the product structure to update the variant and the variant specification.

3. Check in the configurable product
4. Generate a product variant.
5. Check in the product variant to Windchill and auto-create a part structure for the variant.

In Windchill:

1. 1. Verify the newly created variant structure by comparing CAD and part structures in the compare dual structure browser.

Additional Considerations for Configurable Assemblies and Product Structures

The following information is applicable to configurable assemblies:

- In Windchill, the creation of a configurable CAD document requires an appropriate template to be set up.
- During the New CAD Document action, the ATO attributes for Configurable, Collapsible, and Min_Required, Max_allowed will be created for configurable CAD documents. No additional user interface is presented in New CAD

Document page, so the end result is that the proper CAD document (configurable product or configurable module) is created based on the template created.

- The term “configurable module” refers to an assembly that has multiple options, but uses only one option when used in a variant, whereas the term “configurable product” refers to a configurable assembly (in Creo Parametric) or a configurable product structure (in Windchill) that has multiple potential solutions.
- A top-level configurable part should be designated as an end item.
- Auto-associated parts searched or created for configurable CAD assemblies must be configurable parts. If a matching part is found and it is a variant part, it will not be shown in the user interface. You should create a new configurable part for association.
- When attempting to edit associations, be aware that object subtypes must be consistent
- During the **Save As** operation of **Update Parents**, be aware that a standard assembly cannot contain a configurable assembly.
- Design in context functionality is disabled for configurable assemblies.
- Configurable products saved to projects do not provide access to options and option sets.
- To ensure that choices assigned on dependents of a configurable model in Creo Parametric are passed to the associated part structure, choose their configurable module as owner during choice assignment.

3

Customizing and Administering Creo Parametric with Windchill

Configuration Settings in Creo Parametric	97
Configuring Windchill for Interoperation with Creo Parametric.....	109
System Configuration Recommendations	154
Performance Tuning	155
Other Recommendations	158
Windchill Preferences That Control Interaction with Creo Parametric	160

This chapter presents customization and administration information and recommendations for using Creo Parametric integrated with Windchill PDMLink and Windchill ProjectLink. The primary audience is Creo Parametric and Windchill system administrators; however, much of the information can be useful to end users as well.

The topics presented include Creo Parametric configuration information (environment variables and config.pro options) that applies to the interaction with Windchill, and Windchill server-side preferences, as well as specific information on parameter mapping, parameter customization, customizing object naming, automated part creation, supporting custom parts, and customizing the user interface. In addition, recommendations for system configuration and performance tuning are offered.

The final section lists and describes Windchill preferences that are especially relevant to the interaction with Creo Parametric.

Configuration Settings in Creo Parametric

The following sections describe environment variable settings and config.pro options useful for configuring Creo Parametric to work with Windchill.

Environment Variables

Creo Parametric uses a user-visible workspace to manage work-in-process data. Each workspace uses a local disk cache to ensure data integrity and optimize file transfer between Creo Parametric and the server. The cache (which is managed by Creo Parametric and is not visible to the end user), is used to store changed objects prior to an upload to the server, and to keep copies of objects downloaded from the server to speed up subsequent retrieval into Creo Parametric.

As a system administrator, you may wish to put the cache on a larger disk partition than provided by the default location. The following table lists environment variables that can be set by a system administrator to manage the placement of the cache into a suitable partition:

Variable	Values	Description
PTC_WF_ROOT	/path/to/dir, Default on UNIX = ~/wf Default on Windows = [User Profile]\Application Data\PTC\ProENGINEER\ Wildfire\	Overrides the default location of .wf directory. Setting this environment variable will cause Creo Parametric to use the new location as a location for the cache. Note <i>Existing cache data will not be copied to the new location automatically.</i>
PTC_WF_CACHE	/path/to/dir, default=\$PTC_WF_ROOT/. cache/	Allows the specification of additional cache space. If you are running out of disk space in \$PTC_WF_ROOT, you can use this environment variable to define a folder in which all new workspace caches will be stored.

Variable	Values	Description
		<p>Note</p> <p><i>This new folder only applies to newly created workspaces. Existing workspaces will continue to reside in \$PTC_WF_ROOT/.cache</i></p>
PTC_WLD_ROOT	/path/to/dir,	<p>Allows specification of the location of the .ws cache directory.</p> <p>Note</p> <p><i>By default, when you register Windchill servers and authoring applications, a .ws directory is created for cache. It is located under your user profile on your local computer. For every server location listed in the .ws directory, there are subdirectories for each workspace, and within each workspace subdirectory, there are subdirectories for each authoring application that is registered to that server.</i></p>

Note

The environment variable *EPM_MODE*, designed for earlier versions of Creo Parametric, should not be used with Creo Parametric. Because it prevents the **Conflicts** ("check out on-the-fly") window from appearing when users attempt to modify a checked-in model, its use could lead to loss of data when modifications cannot subsequently be checked in.

Config.pro Options

The following table lists Creo Parametric config.pro options that are especially relevant to the Creo Parametric interaction with Windchill:

Config.pro Option	Values	Description
compress_output_files	yes no [default]	Controls whether or not to compress object files to store them.

Config.pro Option	Values	Description
		<p>Compressed files are slower to read and write, one-half to one-third the size, and fully compatible across systems. When set to "yes," stores object files in compressed format. When set to "no," object files are not compressed.</p> <p>Note</p> <p><i>The time spent in compression and decompression could be more expensive on CPU than the benefits for disk or network. In some WAN environments, this could be a helpful for some transfers, but is not recommended for general use.</i></p>
disable_search_path_check	no [default] yes	<p>Controls whether or not the search path is checked for name conflicts when creating, renaming, or copying models.</p> <p>When set to "yes," disables the check of the search path for a naming conflict when a new file is created. This can speed up file creation by postponing the search path check (which includes the entire common-space) until an upload is performed.</p>
dm_auto_open_zip	yes [default] no	<p>Defines how Creo Parametric will handle zip files.</p> <p>If set to "yes", then Creo Parametric opens the zip file and attempts to retrieve objects in the zip file. If the zip file contains more than one file (for example, in the case of assemblies), Creo Parametric will first attempt to open an object in the zip file that has the same name as the zip file itself. If it finds one, it will open it, if not, it will display the contents of the zip file in the File>Open window.</p>

Config.pro Option	Values	Description
		If set to "No", Creo Parametric treats a zip file like a directory, and displays the contents of the zip file in the File ► Open window, allowing the user to pick the file from the zip that he or she wants to retrieve into session.
dm_background_operations	yes no [default]	If set to "yes," allows user to take advantage of the backgrounding operations (for example, during check-ins of large data set). Note <i>Most situations can benefit from a yes setting, which allows working in the foreground while lengthy operations are run in the background.</i>
dm_cache_limit	Integer [default = 0]	Sets the size (in MB) of the cache allocated to the combination of all registered servers and their workspaces on the client hard disk. Recommendation: If possible, set the cache size large enough to accommodate the largest anticipated data set (the downloaded content and the locally modified content prior to upload should be counted separately). A good rule of thumb is 80% of the remaining free space on the disk where Wildfire cache is located. Note <i>A value of "0" (no limit) will tend to fill up the client disk, but could boost performance by eliminating checks on cache size and purges.</i>
dm_checkout_on_the_fly	checkout [default] continue	If set to the default value "checkout," the default action for the Conflicts ("checkout-on-the-fly") window is "Check Out Now."

Config.pro Option	Values	Description
		If set to "continue," the default action for the Conflicts window is "Continue."
dm_http_compression_level	Integer, from 0 (no compression) to 9 (maximum compression) [default = 0]	<p>Sets the level of compression for data upload and download.</p> <p>Although compression speeds up transfer over the network, it uses server CPU and client CPU to perform the compress and decompress operations. In a local area network, where network transfers are rapid, compressing and decompressing data can result in lesser throughput. On a wide area network with lower bandwidth compression can lead to higher throughput. Since this is set per client, PTC recommends that clients in a LAN use a value of 0 (the default) and clients in a WAN use a value of 2 or 3.</p> <p>For older Creo Parametric versions, the following approximate guidelines apply:</p> <p>If client download bandwidth < 3 Mbps enable dm_http_compression_level (at a value of 3).</p> <p>If client download bandwidth > 3 Mbps unset dm_http_compression_level as uncompressed response read times will be faster.</p> <p>As of Wildfire 2.0 M260 and Wildfire 3.0 M090, the following approximate guidelines apply:</p> <p>If client download bandwidth < 20 Mbps, enable dm_http_compression_level (at a value of 3).</p> <p>If client download bandwidth > 20 Mbps set dm_http_compression_level to 0 (uncompressed), so that response read times will be faster.</p>
dm_network_request_size	integer >0	Determines the maximum size, in

Config.pro Option	Values	Description
	[default = 100000]	<p>bytes, of an HTTP upload request when uploading content files to Windchill.</p> <p>The default of 100000 is likely to ensure that each file will be uploaded through a separate http request with minimal process memory consumption overhead.</p> <p>A small value (say 8000) would mean a lot of small HTTP requests to the method server containing the model files which may add overhead; but because the local Wildfire file buffers are filled quickly, the upload starts sooner.</p> <p>A much larger value (say 800000000) may allow the uploading of the entire data set in a single HTTP request, but it could take a while for the client to write the files from local disk to its internal buffer before streaming the content to the server. In addition, because of apparent size limitations of the Microsoft HTTP API you may experience random upload failures with very large file size data sets in Windows. In addition, working with large datasets has been known to cause Internet Explorer to run out of memory.</p>
dm_network_retries	integer >0 [default = 10]	<p>Sets the number of attempts to connect to a Windchill server before the connection is considered broken.</p> <p>Recommended setting: default</p> <p>Note</p> <p><i>If the http connection is unstable, a setting less than the default could increase failures, while a setting greater than the default causes delays if a failure occurs.</i></p>
dm_network_threads	integer >0	Sets the number of concurrent

Config.pro Option	Values	Description
	[default = 3]	<p>threads Creo Parametric uses for uploading and downloading data to and from a Windchill server.</p> <p>The recommended setting depends on the network bandwidth. It is suggested to keep at 3 for a WAN and can be increased to 6 for a fast LAN. However, in most cases, increasing the number of threads in a LAN environment will not improve performance, as the disk will then become the bottleneck. Even in a WAN environment, settings greater than the default are unlikely to improve throughput significantly.</p>
dm_offline_after_event	yes [default] no	Allows you to choose to work offline after a loss of the server connection. Staying online ("no") continues to retry server operations.
dm_offline_options_activated	yes [default] no	<p>If set to yes, the options to synchronize, download, and upload workspace data are checked in the Synchronize Workspaces window.</p> <p>If set to no, download and upload check boxes are unchecked by default for going online.</p>
dm_overwrite_contents_on_update	no [default] yes	<p>Specifies behavior during Update action.</p> <p>If set to "no," does not overwrite the locally modified contents for out-of-date objects, but updates their metadata only</p> <p>If set to "yes," overwrites the locally modified or out-of-date objects with the ones on the server in addition to updating their metadata.</p>

Config.pro Option	Values	Description
		<p>Note</p> <p>If the user wants to abandon the local cache modifications, s/he can perform an explicit download (Add to Workspace) of the model from the server-side workspace, thus overwriting the version of the model in the cache. Alternatively, after the Update to the latest iteration, the user can check it out and upload the modifications from the local cache. The non-default value of "yes" should be used if the user makes only temporary modifications in the cache and never intends to keep them after the Update.</p>
dm_remember_server	yes [default] no	If this option is set to "yes," the last primary server/workspace of a Creo Parametric session is set automatically for the next Creo Parametric session.
dm_save_as_attachment	yes [default] no	<p>Controls the default option for Creo Parametric Save a Copy when models are saved as in non-Creo Parametric format.</p> <p>If set to "yes," the model is by default saved as a secondary content attachment to the original CAD document.</p> <p>If set to "no," the model is by default saved as a (primary) CAD document.</p>
dm_search_primary_server	yes [default] no	If this option is set to "yes," during retrieval, the system searches the primary server for dependencies not found in the workspace
dm_secondary_upload	automatic [default] explicit	Defines the behavior of saving to a secondary server (See also dm_upload_objects).

Config.pro Option	Values	Description
		If this option is set to "explicit," the Creo Parametric File ► Backup command will write data to the cache. The user must then explicitly send that data to the server (using either using either the Upload or Check In commands invoked from the corresponding workspace). If this option is set to "automatic," File ► Backup in Creo Parametric will also upload the Creo Parametric files to the server.
dm_upload_objects	explicit [default] automatic	Defines the behavior of the Save command in Creo Parametric. If this option is set to "explicit," the Creo Parametric File ► Save command will write data to the cache. The user must then explicitly send that data to the server (using either File ► Save and Upload or File ► Checkin). If this option is set to "automatic," File ► Save in Creo Parametric will also upload the Creo Parametric files to the server.
enable_configurable_assembly	Yes No [default]	If set to yes, this option enables the creation of configurable assemblies.
enable_show_changes	no [default] yes	If set to Yes, enables the View Changes window, to allow a user to accept or reject the Windchill editing instructions when attempting to open an annotated CAD structure into Creo Parametric session.
let_proe_rename_pdm_objects	no [default] yes	Determines whether an object retrieved from a PDM database can be renamed in a Creo Parametric session An object rename in Creo Parametric is seen only by parents in session. The object is seen as a new object when saved to the workspace.

Config.pro Option	Values	Description
		This option can be used to replace a standard sub-assembly with a copy of itself with a unique name.
open_simplified_rep_by_default	no (default) yes <name_of_simplified_rep>	<p>Specifies whether to prompt user to select a simplified representation when opening a Creo Parametric file.</p> <p>If set to "yes," user is prompted to open a simplified representation when opening a Creo Parametric file.</p> <p>If set to the name of a simplified representation, the system opens the simplified representation without prompting the user.</p> <p>This option can be useful for using internal simplified reps on small to medium-sized datasets. If set to "yes," the user sees a pop-up listing the simplified reps available in an assembly, for example, when clicking on its hyperlink in the embedded browser. Choosing one of the simplified reps allows the user to add to the workspace and download only the models required for the corresponding simplified reps.</p>
regenerate_read_only_objects	yes (default) no	<p>Specifies whether read-only objects (objects not checked out) are regenerated.</p> <p>Set to "yes," it specifies that read-only parts with relationships to an explicitly modified assembly are modified implicitly upon regeneration of the assembly. (Explicit changes to a checked-in object cause the Conflicts window to appear).</p> <p>By setting to "no," you may avoid having read only workspace objects marked as modified. This, in turn,</p>

Config.pro Option	Values	Description
		can reduce the number of files required for the checkout of an associated assembly
retrieve_data_sharing_ref_parts	yes no [default]	Controls automatic reference parts retrieval for dependent data sharing features. When set to "no," prevents download/opening of components that are not immediately needed
save_model_display	wireframe shading_low shading_high shading_lod	Sets the quality of graphics that are shown on the Windchill information page. Setting this option to shading_lod creates the best images, but requires larger Creo Parametric file sizes to store the additional graphical information. Note <i>Saving the shaded display increases the model file size by 100 or more percent. Setting this option to shading_lod creates the best images, but requires larger Creo Parametric file sizes to store the additional graphical information. Setting to wireframe is the most lightweight format.</i>
save_objects	change- d_and_speci- fied [default] all changed change- d_and_ updated	Determines when an object and its dependent objects (such as a part used in an assembly) are stored. The recommended value in a PDM environment is "changed" (to avoid unnecessarily iterating the top-level object if it was not modified in session).
search_path	<directory paths by full path name>	Specifies list of directories to search (in order) for object/file retrieval. These directories, the working directory, and directories in search.pro

Config.pro Option	Values	Description
		<p>file (refer to search_path_file) are Creo Parametric's search path. Use full path name to avoid problems. It is best to use the minimum number of search paths to a minimum because a large number of search paths increases retrieval time.</p> <p>Note</p> <p><i>In Creo Parametric Wildfire it is not necessary to set the config.pro option, optionsearch_path. By default, when a Windchill server is your primary server, the entire primary server with active workspace is in the Creo Parametric search path.</i></p>
topobus_enable	no [default] yes	Allows direct import of certain non-native files into Creo Parametric session.
web_browser_homepage	string value	Sets the location of Creo Parametric browser homepage.

Note

In Creo Parametric it is not necessary to set the config.pro option search_path. By default, when a Windchill server is your primary server, the entire primary server with active workspace is in the Creo Parametric search path.

The config.pro options that specify storage and retrieval directories, including such options as the following:

- start_model_dir
- pro_library_dir
- pro_format_dir
- pro_materials_dir
- pro_group_dir
- pro_symbols_dir
- pro_catalog_dir

can be set to point to Windchill cabinets. For example, the value of start_model_dir is set to point to a Windchill library cabinet using the following syntax (<server alias> is the name you assign to the server in the Server Management utility):

```
start_model_dir wtpub://<server alias>/Libraries/<library_name>
```

Similarly, the value of `pro_group_dir` is set to point to a Windchill product cabinet using the following syntax:

```
pro_group_dir wtpub://<server alias>/Products/<product_name>>
```

Note

If you retrieve an object from any location other than the primary server, it will be treated as if it were newly created in the Creo Parametric session. This means that actions on the object (for example, save or requesting checkout) are done in the context of the primary server, not the location from which the object was retrieved.

Config.pro options that point to a specific file, including such options as the following:

- `intf_in_use_template_models`
- `template_designasm`
- `template_mold_layout`
- `template_ecadprt`
- `template_solidpart`

can be set to point to Windchill file locations using a string of the proper syntax and the name of the CAD document that manages the file, as in the following example:

```
template_solidpart  
wtpub://<server alias>//libraries/Templates/template_solid_inlb  
s.prt
```

Configuring Windchill for Interoperation with Creo Parametric

The following sections describe customization activities performed in some cases using Windchill properties files and in other cases using preferences that can be set using the Windchill **Preference Management** utility (**Site** ▶ **Utilities** ▶ **Preference Management**). Preferences may also be set at an organization or context level (**<Context or Organization>** ▶ **Utilities** ▶ **Preference Management**). Some preference settings can be accessed and overridden by end users. Some preferences can be locked by an administrator, to prevent users from overriding them.

Because the Windchill preferences are organized within the **Preference Management** utility by category, this guide will identify preferences by including the hierarchy to which they belong (for example, **Display** > **Workspace**).

Displaying the Workspace

When you access Windchill through the Creo Parametric embedded browser, you can always access and create workspaces. However, when Windchill is accessed through a standalone browser, default settings do not display links to the **My Workspaces** page where workspaces are listed and created in Windchill.

You can enable the display of the workspace (including the display of the **Workspaces** minor tab for a context) by setting the preference Display > Workspace to "Yes" (default is "No").

Undocking the Embedded Browser Window

Out of the box, Creo Parametric is enabled with an embedded browser window that is interactive with your Creo Parametric session. To enable you to better take advantage of a multiple display workstation, you can undock the interactive browser from the Creo Parametric user interface and drag it to a second display device while still maintaining the interactive nature of the browser. Other than its display location, the function of the undocked (unembedded) browser is identical to the function of the embedded browser when docked.

Note

It is important not to confuse the term ‘unembedded browser’ with ‘standalone browser.’ The former term refers to a browser capable of full interaction with your Creo Parametric session (exactly like the embedded browser). The term ‘standalone browser’ continues to refer to a browser window that is independent of your Creo Parametric session, and which is capable only of viewing Windchill server information.

The undocking of the browser is set by a config.pro option that specifies whether the browser will remain docked or undocked for the entire session.

To undock the embedded browser, use the following procedure:

1. At the start of your Creo Parametric session, set the config.pro option `web_browser_in_separate_window` to Yes (a No value leaves the browser embedded).
2. Having configured the unembedded browser, you can drag it to another display.

Managing CAD Document and WTPart Naming and Numbering

You can specify how newly-created CAD documents (EPMDocuments) and parts (WTParts) are named and numbered using a policy-managed method. Alternatively, you can use a customization of the Windchill Naming service to specify the names and numbers of CAD documents only. These two options are discussed in the following sections.

Note

Preferences referred to in the following explanation of policy-managed naming and numbering are explained in the section [Windchill Preferences for Naming and Numbering on page 114](#).

Policy-Managed Naming and Numbering

Creo Parametric supports four policies to determine how newly-created objects (either CAD documents or WTParts) are named and numbered. The four policies can be described briefly as follows:

- Auto-numbering
 - The CAD document Number is server-assigned (either OOTB or per your customization)

Note

Customization of autonumbering must ensure that unique numbers are assigned.

- If the system parameter PTC_COMMON_NAME is created during creation of the Creo Parametric model file, its value is copied to the CAD document Name. If Common Name is left blank in Creo Parametric, the default value for the CAD document Name is copied from the Creo Parametric model name. The file extension (.prt) can be optionally dropped (controlled by a preference).
 - The WTPart Number is provided by the WTPart Number generator.
 - The default value for the WTPart Name is copied from the current value of the CAD document Name at the time the WTPart is created. If a file extension is present in the CAD document name, it can be optionally dropped when set in WTPart (controlled by a preference).
 - In any create and edit user interface, the CADDoc and WTPart NAME field is editable.
 - Auto numbering is the default, out-of-the-box naming and numbering policy. The default system does not allow editing of the number by users.
- Name-driven

-
- If the CAD document Number is not set up to be server-assigned, then the CAD document Number is copied from the Creo Parametric file name (the file extension can be dropped—controlled by a preference).
 - If the system parameter PTC_COMMON_NAME is created during creation of the Creo Parametric model file, its value is copied to the CAD document Name. If Common Name is left blank in Creo Parametric, the default value for the CAD document Name is copied from the Creo Parametric model name (file extension can be dropped – controlled by a preference)
 - If the WTPart Number is not set up to be server-assigned, then the default value for the WTPart Number is copied from the CAD document Number (the file extension can be dropped—controlled by a preference)
 - The default value for the WTPart Name is copied from the current value of the CAD document Name at the time the WTPart is created (the file extension can be dropped—controlled by a preference)
 - In any create and edit user interface, the CAD document and WTPart NAME and NUMBER fields are editable by the user.
 - Parameter-driven
 - The CAD document Number is copied from the value of the Creo Parametric designated parameter identified by the preference, Operation > Upload Operation > Upload > Numbering Parameter. (If no preference value is set, Number assignment follows that of the name-driven policy.)
 - The value for the CAD document Name is copied from the Creo Parametric designated parameter identified by the preference, Operation > Upload Operation > Upload > Naming Parameter.
 - The value for the WTPart Number is copied from the value of the Windchill attribute identified by the preference, Operation > Auto Associate > Auto Associate Numbering Parameter.

Note

Setting this preference takes precedence over auto-numbering, to facilitate user intent in the auto association action. Auto-numbering rules (if any) are observed if this preference is unset.

- The value for the WTPart Name is copied from the value of the Windchill attribute identified by the preference, Operation > Auto Associate > Auto Associate Naming Parameter .

Note

If this preference is unset, name-driven (not autonumbering) policy is observed, even if auto-numbering is rules are otherwise in effect.

Name-driven and parameter driven policies can only be used in object-driven creation of objects as they require a source object to create a new object. These policies are used during upload (when a new CAD document can be created based on a model file), and auto-associate, (when a new WTPart may be created for a CAD document).

Note

If the designated parameters change after the creation of objects, neither the associations nor the names of CAD documents or WTParts change.

- Custom

The Object Initialization Rules Administrator, available on the Windchill PDMLink or Windchill ProjectLink **Utilities** tab, provides a way to specify default values for the attributes of a specific object type. The default values are then used when the Windchill solution creates objects of that type. These specifications are called rules. Each rule can contain default values for one object type. The rules that are set only apply when the Windchill solution that is used to create an object does not set a corresponding value. Rules can be setup to provide auto-number generation, but they can also be setup to provide custom behavior (see [Managing CAD Document and WTPart Naming and Numbering on page 111](#)). Rules are also set per context, allowing there to be different naming/numbering policies on different contexts.

Note

Regardless of the naming and numbering policy used, when creating a new object, system uniqueness constraints require that the CAD document attributes Number and File Name must both be unique within a Windchill PDMLink site or within each Windchill PDMLink Workgroup Manager project.

Identifying the Current Naming and Numbering Policy

The algorithm used to understand which policy is currently set in the system (for a particular context and class of object) is as follows:

- If auto-numbering is set in Rules, then the policy is auto-numbering.
- If custom behavior is implemented in Rules, then the policy is custom.
- If neither auto-numbering or custom behavior is set and the parameter or attribute preferences are set in the Windchill **Preference Management** utility, then policy is parameter-driven.
- Otherwise, the policy is name-driven.

Windchill Preferences for Naming and Numbering

The preferences for parameter-driven naming and numbering policy in the Windchill **Preference Management** utility are the following for auto-associate:

- Operation > Auto Associate > Auto Associate Numbering Parameter = <some string parameter>
- Operation > Auto Associate > Auto Associate Naming Parameter = <some string parameter>

Note

Creo Parametric parameters are passed to Windchill in all uppercase characters. The string value must match the name as seen in Creo Parametric for the designated parameter.

The following preferences specify parameter-driven naming and numbering during upload:

- Operation > Upload Operation > Upload > Numbering Parameter = <some string parameter>
- Operation > Upload Operation > Upload > Naming Parameter = <some string parameter>
- By default, none of these four preferences has a value.

Note

The preference Numbering Parameter cannot be used with family table parts that have more than one level of nested instances (upload fails with a uniqueness exception). When Numbering Parameter is used with family table parts that have only one level of instances, values of this parameter need to be different for each instance. You can do this by adding this parameter as a family table column and providing a different number value for each instance.

When set to "Yes" (default is "No"), the following preferences specify dropping the file extensions (such as ".prt" or ".asm") when naming and numbering new objects during an auto-associate action:

- Operation > Auto Associate > Auto Associate Truncate Name File Extension
- Operation > Auto Associate > Auto Associate Truncate Number File Extension

When set to "Yes" (default is "No"), the following preferences specify dropping file extensions during upload:

- Operation > Upload Operation > Upload > Upload Drop Name File Extension
- Operation > Upload Operation > Upload > Upload Drop Number File Extension

Customizing the Naming Service

The Naming service uses the Windchill service delegate mechanism to allow you to specify the following for the new EPMDocument to be created:

- Set a number for the EPMDocument
- Set a name for the EPMDocument

Note

Naming service customization in upload can be used before Windchill 10.0 to generate name and number using custom code. However, before Windchill 10.0, parameters will not be available in an upload request to use in naming service customization.

Note

The Naming service is for the upload action only. The order of precedence used by the system for naming policies and customizations is as follows:

- **Name:**
 1. Naming service customization
 2. Explicitly assigned Common Name through the Creo Parametric **File** ► **New** window
 3. Name parameter (*Operation > Upload Operation > Upload > Naming Parameter = <some string parameter>*)
 4. File Name (*The preference Operation > Upload Operation > Upload > Upload Drop Name File Extension will take effect only if Name is assigned based on File Name (CAD Name)*)
- **Number:**
 1. Naming service customization
 2. Number parameter (*Operation > Upload Operation > Upload > Numbering Parameter*)
 3. File Name (*The preference Operation > Upload Operation > Upload > Upload Drop Number File Extension will take effect only if Number is assigned based on File Name [CAD Name]*)

Use the following steps to customize the Naming service:

1. Create a Java Class that implements the interface EPMDocumentNamingDelegate. The interface definition is as follows:

```
package com.ptc.windchill.uwgm.proesrv.c11n;  
  
public interface EPMDocumentNamingDelegate  
{
```

```
    public void validateDocumentIdentifier(DocIdentifier
docIdentifier);
```

```
}
```

The definition of Class DocIdentifier is as follows:

```
package com.ptc.windchill.uwgm.proesrv.c11n;
```

```
import java.util.HashMap;
```

```
public class DocIdentifier
```

```
{
```

```
{
```

```
    private String m_modelName;
```

```
    private String m_docName;
```

```
    private String m_docNumber;
```

```
    private HashMap m_parameters;
```

```
}
```

```
public DocIdentifier(String modelName, String
docName, String docNumber, HashMap params)
```

```
{
```

```
    m_modelName = modelName;
```

```
    m_docName= docName;
```

```
    m_docNumber= docNumber;
```

```
    m_parameters= params;
```

```
}
```

```
/** get the CAD Name for the model **/
```

```
public String getModelName()
```

```
{
```

```
    return m_modelName;
```

```
}
```

```
/** get the EPMDocument name for the model **/
```

```
public String getDocName()
```

```
{
```

```

    return m_docName;
}

/** set the EPMDocument name for the model */
public void setDocName(String docname)
{
    m_docName = docname;
}

/** set the EPMDocument number for the model */
public void setDocNumber(String docnumber)
{
    m_docNumber = docnumber;
}

/** get the EPMDocument number for the model */
public String getDocNumber()
{
    return m_docNumber;
}

/** get the Pro/E designated parameters for the model. These are
name-value pairs indexed by the name */
public HashMap getParameters()
{
    return m_parameters;
}
}

```

2. In the new class, implement the business logic for naming/numbering EPMDocument in the method:

```
public void validateDocumentIdentifier(DocIdentifier docIdentifier)
```

- The DocumentIdentifier object has the EPMDocument name and number information for the EPMDocument that will be created by the Upload Service.

-
- Use the `DocIdentifier.getModelName()` to get the CAD Name of the EPMDocument that this `DocIdentifier` object represents.
- The Creo Parametric designated parameters may be used to set EPMDocument numbering/naming.
- Use the `DocIdentifier.getParameters()` to get the associated parameters.
- Use the “set” methods on the `DocIdentifier` to set the new name/number values. The Upload Service will use these suggestions if they are feasible.
3. Edit `site.xconf` file (found in `<Windchill>`) to add following property to indicate availability of customization service on the server:
 - `<Service context="default" name="com.ptc.windchill.uwgm.proesrv.c11n.EPMDocumentNamingDelegate" targetFile="codebase/service.properties">`
 - `<Option cardinality="singleton" requestor="wt.epm.EPMDocument" serviceClass="com.ptc.windchill.uwgm.proesrv.c11n.EPMDefaultDocument NamingDelegate"/>`
 - `</Service>`
- Then use the `xconfmanager` tool to apply the changes to `service.properties` file (run `xconfmanager -p`)
- Use the path of your class in place of the value of `serviceClass` (that is, replace `"com.ptc.windchill.uwgm.proesrv.c11n.EPMDefaultDocumentNamingDelegate"` with the path to your class).
4. Restart the method server.

Preferences That Affect Resolution of Incomplete Dependent Objects

While Creo Parametric can display incomplete dependents in the workspace and upload them to the server, Windchill does not allow the check-in of incomplete dependent objects. A preference in the Windchill **Preference Management** utility, **Display ▶ Incomplete object resolution**, can be set to one of four values to allow check-in of an assembly by ignoring certain dependencies or to disallow the ignoring of dependencies (and thereby disallow the check-in of assemblies containing unresolved incomplete dependent objects).

If Incomplete object resolution is set to one of the following values, the behaviors described result:

-
- If set to "Ignore optional dependencies" (default), any CAD tool internal and non-required dependencies are ignored.
 - If set to "Ignore optional reference dependencies," any reference dependencies are ignored.
 - If set to "Ignore internal dependencies only," only internal CAD tool dependencies are ignored.
 - If set to "Do not allow to ignore," no incomplete dependencies can be ignored.

In addition, the following preferences control system behavior toward incomplete dependent objects during the Check In action.

The preference, **Operation ► Check In ► Resolve Incomplete Objects**, controls the default behavior whether or not to resolve incomplete objects automatically upon Check In.

If the preference Resolve Incomplete Objects is set to "yes," or if the **Auto resolve incomplete objects** check box is selected on the **Check In** page, the preference, **Operation ► Check In ► Update Incomplete Objects on Server**, controls the default behavior whether or not to update incomplete objects on server upon resolving incomplete objects upon Check In.

Note

*The server side ghost resolution setting (Incomplete object resolution) is used when a user selects the **Auto resolve incomplete objects** option on the **Check In** page, or in case of using the **Creo Parametric File > Checkin > Auto** command. The setting on the server is not used if the user clears the **Auto resolve incomplete objects** option on the **Check In** page (the user selection not to resolve incomplete objects from the **Check In** page has precedence over the server-side setting). This could result in a valid check-in failure if there are incomplete objects in the check-in list.*

Subtyping CAD Documents

In CAD authoring tools, you can create restricted value parameters that use definitions from a restriction definition file. The restriction definition file defines the parameter name, type, value, range of values and a default value. This feature is useful because it allows you set an attribute for a specific object-type and then set a specific range of acceptable values.

Constraining Attributes

The Windchill counterpart to restricted value parameters is the subtyping feature. Subtyping is accessible through the **Type and Attribute Management** utility and allows you add constraints (such as a value or range of values) to an instance-based

attribute. Additionally, you can use the **Type and Attribute Management** utility to add attributes to the EPMDocument type and its subtypes: both the CAD document and dynamic document (Arbortext document) subtypes.

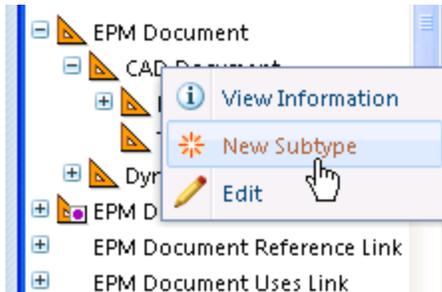
For CAD documents, there is one system-provided subtype, the CAD Document, that an administrator can modify to add attributes that can have different values for each iteration of the object that an administrator can modify. This subtype cannot be deleted. It can also be subtyped see [Flexible Subtyping on page 120](#). There are also additional subtypes, related to CAD documents. They are the following:

- CAD Document Master subtype (on CAD Document Master type)
Attributes that are added to this subtype have only one value for all iterations. Changing the value of an attribute on a CAD Document Master subtype changes that value for all iterations. This type of attribute is the Windchill equivalent of a Pro/INTRALINK non-versioned attribute.
- CAD Document Uses Link subtype (on CAD Document Uses Link type)
Attributes that are added to this subtype are specific to the use of an iteration of an object. For example, if there are four bolts of the same type (bolt.prt) in an assembly, and each bolt needs to be tightened to a specific torque, you can add torque to the Uses Link subtype and then apply a different value to each occurrence of the bolt in the assembly. (In contrast, if you instead add this attribute to the CAD Document subtype, then all bolts in all assemblies would have the same torque wherever they are used.)
- CAD Document Reference Link subtype (on CAD Document Reference Link type)
Attributes that are added to this subtype apply to reference links (again, not to the CAD document, itself).

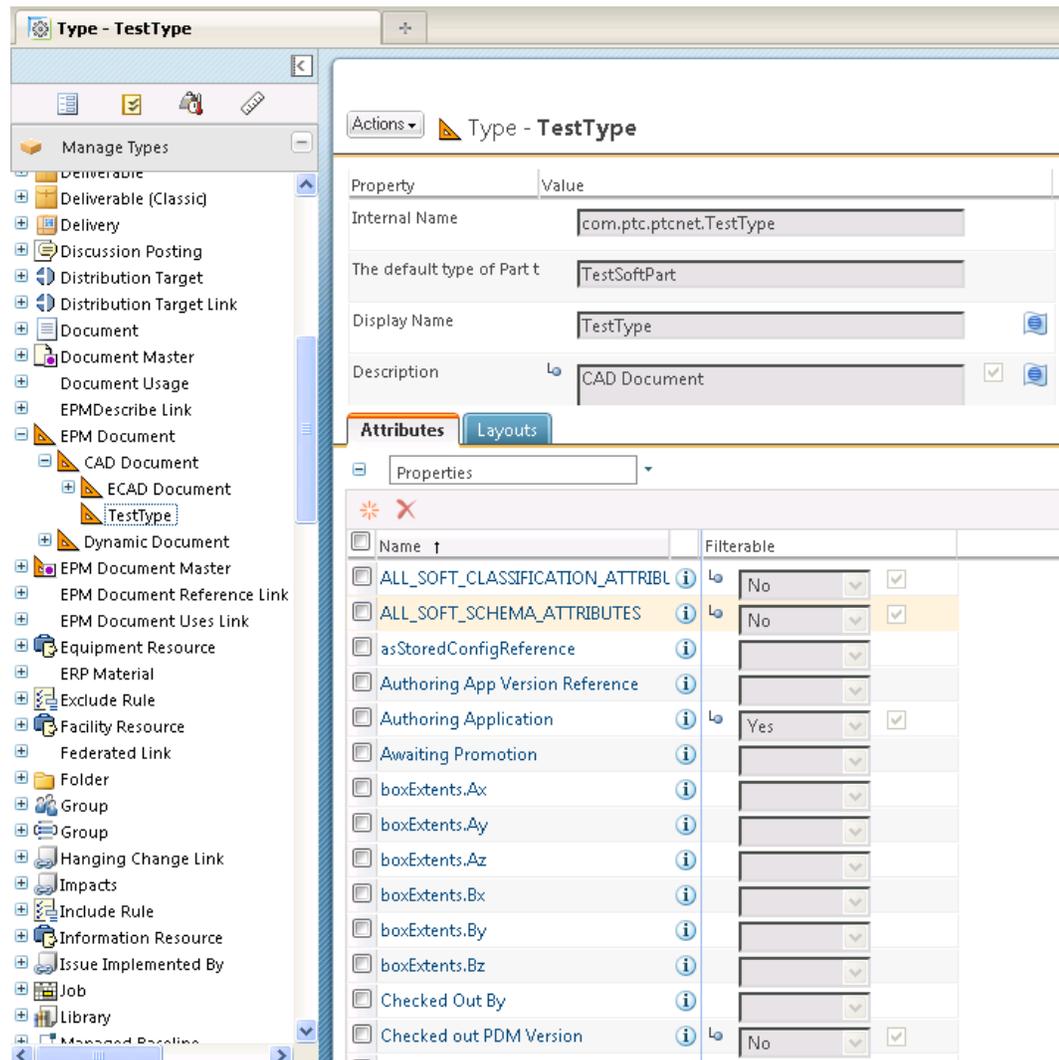
Flexible Subtyping

The term “flexible” subtyping refers to the ability to specify certain CAD document types or subtypes be created when a CAD model is initially checked into Windchill. Combined with the ability to specify what Windchill part subtype is created at auto association (based on the CAD document sub type), it streamlines the process of attribute segmentation among appropriate subtypes.

For example, with administrative privileges, you can access the **Type and Attribute Management** utility, select the CAD Document subtype, and using a right-mouse-button command, create a new subtype for the CAD Document subtype.



Note that in the user interface for creating the new subtype, there is a field to specify the default part subtype to be created for this CAD document subtype



Attributes sets are by default inherited from the supertype, but can be redefined, as required.

Using EPMDDefaultSoftType.xml

For details on subtyping, selecting attributes, and setting constraints, refer to the Windchill Help Center topics on type and attribute management. In addition there is a file, EPMDDefaultSoftType.xml, located at `WT_HOME\codebase\com\ptc\windchill\uwgm\aad\xml`, which can be copied to `WT_HOME\codebase`. The copy of the file can be edited to specify default subtypes. A restart of the method server is required for the edits to take effect.

Example

The default subtypes for the authoring application Creo Parametric are as follows:

```
<AuthAppSoftTypeInfo authAppName="PROE">
<ObjectClassInfo classType="EPMDocument">
<ObjectTypeInfo type="*">
<SoftTypeInfo softTypeId="{ internet_domain_name }.DefaultEPMDocument"/>
<!-- null sub type -->
<SoftTypeInfo subType="*" softTypeId="{ internet_domain_name }.
DefaultEPMDocument"/>
</ObjectTypeInfo>
</ObjectClassInfo>
</AuthAppSoftTypeInfo>
```

where the line `<SoftTypeInfo softTypeId="{ internet_domain_name }. DefaultEPMDocument"/>` is applicable only for the null subtype, and the line `<SoftTypeInfo subType="*" softTypeId="{ internet_domain_name }. DefaultEPMDocument"/>` usually serves as a default mapping for a given type (or all types for which no mapping is explicitly specified if the type = *).

To define a default subtype, use the following procedure:

1. Create a file with the name EPMDDefaultSoftTypes.xml in the `$WT_Home/codebase` directory.
2. Assume that “`{ internet_domain_name }.CreoDoc`” is the subtype name already defined in the **Type and Attribute Management** utility. To use this subtype for all Creo Parametric file types, replace “`{ internet_domain_name }.DefaultEPMDocument`” with “`{ internet_domain_name }.CreoDoc`.” Add the following entry to `$WT_HOME/codebase/EPMDDefaultSoftTypes.xml`:

```
<SoftTypeDescriptor xmlns='http://www.ptc.com'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.ptc.com SoftTypeDescriptor.xsd'>
<AuthAppSoftTypeInfo authAppName="PROE">
<ObjectClassInfo classType="EPMDocument">
```

```

<ObjectTypeInfo type="*" ">
<SoftTypeInfo softTypeId="{ internet_domain_name } .CreoDoc"/> <!-- null
sub type -->
<SoftTypeInfo subType="*" " softTypeId="{ internet_domain_name } .
CreoDoc"/>
</ObjectTypeInfo>
</ObjectClassInfo>
</AuthAppSoftTypeInfo>
<SoftTypeDescriptor>

```

3. Restart the method server.

Command Line Management of Subtyping

A command line utility is provided for you to change the subtype of existing EPMDocuments when moving from an earlier release to X-20. The change of subtype is applied to all iterations of a given EPMDocumentMaster. Constraint validation is only be done for the latest iteration of each revision, and change of subtype will fail if constraints are violated on these iterations.

Behavior of Command Line Utility

The section outlines the behavior of the command line utility. The utility should be executed after upgrade to Windchill 10.0 and before making the upgraded system accessible to all users. Customer should create the required subtypes using **Type and Attribute Management** utility and then use the utility to change subtypes of existing EPMDocuments.

This utility has two modes:

- **Batch Mode:** A csv file will be processed in this mode. When used in this mode collection of EPMDocuments will be processed at a time. The criteria for updating the EPMDocuments with new softtype will be specified in the csv file. The command to use the utility in this mode is:

```
java wt.epm.util.SoftTypeChangeUtility <.csv filename>
```

- **Single Document Mode:** A single EPMDocument will be processed in this mode. The command used in this mode will be:

```
java wt.epm.util.SoftTypeChangeUtility <Document CADName>
<Logical ID of new SoftType>[ contextName]
```

In both the modes of operation, the new subtype specified must be a descendent of the “CAD Document” sub-type. If the new subtype is not a descendent of “CAD Document” then the operation will be aborted

Behavior of the command line utility in Batch mode

To operate in batch mode, you create a comma-separated (.csv) file in which you specify:

- The combinations of values of the attribute triplet: authoringApplication, EPMDocumentType, and EPMDocumentSubType
- The internal name of the subtype that is to be set on documents that belong to a particular triplet

You can specify an asterisk symbol '*' for EPMDocumentType and EPMDocumentSubType. For example, if '*' is specified for the parameter EPMDocumentSubType then all the documents of any EPMDocumentSubType that match the specified authoringApplication and EPMDocumentType will be selected. '*' cannot be specified for the authoringApplication

The EPMDocumentSubType can be left empty to select EPMDocuments that have EPMDocumentSubType as NULL

The format for an entry in the .csv file is as follows:

```
<authoringApplication><EPMDocumentType><EPMDocumentSubType><Logical ID of the new SoftType>
```

Values for the fields of each entry are as follows

- authoringApplication—Specific authoring application
- EPMDocumentType—Specific EPMDocumentType or '*'
- EPMDocumentSubType—Specific EPMDocumentSubType or '*' or empty string
- New SoftType—Specific logical ID of the new subtype to be assigned.

If the expected values for these parameters are not specified then the operation is stopped.

Note

EPMDocSubTypeRB.rbInfo and EPMDocumentTypeRB.rbInfo located at WT_HOME\src\wt\epm folder give complete information about all doc types and subtypes.

The entries of the .csv file will be processed one row at a time. The entry made first will be processed first. If any subsequent entry selects the same EPMDocument, then its subtype will be changed again with the subtype specified in the row that is being processed.

Upon execution, the change of subtype will be applied to all iterations of selected EPMDocuments. Constraint validation will only be done for latest iteration of each revision of selected EPMDocuments and change of subtype will fail if constraints are violated on any of these iterations.

Behavior of the command line utility in Single Document mode

In single document mode you update one EPMDocument at a time. For each EPM document your command line entry is as follows:

```
<Document CADname><Logical ID of new subtype>[ contextName]
```

Note

“contextName” is optional and only required when an EPMDocument in a project context has to be changed. In the case that the contextName is not specified, the utility will assume that the specified EPMDocument belongs to either a ‘Product’ or a ‘Library’.

Upon execution, the change of subtype will be applied to all iterations of a given EPMDocumentMaster. Constraint validation will only be done for latest iteration of each revision and change of subtype will fail if constraints are violated on these iterations.

Mapping Creo Parametric Parameters to Windchill Attributes

Creo Parametric lets you map Creo Parametric designated parameters onto Windchill attributes. Attribute mapping transfers parametric information from the CAD models created in Creo Parametric to the Windchill system. The attribute mapping can be done as follows:

- By implicit parameter-to-attribute mapping
- By explicit parameter-to-attribute mapping

Implicit Parameter-to-Attribute Mapping

Implicit parameter-to-attribute mapping occurs when there is an attribute in Windchill with a name (all uppercase) identical to the name of a designated parameter in a Creo Parametric model file and there is no conflicting mapping specified on the attribute. When the Creo Parametric model file is uploaded into Windchill as content of a CAD document, the values of the Creo Parametric parameter are transferred to the Windchill attribute. For more information on attribute mapping, see the Windchill Help Center.

Note

Using Creo Elements/Pro 5.0 and later releases of Creo with Windchill 10.0 M030 and beyond it is possible to map designated unit-based parameters (and dimensions) to unit-less Windchill attributes. However, for customers upgrading to Creo Elements/Pro 5.0 from earlier releases the neutral data updates will still need to take place, meaning that a family table upgrade is necessary. Those customers already on Wildfire 5.0 or higher, and who have previously upgraded and remapped to unit-based attribute, will not be required to perform a subsequent neutral data upgrade.

Resolving Type Conflicts Between Creo Parametric Parameters and Windchill Attributes

To avoid upload problems in case of a mismatch between the types of a Creo Parametric parameter and the Windchill attribute to which it is mapped, you can set the following property in the site.xconf file:

- `<Service context="default" name="wt.epm.attributes.EPMAAttributeDelegate" targetFile="codebase/service.properties">`
- `<Option cardinality="singleton" requestor=wt.iba.value.IBAHolder" selector="PROE" serviceClass=wt.epm.attributes.EPMAAttributeDelegateWithWarnings"/>`
- `</Service>`

Setting this property and propagating it using xconfmanager allows the system to ignore the mismatch and continue the upload.

Customizing the Parameters in the Download Service

Windchill provides a server-side delegate that can be used to insert parameters into a Creo Parametric model upon download. This mechanism can be used to pass information from the server down to Creo Parametric, where it can be used like any other Creo Parametric parameter (for example, to place information on drawing forms). Parameters beginning with “PTC” or “PROI” are regarded as reserved system parameters and cannot be propagated by the customization. If they are added in the customization, they are ignored by the download service.

Note

This functionality is applicable to all Windchill Workgroup Managers integrating with 3rd party CAD tools

Note

The customized parameters are provided to the client upon download and, unlike system parameters such as PTC_WM_ITERATION, are not updated in the Creo Parametric session or the local cache after a Windchill operation (for example, check in).

For example, if a customized parameter is assigned the value of the CAD document number, its value is provided to the client upon model download. If the CAD document is later renumbered, the value in the Creo Parametric session or the client cache will not be automatically updated.

The Windchill service delegate mechanism is used to allow the customization. The following steps explain the customization process:

1. Create a Java class that implements the interface ModeledAttributesDelegate. The interface definition is as follows:

```

package com.ptc.windchill.uwgm.proesrv.c11n;
import java.util.Collection;
import java.util.HashMap;
import wt.util.WTException;
public interface ModeledAttributesDelegate
{
    /*
    Implement this API to return list of parameters added by
    customization along with it's type (customization profile of the
    server). For example "WT_CADDOC_NUMBER" custom parameter will
    be of type "String.class" (the java class)
    */
    // getAvailableAttributes() returns
    // HashMap<String, Object> which contains
    // HashMap<Attribute name, Attribute type>
    HashMap getAvailableAttributes();
    /*
    This is the API, invoked by the download service on download, to
    be implemented for the customization. Create and return a
    HashMap where key is input object and value is HashMap of
    parameter name - value pairs that must be propagated to Pro/E
    part represented by the EPMDocument (input object). Use the
    getCADName() API on the EPMDocument to identify the Pro/E part
    */
    // getModeledAttributes(Collection docs) returns
    // HashMap<input object, HashMap<Attribute name, Attribute
    value> HashMap getModeledAttributes(Collection docs) throws
    WTException;
}

```

2. Edit site.xconf file (found in <Windchill>) to add following property to indicate availability of customization service on the server:

```

<Service context="default"
name="com.ptc.windchill.uwgm.proesrv.c11n.ModeledAttributesDele
gate" targetFile=codebase/service.properties">

<Option cardinality="singleton"
requestor="java.lang.Object"
serviceClass="com.ptc.windchill.uwgm.proesrv.c11n.DefaultModele
dAttributesDelegate"/>

</Service>

```

Then use the xconfmanager tool to apply the changes to service.properties file (run xconfmanager -p)

Use the path of your class in place of value of serviceClass (that is, replace com.ptc.windchill.uwgm.proesrv.c11n.DefaultModeledAttributesDelegate with the path to your class).

3. Restart the method server.

Configuring the Build Rule

Windchill uses a combination of Windchill Preference Management utility preferences and Windchill properties to control the following functions during execution of the build rule:

- What attributes to publish from a CAD document to a build target, based on the team template of the target
- Specification of whether to use existing part usage links or create new usage links
- Specification of whether to use existing part usage links or create new usage links
- Enabling or disabling the creation of as stored configurations

Controlling Attribute Publishing

You can set the following preferences to define the attributes that are published to the indicated build targets (these preferences are all listed under the categories EPM Service Preferences > Build Service Preferences)

- Attributes to be published on Link
- Attributes to be published on Master
- Attributes to be published on Occurrence
- Attributes to be published on Part

For each preference, the default value is an asterisk (*), which specifies that all attributes are to be published. Specify the specific attributes to be published by replacing the asterisk with a delimiter-separated list of attributes (or specify no attributes by removing the asterisk).

Note

The preference, EPM Service Preferences > Build Service Preferences > Attributes Delimiter, defines the delimiter that separates the listed attributes. A comma (,) is the default value.

The following table lists preferences for the build service:

Preference	Values	Description
Allowed edit of part structure built by build service	Yes No (default)	In the case where a CAD document is owned by an ECAD application, allows editing of the part structure built by build service
Attributes Delimiter	, (default) <character value>	Identifies the delimiter used in listing attributes to

Preference	Values	Description
		be published
Attributes to be published on Link	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the member link
Attributes to be published on Master	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the master
Attributes to be published on Occurrence	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on an occurrence
Attributes to be published on Part	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the part

Specifying Usage Links

To have the build process use existing usage links, leave the property wt.epm.build.subsumeLinks set to the default value true.

To specify that the build process creates new usage links, set the property wt.epm.build.subsumeLinks to false.

If you want the build process to create new links using your usage link class (a subclass of WTPartUsageLink) set wt.epm.build.linkClass to your usage link class. The default value is wt.part.WTPartUsageLink.

Vetoing Operations Based on Owner Application

By default, the following operations are set for a client-side veto on objects owned by specific applications:

- Add link
- Change folder (Move)
- Check in
- Check out
- Delete IBA
- Delete link
- Delete (object)

- Modify property
- Revise

Each operation is controlled by a property that contains the operation name, and the operation is vetoed for all applications listed in the default value. To enable the operation for objects owned by a specific application, remove that application's name from the default listing.

For example, to enable moving (changing the folder) of objects owned by the Pro/INTRALINK Gateway application, you modify the property `wt.epm.veto.change.folder`. The default value for the property is:

```
OPTEGRAGATEWAY, PROINTRALINKGATEWAY, PROPDMGATEWAY,
WORKMANAGERGATEWAY, IDEASTDM.
```

Therefore, you enable the move operation for Pro/INTRALINK Gateway-owned objects by removing `PROINTRALINKGATEWAY`, from the comma-delimited list, yielding the following setting for the property:

```
wt.epm.veto.change.folder=OPTEGRAGATEWAY, PROPDMGATEWAY, WO
RKMANAGERGATEWAY, IDEASTDM.
```

Note

The property `wt.epm.veto.delimiter` defines the character used to delimit the list of owning applications. The default value is comma (,).

Configuring the Initial Collection of Objects for Actions

The Windchill **Preference Management** utility allows you to set preferences for default collection rules on a general, as well as per-PDM-action, basis. To set preferences for default collection rules, navigate to the appropriate section of the **Preference Management** utility as explained for the following options:

- For general setting of collection rule defaults, navigate to the Display > General Collector category.
- For the Add to Project, Move, or Send to PDM actions, navigate to the Integral Operations > <Action> Collector category.
- For the Delete or Revise actions, navigate to the <Action> Collector category.
- For the Save As action, navigate to either the Save As > From Commonspace Collector category, or the Save As > From Workspace Collector category, as appropriate.
- For other actions, navigate to the Operation > <Action> > Collector category. Collection options are available for the following actions:
 - Add to Workspace and Check Out
 - Check In

- Edit Attributes
- Export from Workspace
- Remove from Workspace
- Rename
- Set State
- Undo Check Out
- Update
- Upload

Within each of the collector categories, you are able to set preferences that determine default collection rules applied when an action is initiated. The following table lists and describes the collection preferences that may be available within the respective categories (not all preferences are available in all categories):

Preference	Values	Description
Include dependent CAD Documents	All Required None	Allows user to specify which dependent CAD documents for the collected CAD documents will be by default added to the collection
Include dependent Documents	All None	Allows user to specify which dependent documents for the collected documents will be by default added to the collection
Include dependent Parts	All None	Allows user to specify which dependent parts for the collected parts will be by default added to the collection
Include related CAD Documents	All Initially Selected Only None	Allows user to specify which CAD documents associated to the collected parts will be by default added to the collection
Include related Documents	All Initially Selected Only None	Allows user to specify which documents associated to the collected parts will be by default added to the collection

Preference	Values	Description
Include related Drawings	All Initially Selected Only None	Allows user to specify which drawings related to the collected CAD documents or parts will be by default added to the collection
Include related Family table objects	All None Initially Selected Only	Allows user to specify which family table objects related to the collected generic or instances will be by default added to the collection
Include related Generics	All None Initially Selected Only	Allows user to specify which generics related to the collected instances will be by default added to the collection
Include related Notes	All None Initially Selected Only	Allows user to specify which notes related to the collected parts will be by default added to the collection
Include related Parts	All None Initially Selected Only	Allows user to specify which parts related to the collected documents, CAD documents, or dynamic documents will be by default added to the collection

Note

Out-of-the-box default settings may vary, depending on the action. You can specify the system default setting by selecting Revert to Default on the Set Preference page.

In addition, within each collection category, the preference, Display collected objects, allows users to specify the way in which collected objects are listed in the table. The options are as follows:

- As a List (default)
- As a Structure (shows the object hierarchy)
- As a Structure with Associated Objects

Configuring Check In

The following sections describe preferences used with the Check In action.

Enabling As Stored Configurations

The property Operation > Check In > Create As Stored specifies by default ("Yes") to create an As Stored configuration at the time objects are checked in. If set to "No," an As Stored configuration is not created.

Enabling Baseline Creation

The property, Operation > Check In > Create Baseline upon Check In, specifies by default ("Yes") to create a baseline at the time objects are checked in. If set to "No," a baseline is not created.

Managing ModelCHECK Validation during Check In

Creo Parametric allows you to use ModelCHECK™ as a "gatekeeper" to the Windchill database, which means that to be successfully checked in, models must meet ModelCHECK criteria. This gatekeeper functionality is controlled by the Windchill server, which references the read-only ModelCHECK parameters contained in the models.

Configuring ModelCHECK in Creo Parametric

To enable the gatekeeper functionality, you must first edit the ModelCHECK configuration to enable ModelCHECK to add the required parameters to the data. From the **ModelCHECK Configuration** window within Creo Parametric, edit the initialization file (config_init.mc) and change the following objects:

- Set **MC_ENABLE** to **Y**
This enables ModelCHECK, and is required even if the config.pro option modelcheck_enabled is set to 'yes.'
- Set **RUN_MODE** to **Y**
This enables individual run modes of ModelCHECK: Interactive, Batch, Regenerate, and Save. At least one run mode must be enabled.
- Set **ADD_DATE_PARM** to **Y**
This creates a parameter called MODEL_CHECK in the model files of all models that are checked. This parameter contains the date and time when ModelCHECK was last run.
- Set **ADD_ERR_PARM** to **Y**

This creates a parameter called MC_ERRORS in all models that are checked. This parameter contains the number of errors found in the model when ModelCHECK was last run.

- Set **ADD_CONFIG_PARM** to Y

This creates a parameter called MC_CONFIG in all models that are checked. This parameter contains the names of the ModelCHECK configuration files used for a final check of the model.

- Set **ADD_MODE_PARM** to Y

This creates a parameter called MC_MODE in all models that are checked. This parameter contains the mode in which ModelCHECK was run on the model.

- Set **ADD_VERIFIED_PARM** to Y

This allows the creation of a parameter called MC_VERIFIED in all models that are checked.

After editing the settings, save the configuration. As a result of this change, anytime you run ModelCHECK, these new read-only parameters are added to the data files.

Note

For more information, please see the ModelCHECK Help Topic Collection documentation.

Configuring ModelCHECK in Windchill

After configuring ModelCHECK in Creo Parametric, configure Windchill as follows:

1. In the Windchill **Type and Attribute Management** utility, expand the **Manage Types** node and find and select the CAD Document subtype of EPM Document. Click **Edit** from the **Action** menu for the type to enter edit mode, and click the new attribute icon  on the **Attributes** tab to access the **New Attribute** window and proceed to create attributes with following names and attribute types:
 - MC_ERRORS – (integer)
 - MODEL_CHECK – (string)
 - MC_CONFIG – (string)
 - MC_MODE – (string)
 - MC_VERIFIED – (boolean)

Note

If you click **Apply** after creating an attribute, the **New Attribute** window stays open for creation of the next attribute. When you have created all the attributes, click **Done** to exit the **New Attribute** window. You are returned to the edit mode for the CAD Document type, with the information page for the last-created attribute open in the right-hand pane. For more information, see the Windchill Help Center topic, “Creating a New Attribute”.

2. When you are done creating the attributes, in the attribute information page for the last created MC_ <ERRORS/CONFIG/MODE or VERIFIED> attribute, select the **Visibility** tab. In the **Screen Type** column find the screen types, Create New, Edit, and EPM Upload, and set the value for each of the screen types to **Hidden**. Then click **Save**.
3. Repeat the setting of **Hidden** in the screen types of Create New, Edit, and EPM Upload for the remaining MC_ <ERRORS/CONFIG/MODE or VERIFIED> attributes. When all four attributes have had their visibility values set, click **Done** to exit the edit mode of the CAD Document type page. You can then close the **Type and Attribute Management** utility. For more information on attribute visibility, see the Windchill Help Center topic, “Viewing and Setting Attribute Visibility.”
4. Set the preference Operation > Check In > ModelCHECK Validation to yes (the default is no) to enable ModelCHECK.
5. Set the appropriate modelcheck preferences (also in Operation > Check In) to configure ModelCHECK, as follows:
 - Set **ModelCHECK Number of Errors** to specify the maximum number of ModelCHECK errors allowed. The default is 0.
 - Set **ModelCHECK Number of Hours** to specify the maximum allowable hours between a ModelCHECK verification at the client and the actual model checkin to Windchill. The default is 24.

Note

Wildfire 4.0 M100 and Windchill PDMLink 9.1 M030(1) change the behavior of ModelCHECK Gatekeeper and mark files as out of date with ModelCHECK based on the Workspace status, not a time increment. The status is stored in the MC_VERIFIED parameter/global attribute when available. **ModelCHECK Number of Hours** is still required for any software combination where at least one component is using an older date code.

- Set **ModelCHECK Mode** to specify the run mode used to execute ModelCHECK: Disabled, Interactive (default), Regenerate Explicit, Regenerate Implicit, Regenerate Always, Save, or Batch.

Note

Prior to PDMLink 9.1 M030(1) ModelCHECK Mode is not a valid ModelCHECK Gatekeeper option and is not available to be set.

Executing VDA Checks will set the MC_MODE value to MC_VDA. This value of MC_MODE is not supported by ModelCHECK Gatekeeper.

- Set **ModelCHECK Configuration** to specify the ModelCHECK Configuration files to be used for validation for each LifeCycle name in a specific syntax.

(For example:

```
<Lifecycle_1>:<mch_file1>,<mcs_file1> <Lifecycle2>:<mch_file2>,<mcs_file2>.....
```

The configuration specified by the "Default" life cycle state is fallback behavior. Typical examples are as follows:

- Default:check/default_checks.mch,start/nostart.mcs,constant/inch.mcn
- Basic:check/basic_checks.mch,start/basic_start.mcs,constant/inch.mcn
- Release:check/release_checks.mch,start/release_start.mcs,constant/inch.mcn
- Approval:check/approval_checks.mch,start/approval_start.mcs,constant/inch.mcn
- Review:check/review_checks.mch,start/review_start.mcs,constant/inch.mcn

Note

You need to configure the ModelCHECK conditions in Creo Parametric to obtain the appropriate configurations for the respective LifeCycle Name. A typical example to configure condition.mcc is as follows:

Example: Edit condition.mcc:

- IF (PTC_WM_LIFECYCLE EQ Basic) config=(check/basic_check.mch)(start/basic_start.mcs)(constant/inch.mcn)(status/basic_status.mcq)
- IF (PTC_WM_LIFECYCLE EQ Release) config=(check/release_check.mch)(start/release_start.mcs)(constant/inch.mcn)(status/release_status.mcq)
- IF (PTC_WM_LIFECYCLE EQ Approval) config=(check/approval_check.mch)(start/approval_start.mcs)(constant/inch.mcn)(status/approval_status.mcq)

-
- IF (PTC_WM_LIFECYCLE EQ Review) config=(check/review_check.mch)(start/review_start.mcs)(constant/inch.mcn)(status/review_status.mcq)
 - ELSE set the fallback (CADDocument OIR - LifeCycle Name at context level)
 - ELSE config=(check/basic_check.mch)(start/basic_start.mcs)(constant/inch.mcn)(status/basic_status.mcq)

Note

The Windchill OIR (Object Initialization Rule) pertaining to a CAD document for a context should be examined for the default LifeCycle Name. The fallback value of conditions.mcc for a workspace should match that of the OIR Lifecycle Name.

Enabling Support for Custom Parts

In the Creo Parametric HTML client, you can enable support for custom parts, which extend wt.part.WTPart; however, a custom part must be modeled before any changes are made to the Creo Parametric HTML client. (For more information on extending the Windchill object model, see the *Windchill Specialized Administration Guide*.)

The Creo Parametric HTML client permits use of custom parts in most operations, including download, check out, check in, associate, disassociate, and so on; however, the operations used to create parts, **New ▶ Part** and **Auto Associate Part**, are specific to WTPart. Additionally, when you view the properties of a custom part, any global attributes you may have added to the custom part can be seen; however, newly modeled information is not displayed.

Note

This functionality is applicable to all Windchill Workgroup Managers integrating with 3rd party CAD tools

Whenever "Part" is available in the object type list on the Creo Parametric HTML client object selection page, if "Part" or "All" is selected, both WTPart objects and custom part objects are listed in the page's results table.

Automatic part generation is supported through the **Auto Associate Part** action available on the workspace properties page. To enable automatic custom part generation when using this command, however, you must either create or modify your automatic part creator. For more information, see [Customizing Auto Associate on page 140](#).

Modifying the Properties Page

To configure a custom part-specific properties page you have to create a properties page and/or template processor. For details on how to do this, see the *Windchill Specialized Administration Guide*.

Modifying the HTML Client Object Selection Page

To enable recognition of custom parts as a sub-class of WTPart and not just the supported type in the Creo Parametric HTML client object selection page's default implementation, you must add support for the custom part in the configured `wt.query.SearchAttributeListDelegate`. (For more information see the section, [Customizing the HTML Client Object Selection Page on page 144.](#))

In addition you must modify the Creo Parametric HTML files that use the object selection page, and use the `xconfmanager` to modify or override the type list id entries in `com\ptc\windchill\cadx\propfiles\picker.properties`.

Note

For `wt.query.SearchAttributeList`, which is the default configured search attribute list, the type id is referred to as the query value. (For more information, see [Customizing the HTML Client Object Selection Page on page 144.](#))

Replacing WTPart

If you want your site to only use custom part and not WTParts, then do the following:

1. Add custom part support to HTML Search.
2. In `picker.properties`, use the `xconfmanager` to change the type list entries that contain a type id for WTPart to the custom part type id you created in Step 1.
3. Restart the method server.

Supporting WTPart and Custom Part

If your site will be using both WTParts and custom parts, then do the following:

1. Add custom part support to HTML Search.
2. In `picker.properties`, use the `xconfmanager` to add to the type list entries that contain a type id for WTPart the custom part type id you created in step 1.
3. To add an "All" type list entry for a type list, add an entry with the ALL type id used by the configured search attribute list.
4. Restart the method server.

Administering Revision

Administrators can configure how the system behaves during a revision operation. Server side settings can determine whether to:

-
- Allow revision to a level other than the next in the revision scheme
 - Create or maintain passive associations during a revision action
 - Synchronize revision levels of CAD documents and parts during an autoassociate action

Note

The revision level synchronization behavior described in the section, [Configuring the Synchronization of Revision Levels During Auto Associate](#) on page 139, also applies to the revise action when `AutoAssociateSetRevisionForWTPart` is set to true.

The following sections describe each of these configurations.

Configuring the Ability to Set a Revision Level

The preference, **Revise ► Allow Override on Revise**, allows setting the target revision of a object by adding a **Select Revision** control to the **New Revision** page. The preference, **Revise ► Allow Override On Create CAD Document**, allows setting the target revision of a object by adding a **Set Revision** control to the **New CAD Document** user interface. When either of these preferences is set to "Yes," the user is allowed to set the target revision of the object in the respective user interface. When the value is "No," the object is revised to the next revision level in its series. The default is "No."

Properties for New Revisions

The following is a summary of the properties for new revisions:

- By default, all new revisions should be created in the same location (context and folder) as the original.
- If there is a user interface, the user can override the default location and choose to place the new revision somewhere else.
- For Windchill PDMLink, the team and life cycle are determined by the object initiation rules of the context.
- The view of new parts defaults to the same view as the original.

Configuring the Synchronization of Revision Levels During Auto Associate

A server-side preference, **Operation ► Auto Associate ► Set Revision For Part**, allows you to set the behavior for the revision of CAD documents and parts during the **Auto Associate** operation.

Set Revision For Part can be set to the values "Yes" or "No" (default). When set to "Yes" the revision of a WTPart is set to that of the actively associated CAD document during an auto-associate action.

Auto Associate attempts to set the part's revision to match that of the CAD document when both the following situations apply:

- When an active association is to be created between the part and the CAD document
and
- Only when the auto-associate action creates a new part for association. A matching revision cannot be set to the working copy of a part. Even if the part is initially checked in, because **Auto Associate** checks out the part before creating an active association, a matching revision is not set to such an existing part.

The following rules apply to both the autoassociate and associate (revising both part and CAD document) actions when Auto Associate > Set Revision For Part is set to "Yes."

- A revision matching that of the CAD document is set to the part when the CAD document revision is higher than that of the part.
- A revision matching that of the CAD document is not set to the part if:
 - The CAD document revision is lower than the part
 - The CAD document and part revisions do not belong to the same revision series

If Set Revision For Part is set to "No," Auto Associate continues without trying to set a revision level.

Customizing Auto Associate

Auto Associate functionality can be customized in the following ways:

- Modifying the implementation of the AutoAssociatePartFinderCreator interface
- Modifying the implementation to search for Customized parts or custom parts
- Customizing the Type of CAD documents that can be actively associated
- Preventing the creation of Parts by Auto Associate
- Controlling the default location of parts created by Auto Associate

Note

This functionality is applicable to all Windchill Workgroup Managers integrating with 3rd party CAD tools

Each of these customizations is described in the following sections.

Note

Preferences that control naming and numbering of parts created during Auto Associate are discussed in the section, [Windchill Preferences for Naming and Numbering on page 114](#), and also listed in the table of Auto Associate preferences in the section, [Operations Preferences on page 163](#), along with the preferences discussed in the following sections on Auto Associate.

Using and Modifying the AutoAssociatePartFinderCreator Interface

AutoAssociate uses the implementation of the AutoAssociatePartFinderCreator interface to perform the following actions:

- To search a for matching part
- To create a new part

By default, the AutoAssociate action uses the default implementation of this interface to perform the above-mentioned tasks; however, you can customize the how they are performed using a customized implementation of AutoAssociatePartFinderCreator interface.

The interface is located in: com.ptc.windchill.cadx.autoassociate.
AutoAssociatePartFinderCreator.

The AutoAssociatePartFinderCreator interface supports the following methods:

- findOrCreateWTPart method used to search for matching part
- CreateNewWTPart method used to create new part
- findWTPart method (no longer used)
- isNewPart method (no longer used)
- setIsNewPart method (no longer used)

Note

Even though some methods of the interface are deprecated and no longer used, the implementation class should have dummy implementations of these methods in order to compile the class.

Use the following procedure to implement a customized AutoAssociatePartFinderCreator:

1. Derive your customized class as follows:

```
public class CustomFinderCreator implements  
AutoAssociatePartFinderCreator
```

2. Override the following methods:

- `public WTPart findOrCreateWTPart(EPMDocument epmDoc, EPMWorkspace workspace)`

This method is invoked for each document selected for auto-associate to search for any matching part. You can customize the criteria used to search the part, and the returned part is used by the action to associate to the document.

- `public WTPart createNewWTPart(EPMDocument doc, String partNum, String partName,)`

This method is invoked for each document selected for auto-associate to create a new part. You can customize the properties of the newly created part. The newly created part is associated to the document by the auto-associate action.

Note

The following methods are deprecated and not currently used by the action; however, you need to provide a dummy implementation of these methods to compile the class properly.

- `public boolean isIsNewPart()`
- `public void setIsNewPart(boolean a_IsNewPart)`
- `public WTPart findWTPart(EPMDocument epmDoc)`

Compile the file and place the class in any appropriate location

3. Set the preference **Operation ▶ Auto Associate ▶ Custom Class for Auto Associate Part** to specify the name of the class that implements `AutoAssociatePartFinderCreator` interface.
4. Restart the method server

Modifying the Implementation to Search for Customized Parts or Custom Parts

When performing searches, the default implementation is to search for a `WTPart`.

Note

When you create a customized part, its master must be `WTPartMaster` or a subclass of `WTPartMaster`. The customized part itself must be a `WTPart` or a subclass of `WTPart`.

To customize the implementation to search for a customized part that has been implemented in the codebase (for example, `wt.part.MyCustomPartMaster`), set the preference, **Operation ▶ Auto Associate ▶ Custom Class for Auto Associate Part** to `wt.part.MyCustomPartMaster`

Controlling the Associations Formed by Auto Associate

Several preferences affect the type of associations formed during Auto Associate and the type of CAD documents that are allowed to form them.

Note

1. Model items with .prt and .asm extensions are not subject to the following preferences. If you want to add or remove a model item type that is valid for association, then you need to explicitly identify all types or sub-types that should associate.

The preference, **Operation ▶ Auto Associate ▶ Disallow Product Structure Links for CAD Document Types**, allows you to specify the CAD document types which cannot form an Owner association. These are comma-separated values. The default is <no value>.

The preference, **Operation ▶ Auto Associate ▶ Disallow Product Structure Links for CAD Document Sub-Types**, allows you to specify the CAD document sub-types which cannot form an Owner association. These are comma-separated values. The default is <no value>.

Note

*The allowable values for the preferences, **Disallow Product Structure Links by Document Types** and **Disallow Product Structure Links by Document Sub-Types**, are listed in the table of Auto Associate preferences in the section [Operations Preferences on page 163](#).*

The preference, **Operation ▶ Auto Associate ▶ Create Alternate Link On Check In**, when set to "Yes," allows a CAD-document-to-part association of the next available type (that is, Content) to be created if the matching part found during Auto Associate already has an Owner association, and allows the checkin to continue. The default is "No" (no Content association is formed and the check in fails with an overridable conflict).

Controlling the Creation of Parts by Auto Associate

The preference, **Operation ▶ Auto Associate ▶ Create Associate New Part**, specifies whether a new part should be created if a matching part is not found by Auto Associate. The default is "Owner Only" for all CAD tools. ECAD authoring applications default to "All".

Possible values are:

- **Owner Only:** If a matching part is not found, a new part is created when the CAD document would associate to a part with an "Owner" association.
- **Owner and Contributing Image:** If a matching part is not found, a new part is created when the CAD document would associate to a part with either an "Owner" or "Contributing Image" association.

-
- All: If a matching part is not found, a new part is created when the CAD document would associate to a part with any product structure association ("Owner", "Contributing Image", and "Image").
 - Never: A new part is not created if an existing part is not found, even if it contributes to product structure. Auto associate does not fail, the CAD document is skipped, and other selected CAD documents will try to associate.

Controlling the Default Location of Parts Created by Auto Associate

The preference, **Operation ▶ Auto Associate ▶ Store New Parts with CAD Documents**, when set to "Yes," specifies that the storage location of new part created during Auto Associate be the same as its associated CAD document. By default, the preference is set to "No."

Customizing the HTML Client Object Selection Page

The HTML client object selection page is used in the Creo Parametric HTML client to allow the user to choose objects in the Windchill database that are required to complete an action.

To determine the drop down list, search criteria, and result columns for the object selection page the configured `com.ptc.windchill.cadx.common.picker.PickerSearchAttributeListDelegate` is used. The default configured `PickerSearchAttributeListDelegate` is `com.ptc.windchill.cadx.common.picker.PickerSearchAttributeList`. `PickerSearchAttributeList` delegates to the configured `wt.query.SearchAttributeListDelegate` to create the various type lists on the object selection page will be configured to support and determine the search criteria, and determine the result columns displayed in the object selection page. (For more `SearchAttributeListDelegate` details see [Customizing the HTML Search on page 146.](#))

If this `PickerSearchAttributeListDelegate` implementation is not sufficient, then you can create and configure your own `PickerSearchAttributeList` to be used by the object selection page.

Note

This functionality is applicable to all Windchill Workgroup Managers integrating with 3rd party CAD tools

Modifying the Search Attribute List Delegate

To implement your own custom `PickerSearchAttributeListDelegate`, create a class that implements `wt.query.SearchAttributeListDelegate` and `com.ptc.windchill.cadx.common.picker.PickerSearchAttributeListDelegate` or create a class which sub-

classes `com.ptc.windchill.cadx.common.picker.PickerSearchAttributeList`. See the javadoc for `PickerSearchAttributeListDelegate` and `PickerSearchAttributeList` and their methods for more details.

Note

PickerSearchAttributeList extends SearchAttributeList; therefore, the custom class can be used as the SearchAttributeListDelegate and PickerSearchAttributeListDelegate.

Note

If extending PickerSearchAttributeList, you may have to set the filter to avoid NullPointerExceptions. This issue will be addressed in a future release.

To configure a new `PickerSearchAttributeListDelegate`, use the `xconfmanager` to add an entry to `com/ptc/windchill/cadx/common/picker/picker.properties` similar to:

```
wt.services/svc/default/com.ptc.windchill.cadx.common.picker.  
PickerSearchAttributeListDelegate/  
<unique delegate id which is also specified for com.ptc.windchill.cadx.common.  
picker.pickerSearchAttributeList> /java.lang.Object/0=mine.  
MyPickerSearchAttributeList/duplicate.
```

Using the `xconfmanager`, change the `pickerSearchAttributeList` entry in the `wt.properties` to `com.ptc.windchill.cadx.common.picker.pickerSearchAttributeList=<unique delegate id>`. If there is no entry in `wt.properties`, then `STANDARD` is used as the delegate id

Modifying Type Lists

The Creo Parametric HTML client object selection page uses configured type lists identified by type list ids, which are specified as the object selection page `typeListID` property value.

Type lists are defined in `com\ptc\windchill\cadx\propfiles\picker.properties`.

To add a type list entry for a new type list id, use the `xconfmanager` to add an entry similar to:

```
wt.services/rsc/default/<type list id>/java.lang.Object/0=<comma-seperated list of  
valid query values>
```

If there is only one value in the list, then you do not need any commas. If you want an “All” entry in the type list, you must specify the type list entry value for `ALL` in the list of type ids.

Note

For the default implementation of the object selection page these valid type list values are query values specified in `wt.query.queryResource`.

You can remove type ids from the list of type ids specified for a type list id, but you cannot remove an entry or leave the type list empty.

Customizing the HTML Search

To customize the HTML search to either change the display of the default search objects or to add new classes, see the following file that is distributed as source `Windchill\src\wt\query\SearchAttributeList.java`. As explained in the javadoc for this class, subclass `SearchAttributeList` and make the appropriate entries in service properties and `wt.properties`. Following are methods that should be implemented in a custom `SearchAttributeList`, with examples:

```
public final class MySearchAttributeList extends SearchAttributeList implements
Externalizable {
public void setLocale( Locale locale ) {
// Load in the values for the drop down list for selecting what to search against.
clientLocale = locale;
// **Customize -----
-----
// Add new classes to search to list below.
// Make sure that they are assigned numbers in sequence from 0 to N.
// Set dropDownListCount to N+1.
final int ALL = 0;
final int WTPART = 1;
...
final int MYCLASS = 22
int dropDownListCount = 23;
// -----
-----...
pickList = new String[ classCount ] ;
pickList[ ALL ] =
WTMessage.getLocalizedMessage (RESOURCE, queryResource.ALL, null, clientLocale);
pickList[ WTPART ] =
WTMessage.getLocalizedMessage (RESOURCE, queryResource.WTPART, null, clientLocale);
...
pickList[ MYCLASS ] = WTMessage.getLocalizedMessage (RESOURCE, queryResource.
MYCLASS, null, clientLocale);
pickValues = new String[ classCount ] ;
pickValues[ ALL ] = queryResource.ALL;
pickValues[ WTPART ] = queryResource.WTPART;
...
pickValues[ MYCLASS ] = queryResource.MYCLASS;
// **Customize You will need a string in here to correspond to each item in
pickList
// The string is a space separated list of what classes to query
// against. If you want to query against multiple classes that have a common
parent that
// has all of the attributes that you are interested in use that one class. If
you want
// to query against multiple classes that don't have a good common parent then
you can
// add them to a list and the search will loop through each class and combine
the results
// at the end. All classes in one list must only search against COMMON
attributes or
```

```

// attributes with the same name and of the same class! If you add both a
parent and
// a child class to the list you will get duplicate entries, when the results
are
// combined duplicate entries are not deleted.
queryClass = new String[ classCount];
queryClass[ ALL] =
"wt.part.WTPart wt.doc.WTDocument wt.change2.WTChangeIssue
wt.change2.WTChangeRequest2 " +
"wt.change2.WTChangeInvestigation wt.change2.WTAnalysisActivity
wt.change2.WTChangeProposal " +
"wt.change2.WTChangeOrder2 wt.change2.WTChangeActivity2
wt.csm.businessentity.BusinessEntity " +
"wt.effectivity.ConfigurationItem wt.epm.EPMDocument " +
"wt.replication.unit.WTUnit " +
"wt.part.WTPProductConfiguration " +
"wt.part.WTPProductInstance2 "; // Please remember to keep a space at the
end so that conditionally added items work.
...
queryClass[ WTPART] = "wt.part.WTPart";
...
queryClass[ MYCLASS] = "??.?.MyClass";
// **Customize These are the
// attributes that can be queried against.
inputAttributes = new String[ classCount];
inputAttributes[ ALL] =
"number name lifeCycleState projectId cabinet creator modifier
modifyTimestamp";
inputAttributes[ WTPART] =
"number name view versionIdentifier partType source lifeCycleState projectId
cabinet creator modifier modifyTimestamp";
...
inputAttributes[ MYCLASS] =
"name modifyTimestamp";
// **Customize Each individual
// string must match with the string listed above for the inputAttributes. "0"
stands for no
// input processing. If an attribute is an enumerated type use "0" and the
code will generate
// the drop down list. In the first string: projectId is in the fourth
position in inputAttributes
// so the method to generate the drop down list for it is also in the fourth
position in the
// string. The "0"s and methods must match in number with the number of
attributes listed
// under inputAttributes. You may add a fully qualified method from your
customization package
// as long as it is static and returns a vector of strings.
inputProcessing = new String[ classCount];
inputProcessing[ ALL] =
"0 0 0 wt.query.LocalSearchProcessor.getProjectList
wt.query.LocalSearchProcessor.getCabinetList 0 0 0";
inputProcessing[ WTPART] =
"0 0 wt.query.LocalSearchProcessor.getViewList 0 0 0 0
wt.query.LocalSearchProcessor.getProjectList
wt.query.LocalSearchProcessor.getCabinetList 0 0 0";

```

```

...
inputProcessing[ MYCLASS] =
"0 0";
// **Customize This is similar in concept to inputAttributes only these are
the attributes
// that will be displayed in the search results.
outputAttributes = new String[ classCount];
outputAttributes[ ALL] =
"number name versionDisplayIdentifier displayType lifeCycleState projectId
modifyTimestamp";
outputAttributes[ WTPART] =
"number name versionDisplayIdentifier projectId lifeCycleState
modifyTimestamp";
...
outputProcessing[ MYCLASS] =
"ObjProps 0";
// **New for 6.0
// **Customize This is similar in concept to outputAttributes only this list
is used
// to indicate which attributes can be sorted, can't be sorted, or an alternate
attribute
// that can be sorted to have the same affect as the display attribute. The
string that is used
// here should be the column descriptor so that it can be used to create the
ClassAttribute for
// the query. The query that is used for search is a simple query that will
not sort on all
// of the display attributes. Changing the 0 to 1 for an unsupported attribute
will
// either cause exceptions or sorts that don't work. Attributes of the
following types are
// just some examples of the attributes that will either throw exceptions or
sort incorrectly:
// EnumeratedType, CabinetReference, DataFormatReference,
LifeCycleTemplateReference, ProjectReference,
// and ViewReference.
sortAttributes = new String[ classCount];
sortAttributes[ ALL] =
"1 1 versionInfo.identifier.versionId 0 0 01";
sortAttributes[ WTPART] =
"1 1 versionInfo.identifier.versionId 0 0 1";
...
sortAttributes[ MYCLASS] =
"1 1";
// **New for 6.0
// **Customize This is similar in concept to outputAttributes only this list
is used
// for assigning a unique key to the sort preferences for this search. This
string will
// be persisted and used to retrieve the sort preferences for users. If the
value of one
// of these strings is changed or deleted after the system is in operation it
will create orphaned
// preferences in the system and users will lose the value that they had
persisted for that
// search. New entries can be added when a new search is added so that sort

```

```

preferences
// can be saved for that new search. These strings are arbitrary and never
// displayed to the user.
sortPref = new String[ classCount ] ;
sortPref[ ALL ] =
"all";
sortPref[ WTPART ] =
"wtpart";
...
sortPref[ MYCLASS ] =
"myclass";
}
/**
 *
 * <BR><BR><B> Supported API: </B>>false
 *
 * @param locale
 * @return MySearchAttributeList
 **/
public MySearchAttributeList( Locale locale ) {
setLocale(locale);
}
/**
 *
 * <BR><BR><B> Supported API: </B>>false
 *
 * @return MySearchAttributeList
 **/
public MySearchAttributeList() {
return;
}
}

```

wt.query.SearchAttributeList is always the most up-to-date and should be used as a reference.

The remainder of this section describes two new arrays in wt.query.
SearchAttributeList: sortAttributes and sortPref.

Due to the data structures used on some classes, not all attributes that can be displayed in search results are sortable in the search results. The sortAttributes array in wt.query.SearchAttributeList is used to designate which attributes are sortable, and if an alternate attribute should be used for sorting. The version attribute is an example of an alternate attribute used for sorting: the attribute used to display is versionDisplayIdentifier, but the attribute used to sort on is versionInfo.identifier.versionId. Base java types, such as String and int, are sortable. Use the examples in wt.query.SearchAttributeList to determine if any custom types are sortable. Otherwise, a simple test will show if the attribute works, has no effect, or throws an exception.

The sortPref array (shown in the preceding code) is used to define a sort preference base name so users can define their sort preferences for that “Search On” object. A default for the sort preferences should be defined at the system level so that the first time the user uses the system, or if a user never defines preferences, the

columns will be sorted logically. A default can be defined using `wt.load.LoadFromFile` or by using the Preference Administrator editor from the System Administrator portal page.

If this is a new database, the defaults are loaded as part of running the required section of `wt.load.Demo` (which runs `wt.load.LoadFromFile`). The site defaults can easily be added to or modified using the Preference Administrator. If the database was created on a system before Release 6.0, `wt.load.LoadFromFile` can be used to load the base defaults for the delivered configuration of the HTML search classes. See the “PrefEntry.../wt/query/htmlsearch” entries in `Windchill\loadFiles\preferences.txt` as examples.

Each user preference has an internal name, which is never seen from the client except in the Preference Administrator. Because the current search uses the `wt.query.SearchAttributeList` to allow users to add new searches, and because there has to be a set of sort preferences for each, a unique sort name is needed for each name in the "Search On" list. Each object in the “Search On” list is not necessarily one object, but can be a list of objects. The `sortPref` array in `wt.query.SearchAttributeList` defines a unique string that forms part of the name of the preference. The preferences for sorting are stored in the `/wt/query/htmlsearch` preference node, and the naming format is as follows:

```
<sort preference base name>sortAttrib<#>
```

```
<sort preference base name>sortDirect<#>
```

The `<sort preference base name>` is the unique string from the `sortPref` array in `wt.query.SearchAttributeList`; it has only to be unique within the sort names. The `sortAttrib` is for the attribute name, and the `sortDirect` is to indicate ascending or descending. It is false for ascending and true for descending. The `<#>` is the number of the sort key, 0 = first key, and so on. Following are the preferences that are loaded using `wt.load.LoadFromFile` and `Windchill\loadFiles\preferences.txt` for the All sort:

```
#All
```

```
PrefEntry~allsortAttrib0~number~/wt/query/htmlsearch
```

```
PrefEntry~allsortDirect0~false~/wt/query/htmlsearch
```

```
PrefEntry~allsortAttrib1~versionInfo.identifier.versionId~/wt/query/htmlsearch
```

```
PrefEntry~allsortDirect1~true~/wt/query/htmlsearch
```

In the all-default example, the results are sorted first by the number column and then by the version column, with the number being in ascending order and the version in descending order. Currently, the supported number of sort keys is 3, although theoretically the number of sort keys is limited only by Oracle performance. No testing beyond 3 keys has been done on the system.

Managing Secondary Content

The primary content of a CAD document is a CAD model file; however, Windchill allows you to attach other file types as secondary content. In addition, you can specify which file types should be considered outdated, and which should be automatically downloaded with a download of the primary content, as the CAD document moves through the stages (iterations, revisions, life cycle states, and so forth) of development.

Specifying Whether to Outdate Secondary Content

The preference category **Operation ▶ CAD Data Management ▶ Content Handling ▶ Mark Out Of Date** lists the secondary content categories for your site. For each category of file type, if the preference `<secondary_content_category>` (for example, Instance Accelerator File) is set to yes, it means that upon checking in a content change (not a metadata change) of the CAD document, any secondary content of the specified category type is marked as outdated (for example, in the **Attachments** table on the CAD document information page). A value of no means that the category does not become outdated as it is carried forward with the CAD document.

Users can manually override conflicts caused by outdated files (for example, during a Check In attempt) by:

- Stopping the checkin and manually updating the attachment, thus removing the “outdated” flag
- Removing the attachment from the CAD document
- Resetting the status (removing the Outdated flag)
- Overriding the conflict and checking the CAD document in “as is” – with an Outdated status.

Setting the Preference to Automatically Download Secondary Content

The preference category **Operation ▶ CAD Data Management ▶ Content Handling ▶ Download** lists the secondary content categories for your site. For each category of file type, if the preference `<secondary_content_category>` (for example, Instance Accelerator File) is set to yes, it means that any secondary content of the specified category type is downloaded automatically when the primary content is downloaded. Setting the value to false specifies that the secondary content is not downloaded automatically with the primary content of CAD document.

Managing Drawing Dependents

The large number of dependents that may be associated to drawings can affect performance if an unnecessarily large number of objects are gathered into the workspace during collection activities.

To control this behavior, the preference, Operation > Gathering (Core HTML Component) > **Trace Drawing Optional Dependents**, allows you to specify whether or not optional dependents for drawings should be traced (and collected into the workspace). The possible values for the preference and the resultant behavior are as followed:

- PerConfiguration – Whatever rule the user specifies for configuration (that is, Dependents: All, Required or None) is honored by the collection action.
- Required – If the user specifies All for dependents, only the required dependents are traced and collected (optional dependents are avoided). If the user specifies None, no drawings dependents are collected.

The default setting is PerConfiguration.

Note

This setting only applies to drawings that are included by collection; it does not apply to drawings that are "initially selected."

Controlling the Display of InternalCreo Parametric Relationships

In certain reports accessed through the CAD document information page, Windchill displays a column labeled Dependency Type or Reference Type. This column displays the type of link between the table item and the CAD document reported on, based on the internal relationship of their respective Creo Parametric models.

The Table Display mechanism allows users to filter out display of unwanted objects. In addition, your site may prefer to remove the display of certain internal Creo Parametric relationships in Pro/ENGINEER Wildfire pages. For example, when viewing the References of an assembly drawing you can set a site-wide property in <Windchill>/codebase/wt.properties to display only the associated assembly, the Drawing Model (default behavior is to display all dependency or reference types).

To restrict the view of assembly drawing references to the assembly itself, set com.ptc.windchill.cadx.caddoc.excludeDependencyTypes=<value> to -1. Additional dependencies can be removed from display by adding other, comma-separated values as described in the following table:

Value	Dependency Type	Description
-2	Internal Creo Parametric Instance	Internal Creo Parametric Instance

Value	Dependency Type	Description
-1	Default	Dependencies created by Creo Parametric that are not visible to the user through the reference viewer. (For example, displaying a dimension of a component of an assembly that is a model on a drawing.)
0	Default	Any dependency that does not fall into one of the more specifically defined categories
1	Declared_Layout	From layout to the model that declared it
2	Membership	From an assembly to an assembled model
4	Drawing_Model	From a drawing to any model that is its drawing model or a report
8	Relation_Reference	From an object to a related object
16	Drawing_Format	From drawing to its format
128	Merge_Part	
1024	Drawing_Overlay	From drawing to overlay drawing
8192	PDT_UDF	From a User Defined Feature (UDF) placed in a model to its source UDF Model (*gph for Creo Parametric; *sldfip for SolidWorks; *prt for NX)
262144	UDF_Model	
1048576	Markup_Drawing	
2097152	Interchange_Dependency	
4194304	Substitute_Dependency	
268435456	Concept_Model	

Clean-up of the Event Management Utility

To avoid possible performance issues resulting from an accumulation of a large number of event records in the Event Management utility, add the following site-wide property to wt.properties:

com.ptc.core.task.purgeTasksOlderThanDays=5

Events older than the specified number of days are automatically purged from the Event Management utility.

Administering Table Views

The display of information in many tables is user- and administrator-definable using the **Customize View List** window, available by selecting **Customize** from a table's **Current View** menu. Specific views for tables can be created, or existing views can be edited or saved as new views. Administrative users have the options of making a table view available to all users by selecting the **Share with all users** check box, and of showing the view in all **Current View** lists, by selecting the **Show in current view list** check box, on the first step, **Set Name**, of the **New View** or **Edit View** windows, available from the **Customize View List** window.

Configuring the Number of Workspace Rows Displayed

The property `com.ptc.windchill.uwgm.cadx.ws.sizeToWindow` controls the number of rows displayed in the workspace object list. Set to true (the default setting), the number of rows shown is based on the height of the window.

To display a fixed number of rows (for example, 10 for Windows), set the property to false, as follows:

```
<Property name="com.ptc.windchill.uwgm.cadx.ws.sizeToWindow"
overridable="true" targetFile="codebase/wt.properties" value="false"/>
```

Configuring Automatic Scrolling in the Workspace

The property `com.ptc.windchill.uwgm.cadx.ws.scrollToTable` controls whether or not the workspace page appears automatically scrolled to the beginning of the object list table or appears scrolled to the top of the page. Set to true (the default setting), the page automatically scrolls to show the workspace object list.

To disable automatic scrolling, set the property to false, as follows:

```
<Property name="com.ptc.windchill.uwgm.cadx.ws.scrollToTable"
overridable="tru" targetFile="codebase/wt.properties" value="false"/>
```

System Configuration Recommendations

The following sections describe recommendations for configuring your system to enhance your PDM operations.

Running Multiple Servers

It is recommended that Windchill PDMLink and Windchill ProjectLink be configured to run multiple method servers on servers with multiple CPUs and to run Oracle on a second server, especially when there is a single-CPU server running Windchill.

Using External File Vaulting

Content files persisted in external vaults are retrieved faster than content files stored in Oracle as binary large objects (BLOBS).

Although use of file vaults can add complexity to backup and recovery operations, vault management can be simplified by using the xconfmanager to set the wt.property wt.fv.forceContentToVault = true. This causes all content to vault to the DefaultCacheVault, keeping it out of Oracle BLOBs, without requiring creation of a vaulting rule.

In the event that multiple vaults must be implemented at your site, a vaulting rule applied to the User domain (where EPMDocuments are created) can direct content to vault appropriately.

Note

Following a custom checkin, the user is able to see CAD documents to be vaulted only in the default cache folder until an explicit revaulting action (executed through the replication schedule set by the administrator) is executed.

For more information on external vaulting and vaulting rules see the *Windchill Enterprise Administration Guide*.

Using Content Replication

Content replication provides the means to copy selected content files from a master server to remotely located replica servers for faster access by users at the remote site, thereby significantly improving access time. The files at the replica site remain retrievable by users at the master site. For more information, see the *Windchill Enterprise Administration Guide*.

Performance Tuning

The following sections describe recommendations for maximizing your system's PDM performance.

Setting the Method Server Max Heap Size

It is recommended that the default Java heap size for each method server be set to 512MB in order to cope with large Creo Parametric data sets that are common to the products developed by Creo Parametric users.

For more information on setting the max heap size, see the chapter, Method Server Maximum Heap Size, in the *Workgroup Manager Performance Best Practices Guide*.

Data Compression

The meta data compression option is intended to improve the upload and download performance of the Creo Parametric client for users accessing Windchill across a lower bandwidth network. This feature substantially improves the performance of upload and download operations for large family tables.

Creo Parametric Settings

In Creo Parametric, compression is controlled by a Creo Parametric config.pro setting (`dm_http_compression_level`) as follows:

`dm_http_compression_level` <an integer between 0 and 9 – 0 for no compression, 9 for maximum compression>

Windchill Settings

On the Windchill side, you enable the compression filters provided by the web-servers (for example, `mod_gzip` for Apache1.3.x and `mod_deflate` for Apache2.0.x).

Additional SOAP Compression Filter

Additionally, out of the box, the Windchill `SOAPCompressionFilter` is configured for compressing HTTP response data for special client (for example, Creo Parametric) interactions, such as downloading the contents of a model.

Following additional property settings that control data compression behavior are applicable only to the `SOAPCompressionFilter`.

To use these property settings, add them to the `wt.properties` file:

- `wt.compression.threshold=<size_in_bytes>` – Sets a threshold for which HTTP responses are to be compressed. The default value (0) specifies that all responses are compressed.
- `wt.compression.off.contentEncodings=<encoding_types>` –Identifies HTTP response encoding types (case insensitive) for which compression is switched off. The default encoding types are (space delimited): `identity gzip deflate lws-`

deflate. Setting the value to asterisk (*) switches off compression for all encoding types.

- `wt.compression.off.contentTypes=<content_type>` –Identifies HTTP response content types (case insensitive) for which compression is switched off. The default content types are (space delimited): `image/jpeg image/gif application/zip`.

Tip

While data compression can provide a benefit in a slow network, using compression puts an extra load on CPU resources; therefore, if network speed is not an issue, the use of compression may decrease performance and is not recommended.

Additional Considerations

If the Windchill compression filter is configured and `dm_http_compression_level` preference is set in the Creo Parametric config file (`config.pro`), this setting will also apply to any interaction between the Creo Parametric embedded browser and the server. That is, a non-zero value of the preference will ensure that not only the meta-data of Creo Parametric models but even the content/UI pages will be sent in the compressed form reducing the overall network traffic.

Also note that the Creo Parametric configuration option `dm_http_compression_level` needs to be set before registering the server through Creo Parametric or before connecting to a registered server (if already registered). Any change in the value after the server is registered or connected, will not apply to the running Creo Parametric session.

Maximizing the Oracle Server/Windchill Method Server Connection

Due to the large number of objects and CAD documents involved in database transactions, it is highly recommended that the connection between the Oracle server and the Windchill method server machines is both low-latency and high-bandwidth.

Note

Bulk HTTP data transfer using Apache on Windows 2000 can be restricted by Apache's default send buffer size. We found that setting property `SendBufferSize=16384` in `httpd.conf` significantly improved throughput over high latency, low bandwidth WANs.

Choosing to Display Family Object Symbols in Folders Table

Showing or hiding the family table symbols on CAD document type icons in the **Folders** table can be controlled by the following property:

`wt.clients.showFamilyGlyph`

Because there is a significant performance benefit in bypassing the queries that determine whether or not to show the family table symbols, the property defaults to false OOTB. Therefore, family table symbols on CAD document type icons do not appear by default in the **Folders** table.

To show the family table symbols in the **Folders** table, set `wt.clients.showFamilyGlyph` to true using the `xconfmanager`.

Other Recommendations

The following sections provide additional information about working with Windchill.

Controlling End User Objects

While workspaces are private areas "owned" by their creators, they may sometimes need to be accessed by an administrator. Perhaps the most typical need is to release the check-out lock on an object in a workspace whose owner is unavailable or has left the company.

An administrator with appropriate access privileges can selectively release the objects in one of two ways:

1. Using a standalone browser or an embedded browser, an administrator can locate the part either through Windchill search or by browsing Windchill folders. From the search results or the folder page, the administrator selects the object and performs **Undo Check Out** on the selected object and its dependents using the **Actions** pop-up.
2. Using a standalone browser or an embedded browser, an administrator can navigate to the information page of an object and perform **Undo Check Out** on the object and its dependents using the **Actions** drop-down menu.

Note

*Only a single initially-selected object and its dependents can be processed with administrative **Undo Check Out**.*

In addition, Windchill provides administrators with appropriate access privileges to locate other users' workspaces and delete them. This results in the undoing of any checkouts and removal from the workspace for any objects in the workspace.

To delete a workspace, perform the following procedure:

1. In the context for which you have administrative privileges, select the **Workspaces** minor tab.

The **My Workspaces** page appears.

2. In the lower table, titled Other user's workspaces, enter a user name in the **Enter User Name** field and click **Go**

or

If you are unsure of the user name, you can click **Find** to access the **Users** window where you can search for a user by any of the following criteria:

- **Full Name**
- **User Name**
- **Email**
- **Organization Name**

In the search results, select a user and click **Go**.

3. After you click **Go**, the **Other user's workspaces** table refreshes to display the workspaces owned by that user.
4. Select a workspace and click **Delete**.

The workspace is deleted from the system. Any objects in the workspace are removed from the workspace and any checkouts for those objects are undone.

Online Java Performance Guide

You may want to review the online Java Performance Guide to identify server-side Java settings that can boost performance.

Note

Be sure to carefully evaluate the options prior to implementation. PTC does not currently support them.

For more information on the online Java Performance Guide, see the chapter, *Online Java Performance Guide*, in the *Workgroup Manager Performance Best Practices Guide*

Windchill Folder Structure

It is important to carefully plan the Windchill cabinet/folder structure, and direct Windchill users to keep the number of objects (particularly, the CAD documents) in each Windchill folder to a manageable number (for example, up to a few

hundred CAD documents). If the number is too large, it is difficult for other users to find an object in a folder and wait time is increased during browsing (as the information about each folder is extracted and communicated to the client).

HTTP Protocol

Creo Parametric only communicates with the server through HTTP requests. All HTTP requests (either to get an HTML page from the Windchill server, upload models, or perform a database operation through a SOAP request) are being made through the embedded browser. Therefore, all of the settings that are in effect for the embedded browser (including authentication, HTTP proxy server setting, etc.) apply to the Creo Parametric interaction with the server. If the Windchill server is using secure HTTP (HTTPS), then Creo Parametric also uses HTTPS.

Note

General usage of Creo Parametric (for example, managing CAD data through check-in or check-out) does not involve any applet, and therefore RMI is not used. However, if Creo Parametric is used as a Web browser to access pages containing applets, then RMI should be taken into consideration when configuring the firewall.

Windchill Preferences That Control Interaction with Creo Parametric

The following sections present tables listing important preferences that control aspects of Windchill that are especially of interest to Creo Parametric users and that may not have already been described in the preceding sections of this chapter. The tables are organized by preference category.

Note

Collection preferences for the various action categories are summarized in the section [Configuring the Initial Collection of Objects for Actions on page 130](#).

Create and Edit

Key preferences in the Create and Edit category are described in the following table:

Preferences	Values	Description
Allow checkout of non-latest iterations	Allow checkout of non-latest iterations	There are three options possible. Do not allow the checkout of non-latest iterations. Allow the checkout of non-latest
	Do not allow checkout of non-latest iterations	

Preferences	Values	Description
	Throw an overrideable conflict if a user tries to checkout a non-latest iteration	iterations for valid object types like CAD documents and their dependents. Present an overrideable conflict when attempting to checkout a non-latest iteration.

Display

Key preferences in the Display category are described in the following table:

Preference	Values	Description
Apply Configuration specification to the originally selected object	Yes No (default)	Allows you to specify if the selected configuration specification will be applied or not to the originally selected items
Enable Dependency Processing Type	Yes No (default)	Controls the display of the Dependency Processing Type selector in the Edit Filter page when launched from the collector.
Incomplete object resolution	Ignore optional dependencies Ignore optional reference dependencies Ignore internal dependencies only Do not allow to ignore	Determines whether certain types of incomplete object dependencies can be ignored during a checkin or not
Toolbar Action Descriptions	Yes No (default)	Controls the display of the action description under the icon in the toolbar area of tables and trees
Workspace	Yes No (default)	Determines whether to enable the use of the workspace. If set to no, the workspace navigation link is not displayed.

EPM Service Preferences

This category has the sub-category: Build Service Preferences. In addition, the preference, Send a CAD document / Dynamic Document to PDM without optional dependents, is listed separately and is described in the following table:

Preference	Values	Description
Send a CAD document / Dynamic Document to PDM without optional dependents	Yes No (default)	Determines whether to allow a CAD document or dynamic document to be sent to PDM without new optional dependents. If allowed, links to optional dependents are removed.

Build Service Preferences

The following table lists preferences for the build service:

Preference	Values	Description
Attributes Delimiter	, (default) <character value>	Identifies the delimiter used in listing attributes to be published
Attributes to be published on Link	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the member link
Attributes to be published on Master	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the master
Attributes to be published on Occurrence	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on an occurrence
Attributes to be published on Part	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Identifies attributes to be published on the part
Contributing Content Attributes	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Attributes to be published on a part by the Contributing Content relationship. Attributes are delimited by a character specified in

Preference	Values	Description
		the preference, Attributes Delimiter
Contributing Image Attributes	<String(s), separated by delimiter character set in preference, Attributes Delimiter>	Attributes to be published on a part by Contributing Image relationship. Attributes are delimited by a character specified in preference Attributes Delimiter

Operations Preferences

The operations category contains subcategories corresponding to many common PDM actions. The following subcategories are listed

Auto Associate Preferences

The following table describes the preferences for the **Auto Associate** action:

Preference	Values	Description
Allow Association of Model Items by Model Item Sub-types	CUSTOM, LIBRARY (<comma-separated list>)	Lists Model Item sub-types for which associations are allowed. These are entered as comma-separated values.
Allow Association of Model Items by Model Item Types	COMPONENT (<comma-separated list>)	Lists Model Item types for which associations are allowed. These are entered as comma-separated values.
Auto Associate Naming Parameter	<string>	Identifies the CAD file parameter used when naming a new part during Auto Associate . The default is <no value> (no CAD parameter is used to name the part).
Auto Associate Numbering Parameter	<string>	Identifies the CAD file parameter used when numbering a new part during Auto Associate . The default is <no value> (no CAD parameter is

Preference	Values	Description
		used to number the part).
Auto Associate Truncate Name File Extension	Yes No (default)	Truncates the file extension (from CAD filename) in the part name when the part is created. When set to "Yes," truncates the file extension (from CAD filename) in the part name when the part is created. Default is "No" (file extension is kept in part name).
Auto Associate Truncate Number File Extension	Yes No (default)	Truncates the file extension (from CAD filename) in the part name when the part is created. When set to "Yes" truncates the file extension (from CAD filename) in the part name when the part is created. Default is "No" (file extension is kept in part name).
Create Alternate Link On Check In	Yes No (default)	Allows a link (CAD document to part association) of the next available type to be created if the part already has an Owner link. When set to "Yes," the next available valid link is created if a matching part is found, and the checkin continues. The default is "No" (Check In fails with an overridable conflict).
Create Associate New Part	Owner Only (Default) Owner and Contributing Image All	Specifies whether a new part should be created if a matching part is not found by Auto Associate. The default is "Owner Only" for all CAD tools. ECAD

Preference	Values	Description
	Never	<p>authoring applications default to "All".</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • Owner Only: If a matching part is not found, a new part is created when the CAD document would associate to a part with an "Owner" association. • Owner and Contributing Image: If a matching part is not found, a new part is created when the CAD document would associate to a part with either an "Owner" or "Contributing Image" association. • All: If a matching part is not found, a new part is created when the CAD document would associate to a part with any product structure association ("Owner", "Contributing Image", and "Image"). • Never: A new part is not created if an existing part is not found, even if it contributes to product structure. Auto associate does not fail, the CAD document is skipped, and other

Preference	Values	Description
		selected CAD documents will try to associate.
Create Content Links for Drawings	Yes No (default)	Windchill is capable of locating all drawings referenced by a drawing model and automatically relating them to the part of the model. In some cases, you may want to distinguish manufacturing drawings versus conceptual drawings by using a link in the database, rather than using the calculated relationship. This preference controls whether the auto associate action creates a link in the database. The default is "No".
Custom Class for Auto Associate Part	<string> com.ptc.windchill.uwgm.common.autoassociate.DefaultAutoAssociatePartFinderCreator. (default)	Specifies the name of the class that implements AutoAssociatePartFinder Creator interface. The default is the standard Windchill implementation, com.ptc.windchill.uwgm.common.autoassociate.DefaultAutoAssociatePartFinderCreator.
Disallow Product Structure Links for CAD Document Sub-types	CADASSEMBLY CADCOMPONENT CADDRAWING FORMAT LAYOUT MANUFACTURING MARKUP	Lists CAD document sub-types which cannot be actively associated (form owner links). There are drop-down menus to allow assigning values per authoring application. The default values for Creo Parametric are SKEL_MODEL, EXTERNALSIMPREP.

Preference	Values	Description
	OTHER REPORT WELDMENT	These are comma-separated string values. Note <i>The value OTHER is used for neutral file formats and Creo Parametric text files used as libraries (for example, .mat)</i>
Disallow Product Structure Links for CAD Document Types	CADDRAWING DIAGRAM FORMAT LAYOUT MANUFACTURING MARKUP OTHER REPORT SKETCH UDF CUTTER_LOCATION MACHINE_CONTROL MECHANICARESULTS MECHANICAREPORT	Lists CAD document types which cannot be actively associated, (form owner links). There are drop-down menus to allow assigning values per authoring application. The default top-level values are CADDRAWING, CALCULATION_DATA. The default values for Creo Parametric are shown in the Values column. These are comma-separated values. Note <i>The value OTHER is used for neutral file formats and Creo Parametric text files used as libraries (for example, .mat)</i>
Disallow Structure CAD Document Sub-types	SKEL_MODEL EXTERNALSIMPREP ,	Lists CAD document sub-types which cannot be actively associated (form owner links). There are drop-down menus to allow assigning values per authoring application. The default values for Creo Parametric are SKEL_MODEL, EXTERNALSIMPREP, Sheetmetal. These are comma-separated string values.

Preference	Values	Description
		<p>Note</p> <p>The value <i>OTHER</i> is used for neutral file formats and Creo Parametric text files used as libraries (for example, .mat)</p>
Disallow Structure CAD Document Types	CADDRAWING DIAGRAM FORMAT LAYOUT MANUFACTURING MARKUP OTHER REPORT SKETCH UDF MECHANICARESULTS MECHANICAREPORT	<p>Lists CAD document types which cannot drive product structure. There are drop-down menus to allow assigning values per authoring application. The default top-level values are CADDRAWING.</p> <p>The default values for Creo Parametric are listed in the Values column. These are comma-separated string values.</p> <p>Note</p> <p>The value <i>OTHER</i> is used for neutral file formats and Creo Parametric text files used as libraries (for example, .mat)</p>
Disallow Structure Model Item Sub-types	LIBRARY <comma-separated list>	<p>Lists Model Item sub-types which cannot create an Owner association. These Model Items will create Image associations during Auto Associate. Values for generic ECAD authoring application are LIBRARY, ECAD_BOARD, ECAD_SCHEMATIC, ECAD_COMPONENT</p>
Disallow Structure Model Item Types	<comma-separated list>	<p>Lists Model Item types which cannot create "Owner" association. These Model Items will create "Image"</p>

Preference	Values	Description
		associations during Auto Associate.
Force Autonumbered Part Creation	Yes No (default)	Allows Auto Associate to create an autonumbered part upon check in if the part number specified via CAD document numbering parameter is not found in Windchill and autonumbering is the policy for new parts in Windchill. When set to "No," Check In fails with an overridable conflict to create an autonumbered part. When the value is "Yes," Check In creates an autonumbered part.
Part Master Class for Search	<string> wt.part.WTPartMaster (default)	Identifies the internal name of the part master type searched for during Auto Associate . Search is allowed for a customer defined part or part subclass. This preference can enable searching for customized parts by specifying the fully qualified class name of the master of the customized part, so that the search is restricted to the customized part class (and the whole WTPart class is not searched). The value wt.part.WTPartMaster indicates that the search for part is done on WTPart class. The default is the standard Windchill implementation, wt.part.WTPartMaster.

Preference	Values	Description
Part Structure Override Attribute Name	<string>	Identifies the name of a boolean global attribute that determines if the CAD document participates in build. This global attribute can be on: iteration, master, member link, model item, or model item link. This global attribute also determines whether to auto associate the model to a part.
Phantom Assembly Override Attribute Name		Identifies the name of a boolean global attribute that determines if the CAD document is a phantom assembly. This global attribute can be on: iteration and master. This global attribute is used to mark the part created in auto associate as phantom (hidden from BOM).
Set Revision For Part	Yes No (default)	Sets the revision of a part to match the revision of the CAD document during Auto Associate . When set to "Yes," Auto Associate sets the revision of a part to match the revision of the CAD document. This preference is for business practices that want to keep these revisions in sync.
Store New Parts with CAD Documents	Yes No (default)	Specifies the location of newly created parts to be the same as the location of their associated CAD document. When set to "Yes," newly created parts have the same location as

Preference	Values	Description
		their associated CAD document.

CAD Data Management Preferences

The following table describes the preferences in the CAD Data Management category > Content Handling > Download sub-category:

Preference	Values	Description
Analysis Input	Yes (default) No	Sets the download preference for the Analysis Input content category. When set to "Yes," specifies that the secondary content category Analysis Input is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Analysis Results	Yes No (default)	Sets the download preference for the Analysis Results content category. When set to "Yes," specifies that the secondary content category Analysis Results is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Creo Parametric UGC	Yes (default) No	Sets the download preference for the Creo Parametric UGC content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC is to be downloaded when primary content is downloaded. When set to

Preference	Values	Description
		"No," no secondary content of this content category is downloaded.
Creo Parametric UGC Section	Yes (default) No	Sets the download preference for the Creo Parametric UGC Section content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC Section is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Creo Parametric UGC Section Table of Contents	Yes (default) No	Sets the download preference for the Creo Parametric UGC Section Table of Contents content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC Section Table of Contents is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Design Data	Yes No (default)	Sets the download preference for the Design Data content category. When set to true, specifies that the secondary content category Design Data is to be downloaded when primary content is downloaded. The default is false.
Drawing	Yes (default)	Sets the download

Preference	Values	Description
	No	preference for the Drawing content category. When set to "Yes," specifies that the secondary content category Drawing is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Export	Yes No (default)	Sets the download preference for the Export content category. When set to "Yes," specifies that the secondary content category Export is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Family Table	Yes (default) No	Sets the download preference for the Family Table content category. When set to "Yes," specifies that the secondary content category Family Table is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
General	Yes No (default)	Sets the download preference for the General content category. When set to "Yes," specifies that the secondary content category General is to be downloaded when

Preference	Values	Description
		primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
IDEAS Drawing Sheet	Yes (default) No	Sets the download preference for the IDEAS Drawing Sheet content category. When set to "Yes," specifies that the secondary content category IDEAS Drawing Sheet is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
IDEAS Legacy Drawing	Yes No (default)	Sets the download preference for the IDEAS Legacy Drawing content category. When set to "Yes," specifies that the secondary content category IDEAS Legacy Drawing is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
IDEAS Package		Sets the download preference for the IDEAS Package content category. When set to "Yes," specifies that the secondary content category IDEAS Package is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content

Preference	Values	Description
		category is downloaded.
Image	Yes (default) No	Sets the download preference for the Image content category. When set to "Yes," specifies that the secondary content category Image is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Import	Yes (default) No	Sets the download preference for the Import content category. When set to "Yes," specifies that the secondary content category Import is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Information	Yes No (default)	Sets the download preference for the Information content category. When set to "Yes," specifies that the secondary content category Information is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Instance Accelerator File	Yes (default) No	Sets the download preference for the Instance Accelerator File content category. When set to "Yes," specifies that the secondary content

Preference	Values	Description
		category Instance Accelerator File is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Inventor Design View Document	Yes (default) No	Sets the download preference for the Inventor Design View Document content category. When set to "Yes," specifies that the secondary content category Inventor Design View Document is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Inventor iPart Instance	Yes (default) No	Sets the download preference for the Inventor iPart Instance content category. When set to "Yes," specifies that the secondary content category Inventor iPart Instance is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Inventor Model	Yes (default) No	Sets the download preference for the Inventor Model content category. When set to "Yes," specifies that the secondary content category Inventor Model

Preference	Values	Description
		is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Logical Reference	Yes (default) No	Sets the download preference for the Logical Reference content category. When set to "Yes," specifies that the secondary content category Logical Reference is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Manufacturing	Yes (default) No	Sets the download preference for the Manufacturing content category. When set to "Yes," specifies that the secondary content category Manufacturing is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Mesh	Yes No (default)	Sets the download preference for the Mesh content category. When set to "Yes," specifies that the secondary content category Mesh is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content

Preference	Values	Description
		category is downloaded.
Package	Yes (default) No	Sets the download preference for the Package content category. When set to "Yes," specifies that the secondary content category Package is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Parameter Table	Yes (default) No	Sets the download preference for the Parameter Table content category. When set to "Yes," specifies that the secondary content category Parameter Table is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Text	Yes No (default)	Sets the download preference for the Text content category. When set to "Yes," specifies that the secondary content category Text is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Toolpath	Yes No (default)	Sets the download preference for the Toolpath content category. When set to "Yes," specifies that the

Preference	Values	Description
		secondary content category Toolpath is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.
Viewable	Yes No (default)	Sets the download preference for the Viewable content category. When set to "Yes," specifies that the secondary content category Viewable is to be downloaded when primary content is downloaded. When set to "No," no secondary content of this content category is downloaded.

The following table describes the preferences in the CAD Data Management category > Content Handling > Mark Out Of Date sub-category:

Preference	Values	Description
Analysis Input	Yes No (default)	Sets the mark out-of-date preference for the Analysis Input content category. When set to "Yes," specifies that the secondary content category Analysis Input is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Analysis Results	Yes No (default)	Sets the mark out-of-date preference for the Analysis Results content category. When set to "Yes," specifies that the secondary content category

Preference	Values	Description
		Analysis Results is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Creo Parametric UGC	Yes (default) No	Sets the mark out-of-date preference for the Creo Parametric UGC content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Creo Parametric UGC Section	Yes (default) No	Sets the mark out-of-date preference for the Creo Parametric UGC Section content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC Section is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Creo Parametric UGC Section Table of Contents	Yes (default) No	Sets the mark out-of-date preference for the Creo Parametric UGC Section Table of Contents content category. When set to "Yes," specifies that the secondary content category Creo Parametric UGC Section Table of Contents is marked out of date when primary

Preference	Values	Description
		content is iterated. When set to "No," secondary content is carried forward.
Design Data	Yes No (default)	Sets the mark out-of-date preference for the Design Data content category. When set to true, specifies that the secondary content category Design Data is to be downloaded when primary content is downloaded. The default is false.
Drawing	Yes No (default)	Sets the mark out-of-date preference for the Drawing content category. When set to Yes," specifies that the secondary content category Drawing is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Export	Yes No (default)	Sets the mark out-of-date preference for the Export content category. When set to "Yes," specifies that the secondary content category Export is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Family Table	Yes No (default)	Sets the mark out-of-date preference for the Family Table content category. When set to "Yes," specifies that the secondary content category Family Table is marked out of

Preference	Values	Description
		date when primary content is iterated. When set to "No," secondary content is carried forward.
General	Yes No (default)	Sets the mark out-of-date preference for the General content category. When set to "Yes," specifies that the secondary content category General is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
IDEAS Drawing Sheet	Yes No (default)	Sets the mark out-of-date preference for the IDEAS Drawing Sheet content category. When set to "Yes," specifies that the secondary content category IDEAS Drawing Sheet is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
IDEAS Legacy Drawing	Yes No (default)	Sets the mark out-of-date preference for the IDEAS Legacy Drawing content category. When set to "Yes," specifies that the secondary content category IDEAS Legacy Drawing is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
IDEAS Package	Yes No (default)	Sets the mark out-of-date preference for the IDEAS Package content category.

Preference	Values	Description
		When set to "Yes," specifies that the secondary content category IDEAS Package is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Image	Yes No (default)	Sets the mark out-of-date preference for the Image content category. When set to "Yes," specifies that the secondary content category Image is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Import	Yes No (default)	Sets the mark out-of-date preference for the Import content category. When set to "Yes," specifies that the secondary content category Import is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Information	Yes No (default)	Sets the mark out-of-date preference for the Information content category. When set to "Yes," specifies that the secondary content category Information is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Instance Accelerator File	Yes No (default)	Sets the mark out-of-date preference for the Instance

Preference	Values	Description
		Accelerator File content category. When set to "Yes," specifies that the secondary content category Instance Accelerator File is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Inventor Design View Document	Yes No (default)	Sets the mark out-of-date preference for the Inventor Design View Document content category. When set to "Yes," specifies that the secondary content category Inventor Design View Document is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Inventor iPart Instance	Yes No (default)	Sets the mark out-of-date preference for the Inventor iPart Instance content category. When set to "Yes," specifies that the secondary content category Inventor iPart Instance is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Inventor Model	Yes (default) No	Sets the mark out-of-date preference for the Inventor Model content category. When set to "Yes," specifies that the secondary content category

Preference	Values	Description
		Inventor Model is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Logical Reference	Yes No (default)	Sets the mark out-of-date preference for the Logical Reference content category. When set to "Yes," specifies that the secondary content category Logical Reference is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Manufacturing	Yes No (default)	Sets the mark out-of-date preference for the Manufacturing content category. When set to "Yes," specifies that the secondary content category Manufacturing is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Mesh	Yes No (default)	Sets the mark out-of-date preference for the Mesh content category. When set to "Yes," specifies that the secondary content category Mesh is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Package	Yes (default) No	Sets the mark out-of-date preference for the Package content category. When

Preference	Values	Description
		set to "Yes," specifies that the secondary content category Package is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Parameter Table	Yes No (default)	Sets the mark out-of-date preference for the Parameter Table content category. When set to "Yes," specifies that the secondary content category Parameter Table is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Text	Yes No (default)	Sets the mark out-of-date preference for the Text content category. When set to "Yes," specifies that the secondary content category Text is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Toolpath	Yes No (default)	Sets the mark out-of-date preference for the Toolpath content category. When set to "Yes," specifies that the secondary content category Toolpath is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.
Viewable	Yes (default) No	Sets the mark out-of-date preference for the

Preference	Values	Description
		Viewable content category. When set to "Yes," specifies that the secondary content category Viewable is marked out of date when primary content is iterated. When set to "No," secondary content is carried forward.

Check In Operation Preferences

The following table describes the preferences for the Check In action, exclusive of the Check In Operation > Collection preferences which are listed in a subsequent table:

Preference	Values	Description
Auto Associate upon Check In	Yes No (default)	Controls the default behavior whether or not to perform Auto Associate for objects that do not have associated parts. If set to "Yes," associated parts are created.
Conflict for Out of date Secondary content upon Check In	Yes (default) No	Controls the default behavior whether or not to provide overrideable conflict to the user when secondary content is considered out of date. If set to "Yes," an overrideable conflict is provided.
Create As Stored	Yes (default) No	If set to "Yes," specifies to create an As Stored configuration upon Check In.
Create Baseline upon Check In	Yes No (default)	Controls the default behavior whether or not to create baseline upon Check In. If set to "yes," a baseline is created.
ModelCHECK Configuration	Default: check/ default_checks .mch,start/	Specifies ModelCHECK Configuration files to be

Preference	Values	Description
	nostart.mcs, constant/ inch.mcn Basic: check/ default_checks.mch, start/ nostart.mcs, constant/ inch.mcn Release: check/ default_checks.mch, start/ nostart.mcs, constant/ inch.mcn Approval: check/ default_checks .mch,start/ nostart.mcs, constant/ inch.mcn Review: check/ default_checks .mch,start/ nostart.mcs, constant/ inch.mcn	used for validation for each LifeCycle states in a specific syntax (eg. :, , , ,). The configuration specified by "Default" LifeCycle State is default behavior. Value: Default:check/default_checks .mch,start/nostart.mcs, constant/inch.mcn Basic:check/default_checks. mch,start/nostart.mcs, constant/inch.mcn Release:check/default_checks. mch,start/nostart.mcs ,constant/inch.mcn Approval:check/default_checks.mch,start/nostart.mcs,constant/inch.mcn Review:check/default_checks.mch,start/nostart.mcs,constant/ inch.mcn
ModelCHECK Mode	Disabled Interactive (Default) Regenerate Implicit Regenerate Always Save Batch	Specifies ModelCHECK mode allowed for model to be validated. Default is Interactive.
ModelCHECK Number of Errors	<integer> (default = 0)	Specifies the maximum number of ModelCHECK errors allowed
ModelCHECK Number of Hours	<integer> (default = 24)	Specifies the maximum allowable hours between a ModelCHECK verification at the client and the actual model checkin to Windchill. The default is 24.
ModelCHECK Validation	Yes No (default)	Specifies whether ModelCHECK validation is performed at Check In.

Preference	Values	Description
		If set to "Yes," validation is performed.
Remove objects from Workspace after Check In	Yes No (default)	Controls the default behavior whether or not to remove all objects from Workspace after Check in is completed. All objects include objects being checked in and not being checked in.
Resolve Incomplete Objects	Yes No (default)	Controls the default behavior whether or not to resolve incomplete objects automatically upon Check In.
Undo Check Out objects after Check In	Yes No (default)	Controls the default behavior whether or not to perform Undo Check Out after Check in is completed on objects that have been checked out but not modified.
Update Incomplete Objects on Server	Yes (default) No	Controls the default behavior whether or not to update incomplete objects on server upon resolving incomplete objects upon Check In. This option has no effect unless "Resolve Incomplete Objects" option is turned on through preferences or Check In page.

Note

The Check In category also has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Collection-related Preferences for Actions

The following table lists preferences to specify the default collection behavior for the object types listed. A sub-category, Collection, is available for many actions that are included in the Operation category, These actions include Check Out, Edit Attributes, Relationship Report, Remove from Workspace, Rename, Set State, Undo Check Out, Update, and Upload.

Preferences	Values	Description
Display collected objects	As a List As a Structure As a Structure with Associated Objects	Allow specification of the way collected objects are listed in the table. Each display allows viewing how objects have been collected by displaying the different association in a specific way. Default is "As a list".
Include dependent CAD / Dynamic documents	All Required (default) None	Specifies which dependent CAD / Dynamic Documents for the collected CAD /Dynamic Documents will be by default added to the collection
Include dependent Parts	All None (default)	Specifies which dependent parts for the collected parts will be by default added to the collection.
Include related CAD Documents	All Initially Selected Only None (default)	Specifies which CAD Documents associated to the collected parts will be by default added to the collection.
Include related Drawings	All Initially Selected Only None (default)	Specifies which drawings associated to the collected CAD documents or parts will be by default added to the collection.
Include related Image objects	All Initially Selected Only None (default)	Specifies which image objects related to the collected source documents will be by default added to the collection.

Preferences	Values	Description
Include related Parts	All Initially Selected Only None (default)	Specifies which parts associated to the collected documents, CAD documents, or dynamic documents will be by default added to the collection.
Include related Source objects	All Initially Selected Only None (default)	Specifies which source objects related to the collected representation documents will be by default added to the collection.

Create New Workspace Preferences

The following table describes the preferences applicable to creating a new workspace:

Preference	Values	Description
Additional valid characters on US English locale	<character value> (default = .-_)	Specifies additional valid characters in workspace name on US English locale. All alphanumeric characters are valid by default.
Part centric dependency processing mode	Yes No (default)	If set to "Yes," specifies part centric dependency processing mode.

Edit Attributes Preferences

The following table lists preferences that control the default collection of related objects during Edit Attributes:

Preferences	Values	Description
Display collected objects	As a List As a Structure As a Structure with Associated Objects	Allow specification of the way collected objects are listed in the table. Each display allows viewing how objects have been collected by displaying the different association in a specific way. Default is "As a list".

Preferences	Values	Description
Include dependent CAD / Dynamic documents	All Required (default) None	Specifies which dependent CAD / Dynamic Documents for the collected CAD /Dynamic Documents will be by default added to the collection
Include dependent Parts	All None (default)	Specifies which dependent parts for the collected parts will be by default added to the collection.
Include related CAD Documents	All Initially Selected Only None (default)	Specifies which CAD Documents associated to the collected parts will be by default added to the collection.
Include related Drawings	All Initially Selected Only None (default)	Specifies which drawings associated to the collected CAD documents or parts will be by default added to the collection.
Include related Family table objects	All Initially Selected Only None (default)	Specifies which family table objects related to the collected generic or instances will be by default added to the collection.
Include related Generics	All Initially Selected Only None (default)	Specifies which generics associated to the collected instances will be by default added to the collection.
Include related Image objects	All Initially Selected Only None (default)	Specifies which image objects related to the collected source documents will be by default added to the collection.
Include related Parts	All Initially Selected Only None (default)	Specifies which parts associated to the collected documents, CAD documents, or dynamic documents will be by

Preferences	Values	Description
		default added to the collection.
Include related Source objects	All Initially Selected Only None (default)	Specifies which source objects related to the collected representation documents will be by default added to the collection.

Gathering (Core HTML Component) Preferences

The following table describes the preference applicable to the Gathering sub-category:

Preference	Values	Description
Trace Drawing Optional Dependents	Per Configuration (default) Required	<p>Specifies how to limit collecting drawing dependents for included CAD drawings. If set to "Per Configuration," the system applies the dependents option (All, Required, or None) that the user set to the included drawings.</p> <p>If set to "Required," the system collects only the required dependents of the included drawing even if the configuration is set to All. If the configuration is set to Required, the system collects the required dependents. If the configuration is set to None, the system collects none.</p> <p>This preference only applies to included drawings, not to drawings that are initially selected prior to initiating the action.</p>

General Preferences

The following table describes the preferences applicable to the General sub-category:

Preference	Values	Description
Custom Modeled Part Class	<string> wt.part.WTPart (default)	Specifies a fully qualified class name for a custom modeled part if there are any. The default is wt.part.WTPart. To specify a different custom modeled part, enter the class name.
Map File Extension to "Publication Source" Document Type	<string> (default = xml,sgml,html,txt)	Lists file extensions for dynamic documents that should be mapped to the document type "Publication Source." This is a comma-separated list. The default is xml,sgml,html,txt.
Threshold for enabling parallel processing in "Request Result Collection"	<integer> (default is -1)	Specifies the threshold number of impacted objects that will activate parallel processing for result collection. By default, parallel processing is disabled.

New CAD Document Preferences

The following table describes the preferences applicable to the New CAD document sub-category:

Preference	Values	Description
Is Model Name Unique	Yes	Uniqueness constraint automatically set to true by Creo Parametric. This value should not be changed.
Synchronize CAD Model Name with CAD Doc Number	Yes No (default)	If set to "Yes," synchronizes the CAD model name with the CAD document number in the New CAD Document user interface.

Preference	Values	Description
		This synchronization is not applicable when auto-numbering is in effect.

Preferences for CAD Document Property (Information) Page

The following table describes the preferences applicable to the CAD document information page category:

Preference	Values	Description		
Display collected objects	As a list As a Structure As a Structure with Associated Objects	Allow users to change the way collected objects are listed in the table. Each display allows viewing how objects have been collected by displaying the different association in a specific way. Default is "As a list".		
Exclude Dependency Types for References Report	Value	Dependency Type	The dependency types to be excluded are defined as comma separated integers here. This information will be used in the References report. Additional dependencies can be removed from display by adding other, comma separated values as described in the following table	
	-2	Internal Creo Parametric Instance		Dependency linking object and hidden instance
	-1	Internal Creo Parametric		Dependencies created by Creo Parametric that are not visible to the user through the reference viewer. (For example, displaying a dimension of a component of an assembly that is a model on a drawing)

Preference	Values		Description
	0	Internal Creo Parametric	Any dependencies
	1	Layout Declared	Dependency from the layout to a model that has declared it
	4	Drawing Model or Report	Model defined as a model of the drawing
	8	Relation Reference	Relationship created between two objects in an assembly
	16	Drawing Format	Format on a drawing
	32	Generic Model	Family Table generic (legacy reference)
	64	Manufacturing Assembly	The assembly used for the .mfg file
	128	Merge Part	Dependency from the case where a feature on a part is created due to a merge in an assembly
	256	User Defined	User-defined dependency created in Pro/PDM

Relationship Report Preferences

Note

The Relationship Report category only has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Remove from Workspace Preferences

Note

The Remove from Workspace category only has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Rename Preferences

Note

The Rename category only has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Set State Preferences

Note

The Set State category only has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Undo Check Out Preferences

Note

The Undo Check Ou category only has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Update Operation Preferences

The following table describes the preferences applicable to the Update action (Update Operation > Update):

Preference	Values	Description
Add Primary Contents to Workspace	DOWNLOAD (default) LINK	Controls the default behavior whether content should be added to workspace as link or the file should be available. The default is DOWNLOAD (primary content is downloaded). When set to LINK, primary content is

Preference	Values	Description
		not downloaded, but a link to the content is created for later download as required.

Note

The Update category also has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Upload Preferences

The following table lists preferences for the Upload action:

Preference	Values	Description
Ignore Legacy Parameters	<comma-separated list>	Specifies a list of legacy parameters to ignore during upload operation if corresponding attribute is not present in the type definition on server. This is a comma-separated list and is sensitive to spaces after the comma.
Initial Revision Parameter	<string>	Identifies the file property name that shows the initial revision to be used when uploading a file to Windchill. This revision will be set on first upload and will be further controlled by Windchill.
Naming Parameter	<string>	Specifies the CAD tool parameter to be used when generating a CAD document name upon initial upload
Numbering Parameter	<string>	Specifies the CAD tool parameter to be used when generating a CAD document number upon initial upload
Subtype Parameter	<string>	Identifies the file property name that shows the subtype to be used when

Preference	Values	Description
		uploading a file to Windchill. The default value is UPLOAD_SOFT_TYPE. This property is used to assign a specific subtype to a file or a template. Subtype will be set on first upload and will not change after that.
Upload Drop Name File Extension	Yes No (default)	Specifies if the model file extension is truncated from the file name when using the CAD model file name to generate a CAD document name.
Upload Drop Number File Extension	Yes No (default)	Specifies if the model file extension is truncated from the file name when using the CAD model file name to generate a CAD document number.
Use Current Attribute Mapping	<comma-separated list>	Allows the selective set of designated parameters to be mapped to Windchill attributes using the current attribute mapping definition instead of the mapping that was previously used. List the name of parameters, separated by , (comma). This option is applicable for parameters mapped to Windchill global attribute with type floating number with units only.
Use Explicit Mapping for Designated Parameter Creation	Yes No (default)	This option controls the name of the designated parameter to be created in the file. If the attribute name and parameter name in the file are explicitly mapped, and if the

Preference	Values	Description
		corresponding parameter does not exist in the file, this option allows the user to control the name of the parameter to be created in the file. The default value is "No". Set to No, the name of the parameter will be same as Windchill attribute name. Set to Yes, the name of the parameter will be based on the explicit attribute mapping.

Note

The Upload category also has a sub-category for setting collection defaults. For more information see [Collection-related Preferences for Actions on page 190](#).

Revise

The following table lists preferences for creating new revisions:

Preference	Values	Description
Allow Override On Insert	Yes No (default)	If set to “Yes” will display insert actions and allow user to specify a revision label.
Allow Override On Revise	Yes No (default)	If set to “Yes” will allow user to specify a revision label when creating a new revision.
Allow Override On Create CAD Document	Yes No (default)	If set to “Yes”, will allow the user to set the revision of an uploaded object. If set to “No”, when a new CAD document is checked in, it will use the initial revision label in the sequence.
Allow revise of non-latest revisions	Yes No (default)	If set to “Yes”, will allow the user to revise non-latest revisions for valid object types

Preference	Values	Description
Revision Label Picker Display Count	Integer (default is 10)	Specifies the number of revision labels displayed in the revision label picker.

Save As

The following table describes the preferences for the Save As category that are not included in a Save As subcategory:

Preferences	Values	Description
Allow Replace	Yes No (default)	Controls the availability of the "Replace" button in the Save As UI. The "Replace" button allows a user to perform a global replace of one object in the structure with a different object in the commonspace. The default is "No" (replacement is not allowed), and it is strongly recommended to use the default setting. Replace is only possible for Creo Parametric CAD documents, and this control is not the same as replacing the component in Creo Parametric. It does not guarantee retrieval of the assembly with the replaced component.
Inherit FileName from Name	Yes No (default)	Controls how the file name is assigned by default for a new CAD document using the Save As command in the workspace and commonspace. When set to "Yes", the new file name is set to the same

Preferences	Values	Description
		value as the new CAD document name plus an appropriate extension. When set to "No" (default), preferences for CAD Document Filename Prefix and CAD Document Filename Suffix will be applied to the original file name to produce a new name.
Inherit FileName from Number	Yes No (default)	Controls how the file name is assigned by default for a new CAD document using the Save As command in the workspace and commonspace. Possible values are "Yes" or "No" (default) . When set to "Yes", the new filename will be set to the same as value as the new CAD document number plus an appropriate extension. When set to "No", the new filename is set based on changing the original filename using the CAD Document Filename Prefix and CAD Document Filename Suffix preferences.
Save Selected Objects Only	Yes No (default)	If set to "Yes," only selected objects are marked for Save As and dependent objects are marked for reuse. The default setting is "No." By default selected and dependent objects are all eligible for Save As.

The following table lists the preferences that control the default collection of related objects during Commonsense Save As (subcategory: From Commonsense Collector):

Preferences	Values	Description
Display collected objects	As a List As a Structure As a Structure with Associated Objects	Allow specification of the way collected objects are listed in the table. Each display allows viewing how objects have been collected by displaying the different association in a specific way. Default is "As a list".
Include dependent CAD / Dynamic documents	All Required (default) None	Specifies which dependent CAD / Dynamic Documents for the collected CAD /Dynamic Documents will be by default added to the collection
Include dependent Parts	All None (default)	Specifies which dependent parts for the collected parts will be by default added to the collection.
Include related CAD Documents	All Initially Selected Only None (default)	Specifies which CAD Documents associated to the collected parts will be by default added to the collection.
Include related Drawings	All Initially Selected Only None (default)	Specifies which drawings associated to the collected CAD documents or parts will be by default added to the collection.
Include related Family table objects	All Initially Selected Only None (default)	Specifies which family table objects related to the collected generic or instances will be by default added to the collection.
Include related Generics	All Initially Selected Only None (default)	Specifies which generics associated to the collected instances will be by default added to the

Preferences	Values	Description
		collection.
Include related Image objects	All Initially Selected Only None (default)	Specifies which image objects related to the collected source documents will be by default added to the collection.
Include related Notes	All None (default)	Specifies which notes associated to the collected parts will be by default added to the collection.
Include related Parts	All Initially Selected Only None (default)	Specifies which parts associated to the collected documents, CAD documents, or dynamic documents will be by default added to the collection.
Include related Source objects	All Initially Selected Only None (default)	Specifies which source objects related to the collected representation documents will be by default added to the collection.

The following table lists preferences that control the default collection of related objects during Workspace Save As (subcategory: From Workspace Collector):

Preferences	Values	Description
Display collected objects	As a List As a Structure As a Structure with Associated Objects	Allow specification of the way collected objects are listed in the table. Each display allows viewing how objects have been collected by displaying the different association in a specific way. Default is "As a list".
Include dependent CAD / Dynamic documents	All Required (default) None	Specifies which dependent CAD / Dynamic Documents for the collected CAD /Dynamic Documents will be by default added to the collection

Preferences	Values	Description
Include dependent Parts	All None (default)	Specifies which dependent parts for the collected parts will be by default added to the collection.
Include related CAD Documents	All Initially Selected Only None (default)	Specifies which CAD Documents associated to the collected parts will be by default added to the collection.
Include related Drawings	All Initially Selected Only None (default)	Specifies which drawings associated to the collected CAD documents or parts will be by default added to the collection.
Include related Family table objects	All Initially Selected Only None (default)	Specifies which family table objects related to the collected generic or instances will be by default added to the collection.
Include related Generics	All Initially Selected Only None (default)	Specifies which generics associated to the collected instances will be by default added to the collection.
Include related Image objects	All Initially Selected Only None (default)	Specifies which image objects related to the collected source documents will be by default added to the collection.
Include related Notes	All None (default)	Specifies which notes associated to the collected parts will be by default added to the collection.
Include related Parts	All Initially Selected Only None (default)	Specifies which parts associated to the collected documents, CAD documents, or dynamic documents will be by default added to the collection.
Include related Source	All	Specifies which source

Preferences	Values	Description
objects	Initially Selected Only None (default)	objects related to the collected representation documents will be by default added to the collection.

The following table lists preferences for the Naming Patterns subcategory of the Save as preferences:

Preferences	Values	Description
CAD Document Filename Prefix	< naming pattern > *.* (default)	Specifies a pattern for filename prefix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *.*.
CAD Document Filename Suffix	< naming pattern > *_*.* (default)	Specifies a pattern for filename suffix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *_*.*.
CAD Document Name Prefix	< naming pattern > *.* (default)	Specifies a pattern for CAD document name prefix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *.*.
CAD Document Name Suffix	< naming pattern > *_*.* (default)	Specifies a pattern for CAD document name suffix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *_*.*.
CAD Document Number Prefix	< naming pattern > *.* (default)	Specifies a pattern for CAD document number prefix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *.*.

Preferences	Values	Description
CAD Document Number Suffix	< naming pattern > * _.* (default)	Specifies a pattern for CAD document number suffix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is * _.*
Part Name Prefix	< naming pattern > *.* (default)	Specifies a pattern for part name prefix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *.*
Part Name Suffix	< naming pattern > * _.* (default)	Specifies a pattern for part name suffix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is * _.*
Part Number Prefix	< naming pattern > *.* (default)	Specifies a pattern for part number prefix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is *.*
Part Number Suffix	< naming pattern > * _.* (default)	Specifies a pattern for part number suffix change during Save As action. Use characters and wildcard (*) to set the pattern for the new value. The default is * _.*

Windchill Workgroup Manager

Windchill Workgroup Manager preference category includes many preferences that apply to client settings for Creo Parametric, in addition to the Windchill Workgroup Manager. In many cases, specific preference settings can be specified on a CAD

tool by CAD tool basis. The following sections describe the preferences for the sub-categories under the Windchill Workgroup Manager category of particular interest to Creo Parametric users.

Workgroup Manager Client Preferences

The following table describes client-side preferences for the Windchill Workgroup Manager that can be managed with the Windchill **Preference Management** utility.

Note

The preferences in the sub-categories Family Table Instance Creation and Mapping System Attributes and File Properties are listed in separate tables that follow the main table. See [Family Table Instance Creation Preferences on page 216](#) and [Mapping System Attributes and File Properties Preferences on page 218](#).

Preference	Values	Description
Attach Differences Report upon Check In	Yes No (default)	Controls the default behavior whether or not to generate and attach Differences report on objects that are going to be checked in. This option is applicable in Creo Parametric embedded browser for CAD documents authored by Creo Parametric.
Enable Support for Parameters or Properties with Units	Yes No (default)	Controls whether to enable the feature of supporting parameters or properties with units. If set to Yes—Enable the feature of supporting parameter parameters or properties with units. Before the system provides migration utility, The user must create new global attribute type and assign the default unit manually. If set to No—Disable the feature of supporting parameters or properties with units.
Note default subtype	<code>\${internet_domain_name}.Note</code> (default)	Internal name of Note default subtype. The " <code>\${internet_domain_name}</code> " will expand to the default exchange domain name of the Notes internal name as created on install. If this preference

Preference	Values	Description
		value is changed the internal name must also be changed for the Note subtype in the Type and Attribute Management utility.
Open In CAD Tool For Nonnative Objects	Yes No (default)	Certain licensed modules of Pro/ENGINEER Wildfire 4.0 and later Creo Parametric releases support opening NX and CATIA V5-authored models in Pro/ENGINEER. This option will allow user to control whether or not the "Open in Pro/ENGINEER" action is displayed for these non-native CAD documents. If your Pro/ENGINEER license does not support this type of import or you are not using Pro/ENGINEER Wildfire 4.0+, you may want to change the value of this option to No. Allowed values are "Yes" or "No". Default value is "Yes".
Search Path for Automatically Attach Files on Upload		Specifies paths on disk, separated by a semicolon (;), searched for attachments that need to be automatically added to a CAD or dynamic document upon upload. Order in the preference determines the order in which the directories are searched. User may use environment variables in search path specifying the environment variable as \${environment variable name}.
Synchronize Number and File Name	Yes (default) No	Allows file name and number to be the same. This option is applied when creating new files from an authoring application. Number may be with or without file extension, depending on the Upload Drop Number File

Preference	Values	Description
		Extension preferences.
Undo Checkout Overwrite Local Content	Yes No (default)	<p>Specifies if the model content is overwritten in cache by default when using Undo Checkout. Allowed values: Yes or No. The default value is No.</p> <p>Yes—Locally modified files will be replaced with the last checked in version of the file.</p> <p>No—The CAD/dynamic document will be in a checked in state; but the workspace will still contain the local modified contents.</p>
Update Overwrite Local Content	Yes No (default)	<p>Specifies if the model content is overwritten in cache by default when using Update from a Windchill Workgroup Manager. Allowed values: Yes or No. The default value is No.</p> <p>Yes—Locally modified files will be replaced with the updated copy of the file from latest CAD/dynamic document.</p> <p>No—The CAD/dynamic document version in the workspace will be updated; but the local modified contents will not be over-written. Specifies if the model content is overwritten in cache by default when performing Update from the Windchill workspace. (Creo Parametric File > Update is controlled by the config.pro option, dm_overwrite_contents_on_update).</p>
Upload After Native Save	Yes No (default)	Specifies if the model is uploaded after a native Save action in the authoring application (for example, File > Save).

Preference	Values	Description
		<p>Yes—Content will be uploaded after each native authoring application Save action.</p> <p>No—A native File > Save will not start an upload in the background.</p>
Upload CAD/Dynamic Document and Attachment Filter		<p>Allows configuring the content that is uploaded as additional content of a model. Specifies the triplets which are used to determine what to autoattach. The elements of each triplet are:</p> <p>CAD/Dynamic Document Type—Name pattern to find the CAD or dynamic document to autoattach to on upload.</p> <p>Attachment file type—File name pattern for the file to auto attach.</p> <p>Content Category—The category with which to autoattach.</p> <p>The value of this preference is a string created by concatenating a series of above triplets in the form: [CAD/Dynamic Document Type],[Attachment file type],[Content Category];[CAD/Dynamic Document Type],[Attachment file type],[Content Category];... Wildcards can be used in [CAD/Dynamic Document Type] pattern to specify the CAD / Dynamic Document(s) to do autoattach to. Same wildcards can be put in [Attachment file type] pattern to use similar name to specify the file(s) to autoattach. In an environment where multiple authoring applications are in use, it is recommended to</p>

Preference	Values	Description
		<p>specify this preference value for particular authoring applications, not the general value.</p> <p>Example: *.CATProduct,*.CAT-Process,MANUFACTURING;*.CATPart,*.CATAnalysis,ANALYSIS_INPUT This string triplet specifies that any CATProduct or CATPart found in the CAD Document upload list needs to be autoattached to any CATProcess or CATAnalysis found with same name as the CAD document in the upload.autoattach.searchpath path preference. Any CATProcess or CATAnalysis thus found are attached with MANUFACTURING or ANALYSIS_INPUT content category to CATProduct or CATPart resp. The content category name must be one of the names recognized by Windchill.</p>
Upload Related Drawings	<p>Yes</p> <p>No (default)</p>	<p>Native upload or auto check in of CAD documents will also upload or check in associated drawings.</p> <p>Yes—Related drawings will be uploaded or checked in with CAD Documents.</p> <p>No—Related drawings will not be uploaded or checked in with CAD documents.</p>
Workspace Frame Stack Size	<integer, non-negative>	<p>Controls the number of workspace frames to be maintained. The input must be in integer value, which is equal to or greater than 0. The default value is 0. This option is applicable for the Creo Parametric embedded browser only and will take effect after a Creo Parametric session is initialized.</p>

Design in Context Preferences

The following table lists preferences related to design in context

Preference	Values	Description
Allow Creation and Editing of Creo External Simplified Reps in Windchill	Yes No (default)	When the set to its default value, “No,” the creation of a Creo Parametric design context (external simplified rep or ESR) in Windchill will also be disabled. This preference takes priority over the preference, Show New Design Context Action . If set to Yes, Creo Parametric saves the ESR, and you will be able to create and edit ESRs in Windchill; however, if the ESR contains envelopes, you may not be able to upload it or save any changes in the Edit Design Context Definition page. Additionally, the display of the structure in the Edit Definition and Structure tabs of the ESR’s information page may be inaccurate.
Default Design Context Definition Rule	Assembly Only (default) Full Lightweight Geometry	Determines the default rule when defining a new design context. The default value is "Assembly Only", but the following are possible values: <ul style="list-style-type: none"> • Assembly Only—Exclude all members except the top-level and then select the ones to add. • Full—Include all members and then select the ones to

Preference	Values	Description
		<p>remove. Included models will fully load, such that the geometry may be modified. For Creo Parametric this is "Master Rep" and for CATIA V5 this is "Design Mode".</p> <ul style="list-style-type: none"> • Lightweight—Include all members and then select the ones to remove. Included models will load the geometry in the lightest possible way to minimize retrieval time. For Creo Parametric this is "Graphics Rep" and for CATIA V5 this is "Visualization Mode". • Geometry—Include all members and then select the ones to remove. Included models will load the geometry such that mass property may be performed. For Creo Parametric this is "Geometry Rep" and for CATIA V5 this is "Visualization Mode".
Default Representation	Full Geometry Lightweight	Determines level of detail passed from the Configuration Context to the Design Context. The default value is "Lightweight", but the following values are

Preference	Values	Description
		<p>possible</p> <ul style="list-style-type: none"> • Full—Fully load the model, such that the geometry may be modified. For Creo Parametric this is "Master Rep" and for CATIA V5 this is "Design Mode". • Geometry—Load the geometry such that mass property may be performed. For Creo Parametric this is "Geometry Rep" and for CATIA V5 this is "Visualization Mode". • Lightweight—Load the geometry in the lightest possible way to minimize retrieval time. For Creo Parametric this is "Graphics Rep" and for CATIA V5 this is "Visualization Mode".
Show New Design Context Action	Yes No (default)	Determines whether to show the New Design Context action on the Structure tab of a Creo Parametric or CATIA V5 assembly CAD document's workspace information page. Creo/Elements Pro 5.0 and the Windchill 10.0 M010 Workgroup Manager for CATIA V5 support design contexts created in Windchill. If you are using Wildfire 4.0 or the 10.0 F000 CATIA V5 WGM, this new feature is not supported. You may

Preference	Values	Description
		show or hide the action to create a design context with this preference based on the tools that your company is currently using. The default is "No" or not to show the action.

:

Family Table Instance Creation Preferences

The following table lists preferences related to family table instance creation:

Preference	Values	Description
Ignore managing model instances by specific names	<"*" -separated list>	Allows specifying a set of model instance names that should not be managed as family table CAD Documents. When model instance name matches one that is listed to ignore, family table instance will not be created. Wild cards can used to define naming filters. Multiple names should be separated by /. For example: Default/Flat*/bend.
Include generic name with instance	Yes No (default)	Default behavior is to record the instance name as the CAD document File Name. In order to avoid potential uniqueness issues, this option allows including the generic name as part of the CAD document File Name of the family table instance.
Manage model instances as family table CAD Documents	All (default) Parts Disable	Controls whether CAD model instances will be managed in Windchill as family table CAD documents and if so, will it support assembly

Preference	Values	Description
		<p>instances.</p> <p>All—Part and Assembly instances will be published as CAD Document instances.</p> <p>Parts—Enable managing instances defined in parts as managed functional instance.</p> <p>Disable—Instances will not get published as CAD Document instances.</p>
Manage new model instances by default	<p>Yes</p> <p>No (default)</p>	<p>Controls the default value for the Instance/Non-Instance attribute "PTC_IS_INSTANCE".</p> <p>No—All newly created model instances will get the value PTC_IS_INSTANCE=-No. This means by default newly created instances will not be created as CAD documents. One can set this value to Yes for any instance that should be managed as CAD documents.</p> <p>Yes—All newly created model instances will get the value PTC_IS_INSTANCE=Yes. This means by default newly created model instances will be created as CAD documents.</p>

Mapping System Attributes and File Properties Preferences

The following table lists the preferences related to system attribute mapping and file properties:

Preference	Values	Description
CAD Document Iteration System Attribute	PTC_WM_ITERATION (default)	Identifies the name of the property in the CAD tool that shows the CAD document's iteration. The value may be set to a CAD system attribute. The default value is PTC_WM_ITERATION.
CAD Document Life Cycle State System Attribute	PTC_WM_LIFECYCLE_STATE (default)	Identifies the name of the property in the CAD tool that shows the life cycle state of the CAD document.
CAD Document Life Cycle System Attribute	PTC_WM_LIFECYCLE (default)	Identifies the name of the property in the CAD tool that shows the life cycle of the CAD document.
CAD Document Name System Attribute	PTC_WM_NAME (default)	Identifies the name of the property in the CAD tool that shows the CAD document's Name. The value may be set to a CAD system attribute. The default value is PTC_WM_NAME.
CAD Document Number System Attribute	PTC_WM_NUMBER (default)	Identifies the name of the property in the CAD tool that shows the CAD document's Number. The value may be set to a CAD system attribute. The default value is PTC_WM_NUMBER.
Change Note Attribute	PTC_WM_LAST_CHANGE_NOTE (default)	Identifies the name of the property that shows the text added to the last checkin.
Created By Attribute	PTC_WM_CREATED_BY (default)	Identifies the name of the property that shows the name of the user that created the CAD document.
Created On Attribute	PTC_WM_CREATED_ON	Identifies the name of the

Preference	Values	Description
	(default)	property that shows the date and time that the CAD document was created.
Drawing System Attribute	PTC_WM_IS_DRAWING (default)	Identifies the name of the attribute that determines if an NX part is a drawing. The default is set to PTC_WM_IS_DRAWING.
Modified By Attribute	PTC_WM_MODIFIED_BY (default)	Identifies the name of the property that shows the name of the user that last modified the CAD document.
Modified On Attribute	PTC_WM_MODIFIED_ON (default)	Identifies the name of the property that shows the date and time that the CAD document was last modified.
Organization ID System Attribute	PTC_WM_ORGANIZATION_ID (default)	Identifies the name of the property in the CAD tool that shows the CAD document's organization id. The value may be set to a CAD system attribute. The default value is PTC_WM_ORGANIZATION_ID.
Part Name System Attribute	PTC_WM_PART_NAME (default)	Identifies the name of the property in the CAD tool that shows the CAD document's Owner-associated part Name. The value may be set to a CAD system attribute. The default value is PTC_WM_PART_NAME.
Part Number System Attribute	PTC_WM_PART_NUMBER (default)	Identifies the name of the property in the CAD tool that shows the CAD document's Owner-associated part Number. The value may be set to a CAD system attribute. The default value is

Preference	Values	Description
		PTC_WM_PART_NUMBER.
Part Revision Attribute	PTC_WM_PART_REVISION_FOR_DRAWING (default)	Identifies the name of the property that shows the revision of the part associated with the drawing.
Part State Attribute	PTC_WM_PART_STATE_FOR_DRAWING (default)	Identifies the name of the property that shows the life cycle state of the part associated with the drawing.
Revision System Attribute	PTC_WM_REVISION (default)	Identifies the name of the property in the CAD tool that shows the CAD document's Revision. The value may be set to a CAD system attribute. The default value is PTC_WM_REVISION.

Workspace Preferences

The following table describes the preferences in the Add to Workspace and Check Out sub-category:

Preference	Values	Description
Add Primary Contents to Workspace	DOWNLOAD (default) LINK	Controls the default behavior whether content should be added to workspace as link or the file should be available. The default is DOWNLOAD (primary content is downloaded). When set to LINK, primary content is not downloaded, but a link to the content is created for later download as required.
Open in Application	Yes No (default)	Controls the default behavior whether initially

Preference	Values	Description
		selected objects should also be opened in the authoring application when Add to Workspace or Check Out is performed using the Check Out or Add to Workspace user interface. Check Out performed without the user interface will not open in authoring application regardless of this option. When set to "Yes," specifies that the primary content file is automatically opened in the CAD application upon Check Out.
Reuse Content in Target Workspace	Yes (default) No	Controls the default behavior of how content should be handled when there is already content in the target workspace cache. When set to "Yes," (default) specifies to reuse content existing in workspace. When set to "No," specifies to download new content from server.
Set Configuration for Add to Workspace	DEFAULT (default) LATEST AS_STORED	Controls the default configuration when Add to Workspace action is performed. The DEFAULT option applies latest configuration for the latest iteration, and as stored configuration for the non-latest iteration. When set to LATEST, the default configuration is set to latest and when set to AS_STORED, the

Preference	Values	Description
		default configuration is set to As Stored irrespective of the iteration of the object(s) selected for the action.
Set Configuration for Check Out	DEFAULT (default) LATEST AS_STORED	Controls the default configuration when Check Out action is performed. DEFAULT (default) option applies latest configuration for the latest iteration, and as stored configuration for the non-latest iteration. When set to LATEST, the default configuration is set to latest and when set to AS_STORED, the default configuration is set to As Stored irrespective of the iteration of the object(s) selected for the action.
Set for Check Out	SELECTED_AND_MODIFIED (default) SELECTED REQUIRED ALL	Controls the default set of objects to be checked out. The default is SELECTED_AND_MODIFIED (the initially selected object(s) and any modified dependents are marked as Set for Check Out. The value SELECTED marks only the initially selected object(s). The value REQUIRED marks initially selected object(s) and their required dependents. The value ALL marks initially selected object(s) and all their dependents.

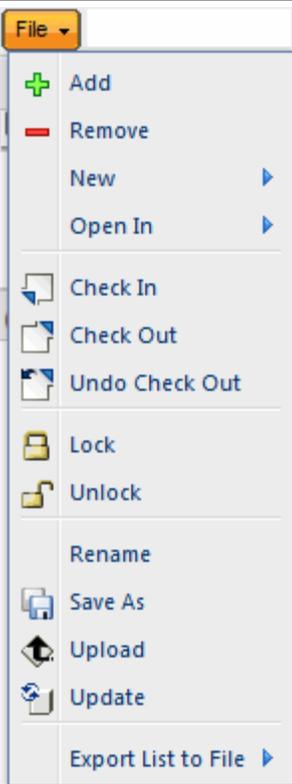
The following table describes the preference applicable to editing workspace preferences:

Preference	Values	Description
Allow Effectivity for CAD Documents	Yes (default) No	Specifies whether effectivity configuration specification is applicable to EPM documents in the workspace. If set to "Yes," allows the user to set effectivity configuration specification for EPM documents in the workspace. If set to "No," use of effectivity is not allowed for such objects.

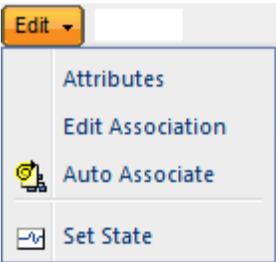
4

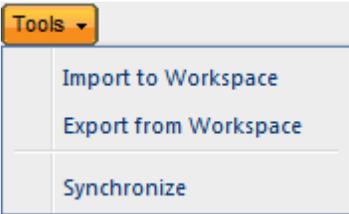
Quick Reference for Menus, Icons, and Symbols

Use the next pages as a quick reference for the menu commands, action icons, and status symbols used in Creo Parametric with Windchill.

File Menu Selections	
	<p>Add allows you to add objects to your workspace.</p> <p>Remove allows you to remove selected objects from the workspace.</p> <p>New allows you to create a new CAD document, graphics dynamic document, or part; or to create a new revision of a selected object.</p> <p>Open allows you to open a selected CAD document's Creo Parametric file or to open a Creo View representation of a selected object.</p> <p>Check In begins the check-in process for selected objects.</p> <p>Check Out begins the check-out process for selected objects.</p> <p>Undo Check Out removes the check-out status on a selected object and discards any local modifications.</p> <p>Lock allows you to make a selected workspace object read-only.</p> <p>Unlock removes the read-only status from a selected object.</p>

File Menu Selections	
	<p>Rename allows you to change the name of selected workspace objects (only if the object has never been checked in).</p> <p>Save As allows you to save a copy of a selected object as a new workspace object (not committed to common-space until checked in).</p> <p>Upload places a selected local object in the server-side workspace.</p> <p>Update compares workspace objects with the workspace configuration specification and replaces the workspace version with a version available on the server, if appropriate.</p> <p>Export List to File allows you to export the workspace Object List to a file in one of the following formats: CSV, HTML, TEXT, XLS, XLSX, XLS Report, XML.</p>

Edit Menu Selections	
	<p>Attributes begins the process of editing attributes for checked-out, selected objects.</p> <p>Edit Association allows you to manually edit the association of objects.</p> <p>Auto Associate begins the process of automatically finding or creating parts to be associated with selected CAD documents.</p> <p>Set State allows you to set a life cycle state for a selected object.</p>

Tools Menu Selections	
	<p>Import to Workspace allows you to bring objects into the workspace from a local directory.</p> <p>Export from Workspace allows you to export workspace objects to a target directory.</p> <p>Synchronize refreshes workspace objects to reflect changes made on the server (for example, a name change).</p>

Workspace Actions Menu Selections	
	<p>Activate allows you to make an inactive workspace active (embedded mode only).</p> <p>Event Management opens the Event Management window for the server with which you are working.</p> <p>Frame commands allow you to apply, lock, or purge frames.</p> <p>Edit Preferences opens the Edit Workspace Options window to view or edit your workspace configuration specification.</p> <p>Delete Workspace allows you to delete the current inactive workspace.</p>

Workspace Action Icons –	Toolbar and Row Actions	
		
 Remove from Workspace	 Update	 Add to Workspace
 Upload	 Auto Associate	 Find in List
 Check In	 New Revision	 Open Information page (row action)
 Check Out	 New Part	 Open in Creo Parametric (row action)
 Undo Checkout	 New CAD Document	 Open in Creo View (row action)

Share Status
 Shared to a project (for Windchill PDMLink only)
 Shared from PDM (for Windchill ProjectLink only)
 Checked-out from PDM (for Windchill ProjectLink only)

General Status
 Locked (Object is read only. Only applicable to objects in the local cache).
 Checked out by you
 Checked out by you in another workspace (seen in the workspace and action pages accessed in the context of a workspace)
 New locally
 Checked out by another user

General Status
☒ Checked-out to a project (for Windchill PDMLink only)
☑ Another Iteration is checked out by you
☒ Another Iteration is checked out by another user
Local Workspace Status
+ Modified locally
Modified Status
⚙ Modifications Need to be Uploaded
+ Modifications uploaded
Modified and not eligible for upload
Out of Date Status
🕒 Out of date - Modified by you
🕒 Out of date - Modified by another user
Out of Date with Workspace Configuration
🕒 Out of date with Workspace configuration - Modified by you
🕒 Out of date with Workspace configuration - Modified by another user

5

Using OIRs for Naming and Numbering

Setting Name and Number to the Same, Non-editable Autogenerated Value.....	231
Turning Off All Autonumbering	233
Setting Editable Autogenerated Values.....	235
Setting Non-editable Autogenerated Values	238
Setting Editable, Identical Value for Name and Number.....	240
Setting Editable, Non-autogenerated Values	242
Setting Autogenerated, Non-editable Values for Number.....	244
Setting Pre-generated, Editable Values	246
Setting Pre-generated, Non-editable Values.....	249

Object Initialization Rules (OIRs) specify object attributes and values to be applied at the time of object creation. This appendix lists sample rules that determine naming and numbering only in the case when using Save As to create a new object that is based on an existing object. For more information about OIRs in general, see the *Windchill Business Administrator's Guide*. The example rules in the following sections address the following scenarios:

- Setting both the name and the number to the same autogenerated value
- Turning off all autonumbering
- Setting editable autogenerated values
- Setting non-editable autogenerated values
- Setting editable, identical value for name and number
- Allowing editable, non-autogenerated values
- Setting autogenerated, non-editable values for number

-
- Setting pre-generated, editable values
 - Setting pre-generated, non-editable values

Setting Name and Number to the Same, Non-editable Autogenerated Value

The following rule sets both Name and Number to the same autogenerated value.

Note

*Setting the preference, **Save As ▶ Inherit FileName From Number** to Yes in the **Preference Manager** ensures that the CAD Name is the same as the autogenerated number with the proper extension.*

Example:

```
- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- Define a variable to hold the generated number
-->
- <VarDef id="GeneratedNumber"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</VarDef>
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
```

```

</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number">
<VarRef id="GeneratedNumber" />
</AttrValue>
- <AttrValue id="name">
<VarRef id="GeneratedNumber" />
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>

```

```

- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
- <!-- <Value
algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>-->
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Turning Off All Autonumbering

The following rule stipulates that autonumbering for both number and name is turned off. No OIRs are specified for either attribute and the user is expected to enter their values in the **Set New Name** window manually. However, the system would still render their new values by using the values of the preferences for the default prefixes and suffixes. The user can override the system-rendered values.

Example:

```

- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">

```

```

- <Arg>- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <AttrValue id="number"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator" ignore="true">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <AttrValue id="name"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator" ignore="true">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">

```

```

<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Editable Autogenerated Values

Upon clicking **OK** in the **Set New Name** window, both number and name are generated automatically using the respective OIRs. Their values would not be visible in the UI (the fields display the string "(Generated).") However, the user is able to override the autogenerated values by manually entering new values in the UI as the corresponding fields.

Example:

```

- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"

```

```

algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttributeAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAlgorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.NumberGenerator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <AttrValue id="name"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.NumberGenerator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">

```

```

<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Non-editable Autogenerated Values

As in the preceding example, Name and Number are set to autogenerated values; however, the user is unable to edit the fields to override the OIRs.

Example:

```
- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <AttrValue id="name"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator">
```

```

<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"

```

```

algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Editable, Identical Value for Name and Number

In this example, both the number and name receive the same pre-generated value which will be displayed in the corresponding fields. They are visible to the user and editable.

Example:

```

- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- Define a variable to hold the generated number
-->
- <VarDef id="GeneratedNumber"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</VarDef>
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>

```

```

- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number">
<VarRef id="GeneratedNumber" />
</AttrValue>
- <AttrValue id="name">
<VarRef id="GeneratedNumber" />
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value

```

```

algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Editable, Non-autogenerated Values

This example is similar to the scenario for turning off all autonumbering, except that empty constraints are specified explicitly for number and name. This is recommended if OIRs are defined for these attributes.

Example:

```

- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->

```

```

- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgor
ithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>

```

```

- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
<AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints" />
<AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints" />
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Autogenerated, Non-editable Values for Number

In this example, only Number is automatically generated using the specified OIR and the new value is not visible in the Set New Name window, which instead shows the string "(Generated)." The field is not editable. This scenario is usually the out-of-the-box default.

Example:

```

- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"

```

```

algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAl
gorithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number"
algorithm="com.ptc.windchill.enterprise.revisionControlled.server.impl.Number
Generator">
<Arg>{ GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0} </Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>

```

```

- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>

```

Setting Pre-generated, Editable Values

In this example, both the number and name fields display pre-generated values calculated using the respective OIRs. The user can override these pre-generated values.

Example:

```
- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
- <AttrValue id="number" algorithm="wt.rule.algorithm.StringConstant">
<Arg>Pre Generated Number</Arg>
</AttrValue>
- <AttrValue id="name" algorithm="wt.rule.algorithm.StringConstant">
<Arg>Pre Generated Name</Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
```

```

- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>

```

```
</AttrConstraint>
</AttributeValues>
```

Setting Pre-generated, Non-editable Values

This is same as the preceding example, except that the fields showing number and name would not be editable. The only difference between the `GetServerPreGeneratedValue` and `GetServerAssignedConstraint` is that the former calculates the value before the Set New Name page is displayed, and that value is therefore visible to the user; the latter calculates the value after the user clicks the OK button and therefore that value is not visible on the page.

Example:

```
- <AttributeValues objType="wt.epm.EPMDocument">
- <!-- set the folder
-->
- <AttrValue id="folder.id"
algorithm="com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgo
rithm">
<Arg>/Default</Arg>
</AttrValue>
- <!-- set the lifecycle
-->
- <AttrValue id="lifeCycle.id"
algorithm="com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttr
ibuteAlgorithm">
- <Arg>
- <!-- Translation of the word "Basic" must be the same as the translation done in
commonLifeCycles.xml
-->
<?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>
Basic
<?loc-end ?>
</Arg>
</AttrValue>
- <!-- set the team template
-->
- <AttrValue id="teamTemplate.id"
algorithm="com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAl
gorithm">
<Arg>Default</Arg>
</AttrValue>
- <!-- set the number to a generated number
-->
```

```

- <AttrValue id="number" algorithm="wt.rule.algorithm.StringConstant">
<Arg>Pre Generated Number</Arg>
</AttrValue>
- <AttrValue id="name" algorithm="wt.rule.algorithm.StringConstant">
<Arg>Pre Generated Name</Arg>
</AttrValue>
- <!-- set the version info to a generated version info
-->
- <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
<Arg>wt.series.HarvardSeries</Arg>
</AttrValue>
- <!-- specify AttrConstraint tag
-->
- <AttrConstraint id="lifeCycle.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="lifeCycle"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="folder.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
</AttrConstraint>
- <AttrConstraint id="number"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="name"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value

```

```
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="teamTemplate"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value
algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint" />
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint" />
</AttrConstraint>
- <AttrConstraint id="organization.id"
algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
- <Value
algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue">
<Value
algorithm="com.ptc.windchill.enterprise.org.server.impl.OwningOrgDefaultAlgo
rithm" />
</Value>
</AttrConstraint>
</AttributeValues>
```