Licensed Third-Party Technology Information

Certain PTC software products contain licensed third-party technology: Rational Rose 2000E is copyrighted software of Rational Software Corporation. RetrievalWare is copyrighted software of Convera Corporation. VisualCafe is copyrighted software of WebGain, Inc. VisTools library is copyrighted software of Visual Kinematics, Inc. (VKI) containing confidential trade secret information belonging to VKI. HOOPS graphics system is a proprietary software product of, and is copyrighted by, Tech Soft America, Inc. G-POST is copyrighted software and a registered trademark of Intercim. VERICUT is copyrighted software and a registered trademark of CGTech. Pro/PLASTIC ADVISOR is powered by Moldflow technology. Moldflow is a registered trademark of Moldflow Corporation. The JPEG image output in the Pro/Web.Publish module is based in part on the work of the independent JPEG Group. DFORMD.DLL is copyrighted software from Compaq Computer Corporation and may not be distributed. METIS, developed by George Karypis and Vipin Kumar at the University of Minnesota, can be researched at http://www.cs.umn.edu/~karypis/metis. METIS is © 1997 Regents of the University of Minnesota. LightWork Libraries are copyrighted by LightWork Design 1990-2001. Visual Basic for Applications and Internet Explorer is copyrighted software of Microsoft Corporation. Adobe Acrobat Reader is copyrighted software of Adobe Systems. Parasolid © Electronic Data Systems (EDS). Windchill Info*Engine Server contains IBM XML Parser for Java Edition and the IBM Lotus XSL Edition. Pop-up calendar components Copyright © 1998 Netscape Communications Corporation. All Rights Reserved. TECHNOMATIX is copyrighted software and contains proprietary information of Technomatix Technologies Ltd. Apache Server, Tomcat, Xalan, and Xerces are technologies developed by, and are copyrighted software of, the Apache Software Foundation (http://www.apache.org/) - their use is subject to the terms and limitations at: http://www.apache.org/LICENSE.txt. UnZip (© 1990-2001 Info-ZIP, All Rights Reserved) is provided "AS IS" and WITHOUT WARRANTY OF ANY KIND. For the complete Info-ZIP license see
ftp://ftp.info-zip.org/pub/infozip/license.html. Gecko and Mozilla components are subject to the Mozilla Public License Version 1.1 at http://www.mozilla.org/MPL/. Software distributed under the MPL is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the MPL for the specific language governing rights and limitations. Technology "Powered by Groove" is provided by Groove Networks, Inc. Technology "Powered by WebEx" is provided by WebEx Communications, Inc. Acrobat Reader is Copyright © 1998 Adobe Systems Inc. Oracle 8i run-time, Copyright © 2000 Oracle Corporation. The Java™ Telnet Applet (StatusPeer.java, TelnetIO.java, TelnetWrapper.java, TimedOutException.java), Copyright © 1996, 97 Mattias L. Jugel, Marcus Meißner, is redistributed under the GNU General Public License. This license is from the original copyright holder and the Applet is provided WITHOUT WARRANTY OF ANY KIND. You may obtain a copy of the source code for the Applet at http://www.mud.de/se/jta (for a charge of no more than the cost of physically performing the source distribution), by sending e-mail to leo@mud.de or marcus@mud.de-you are allowed to choose either distribution method. The source code is likewise provided under the GNU General Public License. GTK+The GIMP Toolkit are licensed under the GNU LPGL. You may obtain a copy of the source code at http://www.gtk.org/, which is likewise provided under the GNU LPGL. zlib software Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

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), is 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.

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA
Contents

Change Record............................................................................................................. xv

About This Guide........................................................................................................... xix
  Intended Audience........................................................................................................ xix
  Overview...................................................................................................................... xix
  Chapter Contents ....................................................................................................... xx
  Related Documentation............................................................................................... xxi
  Technical Support...................................................................................................... xxi
  Documentation for PTC Products................................................................................ xxi
  Comments.................................................................................................................. xxii
  Documentation Conventions....................................................................................... xxii
  Third-Party Products.................................................................................................. xxiii
  Code Examples......................................................................................................... xxiii

Administering Runtime Services ................................................................................. 1-1
  Windchill Configuration Properties .............................................................................. 1-2
    The site.xconf File Format and Contents.................................................................. 1-3
    The xconf.dtd File ................................................................................................ 1-4
  Using the System Configurator .................................................................................. 1-6
    Starting and Stopping the Windchill System.............................................................. 1-7
    Modifying System Properties ................................................................................ 1-7
    Working with Log Files ............................................................................................... 1-9
  Using the xconfmanager Utility .................................................................................. 1-10
    xconfmanager Command Syntax ............................................................................. 1-11
    Viewing xconfmanager Help .................................................................................... 1-14
    Setting Property Values and Propagating Your Changes......................................... 1-14
    Listing Property Information .................................................................................... 1-17
    Other xconfmanager Options .................................................................................. 1-17
  The Windchill PDM Home Page ................................................................................ 1-18
  Setting Up the WebEx Meeting Center ...................................................................... 1-19
    Accessing Meetings................................................................................................. 1-19
    WebEx Setup Properties ......................................................................................... 1-20
    WebEx User Name and E-mail Address .................................................................... 1-20
    Proxy Server for Connection to WebEx Meeting Center ....................................... 1-20
JAR Files and Security ........................................................................................................ 2-9

Administering External File Vaults ................................................................................. 3-1
Overview of Storing and Moving Data in Windchill....................................................... 3-2
Overview of External File Vaults .................................................................................. 3-2
The Central Cache Vault ............................................................................................... 3-4
Windchill Properties for File Vaulting .......................................................................... 3-6
Windchill External Storage Administrator .................................................................... 3-8
Adding and Updating Hosts ......................................................................................... 3-9
Adding and Updating Vaults ......................................................................................... 3-10
Deleting a Vault ............................................................................................................ 3-10
Adding and Updating Folders ...................................................................................... 3-10
Deleting a Folder .......................................................................................................... 3-11
Creating and Updating Mounts .................................................................................... 3-11
Maintaining Your Vaults .............................................................................................. 3-12
Creating a Vaulting Policy ............................................................................................ 3-13
Examining Existing Rules ............................................................................................ 3-13
Creating New Rules ..................................................................................................... 3-15
Modifying and Deleting Rules ..................................................................................... 3-16
Managing Revaulting .................................................................................................... 3-16
Examining Existing Revaulting Schedule Items.......................................................... 3-17
Viewing the Results of Revaulting .............................................................................. 3-18
Creating a Revaulting Schedule Item ........................................................................... 3-18
Updating a Schedule Item ............................................................................................ 3-20
Viewing a Schedule Item ............................................................................................. 3-20
Canceling a Schedule Item ........................................................................................... 3-21
Deleting a Schedule Item ............................................................................................. 3-21
Forcing Content to Vault ............................................................................................. 3-21
Changing from a Multiple Vault to a Single Vault Architecture .................................. 3-22
Changing the Location of Files in External Vaults ....................................................... 3-23
Moving a Server Using External Vaulting to a Different Host ..................................... 3-24

Administering Content Replication .............................................................................. 4-1
Overview ....................................................................................................................... 4-2
Setting Up Sites and Keys ............................................................................................ 4-4
Modifying the Local Site .............................................................................................. 4-5
Creating the Master Site ............................................................................................... 4-5
Creating a Replica Site .................................................................................................. 4-7
Creating and Placing Security Keys ............................................................................. 4-10
About Installing Two Types of Replica Sites ............................................................... 4-11
Installing a Full-scale Replica Site ................................................................................ 4-11
Administrative Conflicts of Common Administrative Objects ............................................ C-27

**Customizing Online Help** ........................................................................................................ D-1

  WebHelp Overview .................................................................................................................. D-2
  Customization Summary........................................................................................................ D-2
  Customizing Topic Content .................................................................................................. D-3
    Adding a New Topic ........................................................................................................ D-3
    Modifying an Existing Topic .......................................................................................... D-5
    Deleting an Existing Topic .............................................................................................. D-6
  Customizing Navigation Pane Content ............................................................................. D-6
    DHTML Table of Contents .............................................................................................. D-6
    DHTML Search ............................................................................................................... D-8
  Customizing Topic Appearance ....................................................................................... D-10

**Index**
### Change Record

The following table lists the major changes made in this guide for Windchill 7.0.

#### Table 1 Changes for Windchill 7.0

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1, <strong>Using the System Configurator</strong></td>
<td>Describes the redesigned interface of the System Configurator.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Using the xconfmanager Utility</strong></td>
<td>Describes the new utility for editing Windchill property files.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Setting Up the WebEx Meeting Center</strong></td>
<td>Improved description of the setup needed for Meeting Center.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Administering Organizations</strong></td>
<td>Describes the new features relating to organizations.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Administering Desktop Integration</strong></td>
<td>Describes the new feature that allows you to create and edit Windchill objects in Microsoft Office applications.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Running the Windchill ProjectLink Usage Report Utility</strong></td>
<td>Describes the utility that allows you to collect information about Windchill ProjectLink usage.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Administering User Preferences</strong></td>
<td>Updated to describe changes relating to containers.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Changing Authentication Text Between Servlet and Windchill Adapter</strong></td>
<td>Added to document how to change the secret text used in authenticating the servlet and Windchill adapter.</td>
</tr>
<tr>
<td>Chapter 1, <strong>Windchill Software Maintenance and Best Practices</strong></td>
<td>Introduces the service pack installation and best practices for maintenance.</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Chapter 4, Setting Up Sites and Keys</td>
<td>Presents complete documentation for installing a full-scale replica site and a lightweight replica site.</td>
</tr>
<tr>
<td>Chapter 4, Content Replication and Windchill Visualization Service</td>
<td>Explains how to insure the replication of viewables.</td>
</tr>
<tr>
<td>Chapter 4, Configuring Properties</td>
<td>Explains the <code>wt.fv.replicationFileSizeThreshold</code> property that controls the minimum size of files replicated.</td>
</tr>
<tr>
<td>Chapter 4, Replication and Compression</td>
<td>Explains compression in content replication.</td>
</tr>
<tr>
<td>Chapter 4, Improving Content Replication Performance</td>
<td>The Local Content Cache and Content Cache Server are now explained in this chapter rather than a separate chapter.</td>
</tr>
<tr>
<td>Chapter 5, Configuring External File Vaulting or Replication With FvLoader</td>
<td>Explains how to use the FvLoader utility to configure file vaulting and content replication. This is a new chapter.</td>
</tr>
<tr>
<td>Chapter 6, Controlling the Destinations of Imported Objects with Context Mapping Files</td>
<td>Explains context mapping file for use in import.</td>
</tr>
<tr>
<td>Chapter 6, Import and Export of EPMDocuments</td>
<td>Discusses prerequisites for importing and exporting EPMDocuments.</td>
</tr>
<tr>
<td>Chapter 6, Exporting with the Export User Interface</td>
<td>Explains the graphical user interface for Windchill Export.</td>
</tr>
<tr>
<td>Chapter 6, Importing with the Import User Interface</td>
<td>Explains the graphical user interface for Windchill Import and the backward compatibility for RatioDefinition and RatioValue.</td>
</tr>
<tr>
<td>Chapter 6, Additional Export and Import Actions</td>
<td>Explains Windchill Export and Import actions that do not appear in the graphical user interface.</td>
</tr>
<tr>
<td>Chapter 6, Access Control for Export and Import</td>
<td>Suggests access control rules for Windchill Import and Export.</td>
</tr>
<tr>
<td>Chapter 8, Queue Entry States</td>
<td>Adds Sereve state.</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Chapter 8, <strong>Background Queue Properties</strong></td>
<td>Describes updated properties.</td>
</tr>
<tr>
<td>Chapter 8, <strong>Maintaining the Queue</strong></td>
<td>Describes updated procedure.</td>
</tr>
<tr>
<td>Chapter 9, <strong>About Indexing</strong></td>
<td>Updated the example file shown in step 3.</td>
</tr>
<tr>
<td>Chapter 10, <strong>Customizing and Administrating Pro/ENGINEER Wildfire</strong></td>
<td>Presents customization and administration information, and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill PDM, Windchill PDMLink, and Windchill ProjectLink. This is a new chapter.</td>
</tr>
<tr>
<td>Appendix A, <strong>Windchill Runtime Environment</strong></td>
<td>Updates made for Windchill 7.0.</td>
</tr>
<tr>
<td>Appendix B, <strong>Windchill Considerations for Security Infrastructures</strong></td>
<td>Updates made for Windchill 7.0.</td>
</tr>
<tr>
<td>Appendix C, <strong>Import and Export Policies, Mapping Rules, and Conflict Messages</strong></td>
<td>Changes made to the appendix title.</td>
</tr>
<tr>
<td>Appendix C, <strong>Hierarchical Instance Based Attribute Definitions, Exporting, and Importing</strong></td>
<td>Explains preparation for importing hierarchical Instance Based Attribute definitions.</td>
</tr>
<tr>
<td>Appendix C, <strong>Conflict Messages</strong></td>
<td>Improves documentation of import conflicts.</td>
</tr>
<tr>
<td>Appendix D, <strong>Customizing Online Help</strong></td>
<td>Explains how to customize Windchill online help. This is a new appendix.</td>
</tr>
</tbody>
</table>
About This Guide

Intended Audience

The Windchill System Administrator’s Guide serves as a reference guide for Windchill system administrators for all Windchill solutions.

The following table illustrates the responsibilities and skills of system administrators:

<table>
<thead>
<tr>
<th>Responsibilities</th>
<th>System Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keeping the system running.</td>
<td>Keeping the system running.</td>
</tr>
<tr>
<td>Interfacing with other systems.</td>
<td>Interfacing with other systems.</td>
</tr>
<tr>
<td>Administering Windchill applications.</td>
<td>Administering Windchill applications.</td>
</tr>
<tr>
<td>Skills</td>
<td>Understanding Windchill server and client, HTML, HTTP, and Oracle.</td>
</tr>
</tbody>
</table>

Business and application administrators should refer to the Windchill Business Administrator’s Guide.

Overview

The Windchill System Administrator’s Guide describes responsibilities and roles of Windchill system administrators, providing conceptual and background information to help them understand the nature of system administration tasks.

Note: The Windchill Administrator's Guide, which was available in Windchill release 5.1 and earlier, has been reorganized for release 6.0 and later. Most of the information is now available in this guide or in the Windchill Business Administrator’s Guide. Information that covered vertical applications, such as Windchill Sourcing Factor, or third-party applications, such as the Workgroup Managers, has been moved to the individual guides for those applications.
Chapter Contents

The *Windchill System Administrator's Guide* is composed of the following chapters and appendixes:

This chapter, About This Guide, provides an overview of the guide and summarizes the contents of individual chapters.

Chapter 1, Administering Runtime Services, describes the System Configurator, which provides GUI-based access to the Windchill properties files and a mechanism for starting and stopping the Windchill server manager and all method servers. It describes the xconfman command line utility, which is used to edit property files. The chapter also describes other administrative responsibilities that are associated with the authentication process, backing up your system, and managing log files.

Chapter 2, Administering the Bootstrap Client, describes the bootstrap feature of Windchill, with information related to administrative responsibilities for creation and maintenance of JAR files when the bootstrap feature is enabled.

Chapter 3, Administering External File Vaults, describes the creation and maintenance of external file vaults.

Chapter 4, Configuring External File Vaulting or Replication With FvLoader, describes the FvLoader utility, which is a shortened version of File Vault Object Loader.

Chapter 5, Administering Content Replication, describes replica vaults, which store data that has been replicated from less rapidly accessible external vaults, or from the Windchill Oracle database.

Chapter 6, Windchill Import and Export, describes files and configuration properties for moving content and metadata to and from Windchill sites.

Chapter 7, Administering Content Holders and Content Objects, describes configuration properties for content handling, including procedures for adding and updating DataFormat objects and configuring your browser for upload and download operations.

Chapter 8, Administering Libraries, describes the definition, configuration, and bulk loading of index collections for use with Windchill search engines.

Chapter 9, Configuring and Administering Background Queues, describes the configuration of background queues, which are used to delay the completion of noncritical tasks and to speed up completion of time-critical tasks.

Chapter 10, Customizing and Administrating Pro/ENGINEER Wildfire, presents customization and administration information and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill PDM, Windchill PDMLink, and Windchill ProjectLink.

Appendix A, The Windchill Runtime Environment, describes Windchill's runtime architecture.
Appendix B, Windchill Considerations for Security Infrastructures, provides some basic Windchill information for dealing with firewalls, and other security issues.


Appendix D, Customizing Online Help, describes how to customize Windchill online help.

**Related Documentation**

The following documentation may also be helpful:

- *Windchill Business Administrator’s Guide*
- *Windchill Installation and Configuration Guide*
- *Windchill User's Guide*
- *Windchill Application Developer's Guide*
- *Windchill Customizer's Guide*
- *Windchill Adapter Guide*
- *Windchill Performance Tuning Guide*
- properties.html file

If these books are not installed on your system, see your installer.

**Technical Support**

Contact PTC Technical Support via the PTC Web site, phone, fax, or e-mail if you encounter problems using Windchill.

For complete details, refer to Contacting Technical Support in the *PTC Customer Service Guide* enclosed with your shipment. This guide can also be found under the Support Bulletins section of the PTC Web site at:

[http://www.ptc.com/support/index.htm](http://www.ptc.com/support/index.htm)

The PTC Web site also provides a search facility that allows you to locate Technical Support technical documentation of particular interest. To access this page, use the following link:

[http://www.ptc.com/support/support.htm](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 License Management using the instructions found in your *PTC Customer Service Guide* under Contacting License Management.
Documentation for PTC Products

PTC provides documentation in the following forms:

- Help topics
- PDF books

To view and print PDF books, you must have the Adobe Acrobat Reader installed.

All Windchill documentation is included on the CD for the application. In addition, books updated after release (for example, to support a hardware platform certification) are available from the Reference Documents section of the PTC Web site at the following URL:

http://www.ptc.com/cs/doc/reference/

Comments

PTC welcomes your suggestions and comments on its documentation—send comments to the following address:

documentation@ptc.com

Please include the name of the application and its release number with your comments. For online books, provide the book title.

Documentation Conventions

Windchill documentation uses the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold</td>
<td>Names of elements in the user interface, such as buttons, and menu paths. Required elements and keywords or characters in syntax formats.</td>
<td>Click <strong>OK</strong>. Select <strong>File &gt; Save</strong>. <code>create_&lt;tablename&gt;.sql</code></td>
</tr>
<tr>
<td>Italic</td>
<td>Variable and user-defined elements in syntax formats. Angle brackets (&lt; and &gt;) enclose individual elements.</td>
<td><code>create_&lt;tablename&gt;.sql</code></td>
</tr>
<tr>
<td>Monospace</td>
<td>Examples Messages</td>
<td><code>JavaGen &quot;wt.doc.*&quot; F true Processing completed</code></td>
</tr>
</tbody>
</table>
Third-Party Products

Examples in this guide referencing third-party products are intended for demonstration purposes only. For additional information about third-party products, contact individual product vendors.

Code Examples

Some code examples in this guide have been reformatted for presentation purposes and, therefore, may contain hidden editing characters (such as tabs and end-of-line characters) and extraneous spaces. If you cut and paste code from this manual, check for these characters and remove them before attempting to use the example in your application.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Caution" /></td>
<td>The <strong>Caution</strong> symbol indicates potentially unsafe situations which may result in minor injury, machine damage or downtime, or corruption or loss of software or data.</td>
<td>When you add a value to an enumerated type (for example, by adding a role in the RolesRB.java resource file), removing that value can result in a serious runtime error. Do not remove a role unless you are certain there is no reference to it within the system.</td>
</tr>
</tbody>
</table>
This chapter provides system administration information related to Windchill runtime services. For a diagram of the complete Windchill directory structure, see the *Windchill Application Developer’s Guide*.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windchill Configuration Properties</td>
<td>1-2</td>
</tr>
<tr>
<td>Using the System Configurator</td>
<td>1-6</td>
</tr>
<tr>
<td>Using the xconfmanager Utility</td>
<td>1-10</td>
</tr>
<tr>
<td>The Windchill PDM Home Page</td>
<td>1-18</td>
</tr>
<tr>
<td>Setting Up the WebEx Meeting Center</td>
<td>1-19</td>
</tr>
<tr>
<td>Administering Organizations</td>
<td>1-21</td>
</tr>
<tr>
<td>Administering Desktop Integration</td>
<td>1-25</td>
</tr>
<tr>
<td>Running the Windchill ProjectLink Usage Report Utility</td>
<td>1-26</td>
</tr>
<tr>
<td>Administering User Preferences</td>
<td>1-29</td>
</tr>
<tr>
<td>Ensuring Proper Backup and Recovery</td>
<td>1-34</td>
</tr>
<tr>
<td>Maintaining Log Files</td>
<td>1-34</td>
</tr>
<tr>
<td>Configuring Your Windchill Environment</td>
<td>1-35</td>
</tr>
<tr>
<td>Changing Authentication Text Between Servlet and Windchill Adapter</td>
<td>1-37</td>
</tr>
<tr>
<td>Administering the Authentication Process</td>
<td>1-38</td>
</tr>
<tr>
<td>Troubleshooting User Authentication</td>
<td>1-39</td>
</tr>
<tr>
<td>Windchill Scheduler</td>
<td>1-40</td>
</tr>
<tr>
<td>Windchill Software Maintenance and Best Practices</td>
<td>1-41</td>
</tr>
<tr>
<td>About the windchill Command</td>
<td>1-45</td>
</tr>
<tr>
<td>About the windchill shell</td>
<td>1-47</td>
</tr>
</tbody>
</table>
Windchill Configuration Properties

Windchill uses standard Java property files to dynamically configure many optional or site-dependent settings. The primary properties file, wt.properties, is located in the Windchill codebase directory, where it is available for downloading into clients. It contains properties that affect both client and server Java classes.

Properties available only to server-side classes are located in separate property files. For example, properties that control access to the database, including a database password, are located outside the codebase in the Windchill db directory, in the file named db.properties.

To manage site property settings, Windchill no longer recommends that you use a text editor to edit individual property files. Instead, all site changes to property files are maintained in the site.xconf file that is located in the directory where Windchill is installed. Use the following utilities to update the site.xconf file and then propagate the changes to property files:

- The xconfmanager is a command line utility that you can run to add, remove, or modify properties in any Windchill property file. The utility saves your changes in the site.xconf file and provides an option to generate updated property files using the updates in the site.xconf file.
- The System Configurator provides an interface for updating properties in the most common set of property files. From this application, authorized users can modify property values, save the changes to the site.xconf, and generate updated property files.

The changes made through either of these utilities are saved in the site.xconf file and propagated to respective property files. When you restart your Windchill system, the resulting changes are implemented.

Windchill creates the site.xconf file when Windchill Info*Engine is installed and adds all properties that are set during the installation of all Windchill solutions to the file. During the installation process, Windchill also creates the declarations.xconf file that contains a list of configuration references to PTC-supplied XCONF files that are used to specify the out-of-the-box default values for properties in many of the property files. Although not all property files are initially generated from XCONF files, you should always make changes to Windchill properties through either the xconfmanager or the System Configurator.

Note: By using these utilities, your site.xconf file will always contain your site-specific changes. By maintaining site-specific changes to properties in the site.xconf file, you can easily identify what changes were made and these changes can be maintained when you make updates to your Windchill solution.

As shown in the following diagram, making property changes through the utilities that Windchill provides always updates the site.xconf file. Then Windchill propagates the changes to properties files using the site.xconf file and the XCONF files that it maintains. In this diagram, the declarations.xconf file has references to
three sample internal XCONF files that then are used by Windchill to update referenced property files:

When you use the System Configurator to update properties, Windchill always propagates the changes to the corresponding property files. For additional information about using the System Configurator, see Using the System Configurator.

When using the xconfmanager, you must explicitly tell it to propagate XCONF file changes. To propagate changes using the xconfmanager, you must include the -p option. For information about using the xconfmanager, see Using the xconfmanager Utility.

Whenever you change a property setting using either the System Configurator or the xconfmanager, Windchill creates backup XCONF and property files of all files that are updated in the .xconf-backup directory where Windchill is installed. The file names for the back up files are modified to include a 3-digit number that identifies the backup file order. For example, the first three backup files created for the site.xconf file are named site.000.xconf, site.001.xconf, and site.002.xconf.

Windchill also maintains an internal cache containing the latest XCONF file information and maintains other internal XCONF files that it uses to determine what property files need to be updated. Do not manually modify these files.

The following sections provide information about the site.xconf file and the xconf.dtd file, which is used to validate all Windchill XCONF files.

**The site.xconf File Format and Contents**

The site.xconf file is an XML file that is formatted according to the xconf.dtd. The file is automatically updated to contain an element for every property setting change that is made through either the System Configurator or the xconfmanager.

The configuration elements included in the site.xconf file are as follows:
• Each Property element names a property, its target property file and the value of the property. The xconfmanager and System Configurator add this element to the site.xconf file when you set specific property values.

• Each ResetProperty element names a property and its target property file. The xconfmanager and System Configurator add this element to the site.xconf file when you reset properties to their default values.

• Each UndefineProperty element names a property and its target property file. The xconfmanager adds this element to the site.xconf file when you undefine properties so that their values are null.

Note: Although PTC recommends that you use either the System Configurator or the xconfmanager to modify the contents of the site.xconf file, some administrators may chose to modify the site.xconf file without using the Windchill tools. If you do manually modify the site.xconf file, be sure to format elements according to the xconf.dtd, which is documented in the next section. To propagate your changes to the affected property files, you must run the xconfmanager with the -p option and, to use the updated property files, you must restart your Windchill solution.

For examples of using the xconfmanager, see Using the xconfmanager Utility. For information about using the System Configurator, see Using the System Configurator.

The xconf.dtd File

Windchill uses the xconf.dtd to validate all elements in all XCONF files that it uses, including the site.xconf file. To ensure that this validation takes places for all XCONF files, no matter where they are located in the codebase and without access to the internet, the xconf.dtd is supplied using a JAR file and is not readily available through the Windchill directory structure.

The contents of the DTD file is as follows:

```xml
<!ENTITY % targetFile 'targetFile CDATA #IMPLIED'>
<!ENTITY % serviceProvider 'serviceProvider (wt|wtCustom|typeBased) #IMPLIED'>
<!ENTITY % name 'name CDATA #REQUIRED'>
<!ENTITY % context 'context CDATA "default"'>
<!ENTITY % overridable 'overridable (true|false) "true"'>
<!ENTITY % multivalued 'multivalued CDATA #IMPLIED'>

<!ELEMENT Configuration
(Property|Service|Resource|ConfigurationRef|ResetProperty|UndefineProperty|Propagat
ionAction)*)
<!ATTLIST Configuration
xmlns:xlink CDATA #IMPLIED
%targetFile;
%serviceProvider;
```
>
<!-- PTC to set "defaults", configurer to set "values" -->

<!ELEMENT Property (Documentation)?>
<!ATTLIST Property
  %name;
  default CDATA #IMPLIED
  defaultUnix CDATA #IMPLIED
  defaultWindows CDATA #IMPLIED
  value CDATA #IMPLIED
  %targetFile;
  %overridable;
>
<!-- %multivalued; this has been removed for now until the feature is fully
implemented -->

<!ELEMENT Documentation (Synopsis,Description,Deprecation?)>
<!ATTLIST Documentation
category CDATA #IMPLIED
key CDATA #IMPLIED
>
<!ELEMENT Synopsis (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Deprecation (#PCDATA)>

<!ELEMENT ResetProperty EMPTY>
<!ATTLIST ResetProperty
  %name;
  %targetFile;
>
<!ELEMENT UndefineProperty EMPTY>
<!ATTLIST UndefineProperty
  %name;
  %targetFile;
>
<!ELEMENT Service (Option)?>
<!ATTLIST Service
  %name;
  %context;
  %targetFile;
  %serviceProvider;
>
<!ELEMENT Resource (Option)?>
<!ATTLIST Resource
  %name;
  %context;
  %targetFile;
  %serviceProvider;
>
<!-- For Service/Options requires serviceClass and cardinality. For
Resource/Options requires resource attribute -->

<!ELEMENT Option EMPTY>
<!ATTLIST Option
  selector CDATA #IMPLIED
  requestor CDATA #REQUIRED
Using the System Configurator

Note: Access to the System Configurator is restricted to users identified as valid system administrators. Valid system administrators are specified in a comma-separated list in the property wt.sysadm.administrators, which is located in the wt.properties file. In the following example, only the user defined by wt.admin.defaultAdministratorName, user1 and user2 have permission to access the System Configurator:

wt.sysadm.administrators=${wt.admin.defaultAdministratorName},user1,user2

Authorized users can access the System Configurator application by clicking System Configurator from your Windchill solution. The link to the System Configurator in Windchill PDM is located on the System Administration home page. For Windchill PDMLink and Windchill ProjectLink, the link is on the Site Utilities page.

Use the System Configurator to accomplish the following:

- View information about the Windchill server manager and method servers, as well as stop and start the servers.
- Manage background queues.
- Modify system property files.
- View all available log files and e-mail snapshots of log files.

For detailed procedures and explanations of System Configurator fields, click the Help button on any System Configurator page. Managing background queues is also described in the Configuring and Administering Background Queues chapter.
Starting and Stopping the Windchill System

When your server manager is up and you access the System Configurator, the Server Status page opens by default. Or, if you have another page open, you can click the Server Status tab to display the Server Status page.

To start or stop the server manager or a method server, click the icon in the Actions column of the associated row on the Server Status page. If the server manager or method server is running, clicking the icon stops it. If the server manager or method server is stopped, clicking the icon starts it.

When your server manager is down, you can access the System Configurator Server Status page by entering the following URL in your Web browser:

http://<host>/<webApp>/wtcore/jsp/wt/sysadm/SystemConfigurator.jsp

where <host> is the host system (and port, if you are not using the default port of 80) where Windchill is installed and <webApp> is the Web application name defined for Windchill (the default is Windchill).

From the Server Status page, you can then start or stop the server manager, start or stop individual method servers, and check the status of all servers.

Modifying System Properties

Click the Edit Properties tab to display the links to individual property files. The Edit Properties page displays links to the Windchill property files that have been established as property names that end in .properties in the wt.properties file and are available from the location specified in the property. The values for these properties consist of the fully qualified property file path names where the files reside. For example, if the pom property file is defined in wt.properties as:

wt.pom.properties=$(wt.home)$(dir.sep)db$(dir.sep)db.properties

and the db.properties file does reside in the codebase directory where your Windchill solution is installed, then the pom link appears on the Edit Properties page. If the property did not end in .properties or the db.properties file did not reside in the specified directory, then the pom link would not appear on the Edit Properties page.

To add properties and corresponding values to a property file, use the xconfmanager utility. For information about the xconfmanager utility, see Using the xconfmanager Utility.

Use the following procedure to modify system properties from the System Configurator:

1. On the Edit Properties page, click the link to property file that you want to modify.

   The Search For field appears.
2. Enter either the property name of a specific property you want to modify, a partial name with asterisk (*) wild card characters to display a subset of the properties, or just the * to display all properties in the file, and then click **Search**.

The table of properties refreshes with the search results.

3. To change the view displayed in the table, you can click **Advanced** or **Basic**, from the **Current View** drop-down list. The value selected is set in the wt.sysadm.advancedView property and determines how the property names are displayed:

- If **Advanced** is selected, the value is true (default), and property names are displayed as programmatic names.
- If **Basic** is selected, the value is false, and property names are displayed as short descriptions.

In the example that follows, **Advanced** is selected, which causes properties to be displayed as programmatic names.

You can edit the **Value** text fields or click **true** or **false** to change the property. The **Default** check box, on the right, is checked if the value displayed is the default value. A blank check box indicates that the **Value** text field comes from the properties file or that the value has been changed from the default value. Select the check box to return to the default value.

If you want to cancel changes you made to the properties value, click **Reset**, at the bottom of the page.

When you are satisfied with your changes, click **OK** to save. Your changes are written to the site.xconf file and affected property files are regenerated using your changes. Backup copies of XCONF files are saved in the .xconf-backup directory where Windchill is installed. Backup copies of the property files are also saved in the .xconf-backup directory. Examples of backup copy property file names are tools.000.properties, tools.001.properties, db.000.properties, wt.000.properties,
and wt.001.properties. The properties with values selected to be the default are excluded from the changed property file.

**Note:** The regenerated property files are used to set system properties when the system is next restarted.

### Working with Log Files

Click the **Edit Logs** tab to display the links to individual log files. The **Edit Logs** page links are derived from property values in the wt.properties file that have *log* in the property name and end in .log. When you select a file to view or edit it, a copy of the log file opens. Changes made do not affect the actual log file, which is maintained in the host file system. Changes to the actual log file that occur after you have selected it are not reflected on the displayed copy.

The following is an example of the Method Server log file that shows the last 20 log entries:

```
Tue 4/20/2003 10:47:10: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
Tue 4/20/2003 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 100
```
E-mailing Log Files

At the bottom of the Edit Logs page, you can use the following fields to e-mail the copy of the current log file to others:

Follow the procedure below to e-mail the log file copy:

1. Enter e-mail addresses of your recipients in the E-mail To and Copy E-mail To text boxes. If you are sending to multiple addresses, separate addresses with a space.
2. Type a subject in the Subject text box.
3. Click Send at the bottom of the page.

Using the xconfmanager Utility

The xconfmanager is a command line utility that you can run to add, remove, or modify properties in any Windchill property file except the following:

- associationRegistry.properties
- classRegistry.properties
- descendentRegistry.properties
- modelRegistry.properties

These property files are maintained using the Information Modeler utility and should not be modified outside of this utility.

The xconfmanager utility saves your changes in the site.xconf file and provides an option to generate updated property files using the updates in the site.xconf file. The site.xconf file contains the changes made to Windchill property files starting with the installation and continuing with each use of the xconfmanager utility and the System Configurator. The site.xconf file is located in the directory where Windchill is installed.

Anyone with write access to the XCONF and property files under Windchill installation directory can successfully run the xconfmanager utility.
The following sections describe how to enter the xconfmanager command and how to set property values and list property information using the command. The last section describes the other xconfmanager options that may be useful when running your Windchill solution.

The xconfmanager utility is located in the bin directory where your Windchill solution is installed. For example, if Windchill solution is installed in the C:\ptc\Windchill directory, then the utility is in the C:\ptc\Windchill\bin directory.

Before executing the xconfmanager command, set up your environment by using the windchill shell. To use the shell, enter the following on the command line:

```
windchill shell
```

Then from the new window that opens, you can enter the xconfmanager command, as described in the next section.

**xconfmanager Command Syntax**

The syntax of xconfmanager command that administrators should use is as follows:

```
xconfmanager {-FhuwV} {-r <product_root>} {-s <property_pair> {-t <property_file>}}
{-reset <property_names>} {--undefined <property_names>} {-d <property_names>} {-p}
```

The brackets ({}) in the syntax indicate optional parameters and indicate parameters that you specify together. The syntax includes only the short version of each parameter name. Parameter names are case-sensitive; enter the names using the case shown in the syntax and the following table.

In the following table, all parameter names are listed in alphabetical order with corresponding parameter descriptions:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d or --describe</td>
<td>Lists the values that are currently being set and the corresponding XCONF file where each value is set for the specified properties. &lt;br&gt; &lt;property_names&gt; is a comma-separated list of property names. &lt;br&gt; This option executes after all parameter setting options and the -p option have executed.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>-F</code> or <code>--force</code></td>
<td>Forces the propagator to ignore its cache of XCONF-to-properties file dependencies and ignore the timestamp comparison it usually does to determine which property files need to be updated. Using this option propagates all site-specific changes to property files. Use this option in place of <code>-p</code> if you suspect that the cache is out of date.</td>
</tr>
<tr>
<td><code>-h</code> or <code>--help</code></td>
<td>Displays the help for the xconfmanager command.</td>
</tr>
<tr>
<td><code>-p</code> or <code>--propagate</code></td>
<td>Propagates all changes that have been made to XCONF files into the property files that are being used. This option always executes after any options that set properties. This execution order ensures that the newly set properties are included in the propagation. Updated property files are accessed when the Windchill solution is restarted.</td>
</tr>
<tr>
<td><code>--reset</code></td>
<td>Resets the site specific value of a property or set of properties to the declared default values. <code>&lt;property_names&gt;</code> is a comma-separated list of property names.</td>
</tr>
<tr>
<td><code>-r</code> or <code>--productroot</code></td>
<td>The root directory from which all relative paths are based for XCONF references specified in the declarations.xconf file and target file paths specified in the <code>-t</code> parameter. The default root directory is the bin directory where the Windchill solution is installed.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| --reset      | Resets the site specific value of a property or set of properties to the declared default values.  
  <property_names> is a comma-separated list of property names. |
| -r or --productroot | The root directory from which all relative paths are based for XCONF references specified in the declarations.xconf file and target file paths specified in the -t parameter. |

The default root directory is the bin directory where the Windchill solution is installed.
Note: The xconfmanager executes the -s, --reset, and --undefine parameters in the order that they are specified in the command. This means that if the same property is set in multiple parameters, the last setting is used.

The xconfmanager always executes the -p parameter after all specified -s, --reset, and --undefine parameters. This is done so that all parameter settings are included in the propagation.

The xconfmanager always executes the -d parameter after all specified -s, --reset, --undefine, and -p parameters. This is done so that the descriptions returned include all of the parameter settings made on the command.

**Viewing xconfmanager Help**

Use the -h or --help parameter on the xconfmanager command to list the xconfmanager command syntax and provides a description of each parameter.

**Setting Property Values and Propagating Your Changes**

The xconfmanager utility provides options that allow you to manage the properties in a Windchill property file as follows. You can:

- Set a property value to specific value by using the -s and -t options.
- Set a property value to the declared default value by using the --reset option.
- Set a property value to null (instead of an empty string) using the --undefine option.
- Propagate the site changes stored in the site.xconf file to all affected property files using the -p option.

**Setting Specific Property Values**

On the xconfmanager command, use the -s parameter to set a specific property value and the -t parameter to set the target property file for the property setting. In

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--undefine</td>
<td>Resets the specified properties such that their values will be null (instead of an empty string) when read through a java.util.Properties instance. &lt;property_names&gt; is a comma-separated list of property names.</td>
</tr>
<tr>
<td>-v</td>
<td>Turns on verbose console output, which shows full exception stack traces.</td>
</tr>
<tr>
<td>-V</td>
<td>Turns on debug verbose console output. This option shows full exception stack traces and additional information.</td>
</tr>
</tbody>
</table>
a given xconfmanager command, you can specify multiple -s parameters. However, all properties specified must reside in the same target property file; there can only be one -t parameter.

The property values you set must conform with the specification for java.util.Properties. The following guidelines will help ensure that you set properties correctly:

- Use forward slashes (/) in file paths so that the platform designation is not an issue.
- To specify a property whose value contains characters that might be interpreted by your shell, escape them using the appropriate technique for the shell you are using.

For example, on a Windows system you can include spaces in a value by enclosing the argument with doubles quotes. For example, use the following:

-s "wt.inf.container.SiteOrganization.name=ACME Corporation"

On a UNIX system, you can use doubles quotes or you can escape the space character with a backslash. For example, use the following:

-s wt.inf.container.SiteOrganization.name=ACME\ Corporation"

On UNIX, dollar signs are usually interpreted by shells as variable prefixes. To set a property value that has a dollar symbol in it, use single quotes around the argument so that it is not interpreted by the shell or use backslash to escape the dollar symbols. For example, use either of the following:

-s 'wt.homepage.jsp=$(wt.server.codebase)/wtcore/jsp/wt/portal/index.jsp'

or

-s wt.homepage.jsp=\$(wt.server.codebase)/wtcore/jsp/wt/portal/index.jsp

Other than escaping arguments so that the command line shell does not misinterpret them, the values should not need to be escaped any further to be compatible with XML or property file syntaxes. The xconfmanager escapes property names and values automatically if necessary.

The following xconfmanager command used on a Windows system sets the wt.properties property file wt.temp property to the WCtemp directory that is under the Windchill installation directory [as defined by $(wt.home)]:

xconfmanager -s wt.temp=$(wt.home)/WCtemp -t wt.properties -p

Assuming that the command was executed from the Windows C:\pte\Windchill\bin directory, then the resulting output is:

Default product root=C:\pte\Windchill\bin\..
java -jar "C:\pte\Windchill\bin\.\codebase\WEB-INF\lib\install.jar" -r "C:\pte\Windchill\bin\" -s wt.temp=$(wt.home)/WCtemp -t wt.properties -p
Propagating xconf data to target files...

The xconfmanager creates a backup of the current site.xconf file, adds the property element for wt.temp to the site.xconf file (replacing any existing property setting that had been in the site.xconf file), and then propagates the change to wt.properties.

Restoring a Property Value to Its Default Value

Use the --reset parameter on the xconfmanager command to restore one or more properties to their default values. To specify multiple properties in the parameter, separate the properties using a comma.

The following xconfmanager command resets the wt.temp property:

```
xconfmanager --reset wt.temp -p
```

Assuming that the command was executed from the Windows C:\\ptc\\Windchill\\bin directory, then the resulting output is:

```
Default product root=C:\\ptc\\Windchill\\bin\\..
java -jar
"C:\\ptc\\Windchill\\bin\\..\\codebase\\WEB-INF\\lib\\install.jar"
-r "C:\\ptc\\Windchill\\bin\\.." --reset wt.temp -p
Propagating xconf data to target files...
```

The xconfmanager creates a backup of the current site.xconf file, removes any existing property settings for the specified properties that had been in the site.xconf file, adds a ResetProperty element for each property that was specified (in this case, only wt.temp), and then propagates the change to property files that have the specified properties (in this case, only wt.properties).

Setting a Property Value to the Null Value

Use the --undefine parameter on the xconfmanager command to set one or more properties to null values. To specify multiple properties in the parameter, separate the properties using a comma.

The following xconfmanager command sets the wt.services.service.1160 property to null (which disables the service):

```
xconfmanager --undefine wt.services.service.1160 -p
```

Assuming that the command was executed from the Windows C:\\ptc\\Windchill\\bin directory, then the resulting output is:

```
Default product root=C:\\ptc\\Windchill\\bin\\..
java -jar
"C:\\ptc\\Windchill\\bin\\..\\codebase\\WEB-INF\\lib\\install.jar"
-r "C:\\ptc\\Windchill\\bin\\.." --undefine
wt.services.service.1160 -p
Propagating xconf data to target files...
```
The xconfmanager creates a backup of the current site.xconf file, removes any existing property settings for the specified properties that had been in the site.xconf file, adds an UndefineProperty element for each property that was specified (in this case, only wt.services.service.1160), and then propagates the change to property files that have the specified properties (in this case, only wt.properties).

**Listing Property Information**

Use the -d parameter on the xconfmanager command to list information about one or more properties. To specify multiple properties in the parameter, separate the properties using a comma. The resulting output includes the current value of each property and the location of the files where each property is set.

The following xconfmanager command lists the information for the wt.home property:

```
xconfmanager -d wt.home
```

Assuming that the command was executed from the Windows C:\ptc\Windchill\bin directory, then the resulting output is:

```
Default product root=C:\ptc\Windchill\bin\..
java -jar "C:\ptc\Windchill\bin\..\codebase\WEB-INF\lib\install.jar"
-r "C:\ptc\Windchill\bin\.." -d wt.home
WARNING: Propagation of xconfs to properties was not requested. To ensure your properties are up to date, re-run with the -p option.

Property information for 'wt.home':
Values:
- C:\Windchill
  Locations:
  - file:/C:/Windchill/site.xconf, line 9
  - file:/C:/Windchill/codebase/wt.properties.xconf, line 17
```

**Other xconfmanager Options**

The xconfmanager utility provides additional options that can be useful when setting up a Windchill cluster, performing customizations, or analyzing system problems:

- To specify the root directory that is not the default root directory, use -r. The default root directory is the bin directory under the Windchill installation directory.

  The xconfmanager utility uses the root directory when relative paths for XCONG references and target file paths are used.

- To force propagation of all property values listed in the site.xconf, use -F instead of using -p. The -F option forces the propagation regardless of the analysis that is done to determine which files are already up-to-date.
The Windchill PDM Home Page

Following installation, you can open the Windchill PDM application home page. The Server Manager and Method Server must be running. Java Server Pages (JSPs) are used to display the Windchill application home page. Therefore, you must have a JSP engine set up to display the page.

The home page appears something like the following:

The first time you access the Windchill home page, you can select one of the links listed under Available Homes to make that page your personal home page. The next time you access Windchill, it will open to that page. If, at any time, you want to change your personal home page, click Options or Site Map, and then click the link to the page you want as your new personal home page. (The Options and Site Map links appear near the top and at the bottom of the menu bar on your personal home page.)
Setting Up the WebEx Meeting Center

A meeting is a scheduled block of time noted on the Meetings page informing you when and where a meeting is being held, the meeting creator, and who else has been invited. You can create or participate in standard meetings, web-based meetings, or ProductView peer-to-peer collaborative sessions. All meetings contain the same information; however, a web-based meeting is powered by WebEx, and a visualization collaborative session is conducted through ProductView.

The following sections describe how to access meetings, set up the WebEx properties that are required, set up a WebEx user name and e-mail address, and set up a proxy server for Meeting Center.

Accessing Meetings

You can access meetings from Windchill solutions as follows:

- For Windchill PDM, the Meeting Center icon appears in the icon bar at the top of the Windchill home page. This icon is identified below:

  ![Meeting Center icon](image)

  When you click the Meeting Center icon, the Windchill meeting page appears. From this page, you can see existing meetings, create meetings, and cancel meetings.

- For Windchill ProjectLink, clicking the Meetings link on the Home tab displays the My Meetings table. Clicking the Meetings link on the Project tab displays the Meetings table. From either of these tables, you can see existing meetings, create meetings, and cancel meetings.

- For Windchill PDMLink, clicking the Meetings link on the Home tab displays the My Meetings table. From this table, you can see existing meetings, create meetings, and cancel meetings.

Note: In order to execute a web-based meeting, you must have an active license established through WebEx Communications, Inc. Refer to [www.webex.com](http://www.webex.com) for more information.
WebEx Setup Properties

To enable web-based meetings in the Meeting Center and to connect to the WebEx server, add the following properties to the wt.properties file using the xconfmanager.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.meeting.centerUrl</td>
<td>Specifies the URL for the WebEx server. For example, <a href="http://ptc.webex.com">http://ptc.webex.com</a></td>
</tr>
<tr>
<td>wt.meeting.partnerId</td>
<td>Specifies the partner ID for the WebEx server.</td>
</tr>
</tbody>
</table>

For example, execute the following command from a windchill shell:

```
xconfmanager -s wt.meeting.centerUrl=<url_value>
-s wt.meeting.partnerId=<partner_id>
-t <Windchill>/codebase/wt.properties -p
```

Where `<Windchill>` is the location where Windchill is installed.

To diagnose problems in setting up the connection to the WebEx server, set wt.meeting.verbose to TRUE in the wt.properties file.

WebEx User Name and E-mail Address

When you host a meeting, the WebEx account used is based on your Windchill user name. The account name is `webex_<user name>`.

Every WebEx account must have a unique e-mail address associated with it. Two accounts cannot use the same e-mail address.

Proxy Server for Connection to WebEx Meeting Center

The WebEx server is always located outside the corporate firewall, and Windchill servers are usually located inside the firewall. Your site may require the use of a proxy server for HTTP connections through the firewall. To make it possible for the Windchill server to connect to the Webex server, the wt.manager.cmd.MethodServer entry in wt.properties must be modified to look like the following example:

```
wt.manager.cmd.MethodServer=\ 
  cmd.exe /C start "MethodServer" /MIN \ 
  "$\{wt.java.cmd\}" -classpath "$\{wt.java.classpath\}" \ 
  -Djava.protocol.handler.pkgs=HTTPClient \ 
  -Dhttp.proxyHost=proxy.mycompany.com \ 
  -Dhttp.proxyPort=8080 \ 
  -Dhttp.nonProxyHosts=.mycompany.com \ 
  -Xms32m -Xmx64m -Xnoclassgc -noverify 
wt.method.MethodServerMain
```
The bold-faced entries are the required changes.

In this example the proxy server is located on host proxy.mycompany.com and is listening on port 8080. This proxy server is to be used for all HTTP connections, except those with host names ending in .mycompany.com.

**Administering Organizations**

Windchill solutions use organization objects (WTOrganization) and organization containers when administering organization information.

The development of products and the subsequent management of product information throughout their entire life cycle is truly a collaborative process involving a number of organizations, including suppliers, contract manufacturers, and design partners. The Windchill solutions use organization containers as follows:

- To define your digital product value-chain.
- To define the organizational members and the roles they play within your business processes.
- To define data ownership responsibilities.
- To define the level of engagement that organizations have within your system and business processes.

All Windchill solutions, when configured, contain a host organization. This organization represents your enterprise and is associated with an organization container. The users in the host organization either author product information or in some way are consumers of this information.

As you define your digital product value-chain, additional organization objects can be created to represent your suppliers, contract manufacturers, or design partners. Each of these organizations may have a unique identifier such as a CAGE or DUNS code. Optionally, the various organizations within your digital product value-chain can have their own containers for which you can delegate product, library, or project authority. Product information created in the context of each organization through their products, libraries, or projects would be owned by each respective organization and possibly identified according to their identification policies. Product information owned by external organizations that do not have their own containers can also be supported using the administrative capabilities defined in this section.

Windchill PDM has only one organization container (and corresponding organization object) that is created during installation. Additional organization objects can be created using the Principal Administrator, but no additional organization containers can be created.

In Windchill PDMLink and Windchill ProjectLink, organization containers (and corresponding organization objects) can be created for each of the business organizations and or business units that are collaborating together through the
Windchill solutions. Each organization inherits templates (document, workflow, and life cycle templates) and groups defined in the parent site container and then defines its own organization-specific templates, groups, types and roles. A separate group of administrators is associated with each organization to manage the organization templates, groups and policies. The organization administrator also can control who is allowed to create application containers (products, libraries and projects) within their organization.

Windchill PDMLink and Windchill ProjectLink provide client user interfaces for doing most activities that are related to administering organizations. These interfaces and how to use the Principal Administrator are described in the following chapters of the *Windchill Business Administrator’s Guide*:

- Administering the Site
- Administering Organizations
- Administering Principals

In addition to the activities described in these chapters of the *Windchill Business Administrator’s Guide*, system administrators can administer organizations as described in the following sections.

**Restricted Organizations**

In Windchill PDMLink and Windchill ProjectLink, when you create the organization container, you can specify whether the users in the organization context that you are creating are restricted from seeing users in other organizations. This is done through the **Allow entire user and group directory selection** check box that is available when you are creating or updating an organization container. When you do not allow users to see other users and groups, the organization is a restricted organization.

Out of the box, users are associated with the domain that has the same name as the organization. For example, if the organization is named Org1, then the domain is named Org1. When the organization is not associated with an organization container, then the domain is in the Site context and its parent domain is the User domain. When an organization container is created, the organization domain for it’s users is moved from the Site context to the organization context. The parent domain remains the User domain in the Site context.

When creating organization objects using the Principal Administrator, the object created is considered a restricted organization if you use the default Windchill domain and have not modified access control rules for domain and ancestor domains. Depending on the modifications you have made, you may need to make changes to restrict the users in organizations that are not associated with an organization container. For example, you can do the following:

- Use the Policy Administrator to reparent the organization domain used for the members of the organization from the User domain to the / (root) domain. This ensures that no access control rules are inherited from the User domain.
If you haven’t modified the access control rules for the User domain, this step may not be needed.

- Use the Principal Administrator to update the Unrestricted Organizations group that is in the User domain to remove the organization (if it is present).

**Note:** When creating an organization object through Principal Administrator, the organization is not automatically added to the Unrestricted Organizations.

### Internal Organizations

When a Windchill solution is installed and an organization container is created, then this organization automatically owns the parts and documents that are created under the organization context. In Windchill PDMLink and Windchill ProjectLink, you can also create additional organization contexts under which parts and documents can be automatically owned.

To change the out-of-the-box functionality so that a user who creates a part or document can specify which organization owns the part or document, you must do the following things:

- Create or update organization objects so that they can be used as internal organizations as described in the next section.

An internal organization is identified by an internationally coded number that is assigned when the organization registers with a specific site. For site registration information, see the [Registration Authority](#).

**Note:** You cannot create an internal organization for a specific organization if the organization is not registered.

- Set properties in the `wt.properties` file, as described in [Setting Internal Organization Properties](#).

- Configure the container where the parts or documents will reside so that the user enters the part or document number (rather than having the numbers auto-generated). How to turn off autonumbering is described in the Object Initialization Rules help that is accessible from the Object Initialization Rules Administrator. For information about the Object Initialization Rules Administrator, the [Windchill Business Administrator’s Guide](#).

### Creating Internal Organizations

Use the Principal Administrator to create a new organization object or update an existing organization object to include the following attributes on the object:

- **The Organization ID Type** indicates the type of organization identifier that is specified for the Organization ID. Select the type from the drop-down list. By default, the drop-down list contains CAGE, DUNS, and ISO6523.
If the drop-down list does not contain the type required for your company organization, modify the list by updating the wt.org.organizationTypes property that is located in the wt.properties file.

- The **Organization ID** specifies the globally unique organization identifier under which the organization is registered.

- The **Windchill Domain** identifies the administrative area where the organization object resides. The domain selected must have access control rules set for the WTOrganization object type so that the users who create part and documents have read access to the organization objects that you want to use as internal organizations. For information on how to set up a domain for this use, see the next section, Setting Up Domains for Use with Internal Organizations.

If you enter a value for **Organization ID** in an organization object, then Windchill combines the organization ID type number and ID and stores the resulting value in the organizationIdentifier attribute of the organization directory entry. The format of the attribute is:

```
<ICD_number>$<org_ID>
```

where:

- `<ICD_number>` is the international code designator number assigned to the organization ID type. For example, the CAGE ICD number is 0141. For a list of ICD numbers, see the Registration Authority.

- `<org_ID>` is the organization identification number assigned when the organization was registered.

**Setting Up Domains for Use with Internal Organizations**

To allow users to identify the ownership of parts and documents that are created in the solution with external vendors and suppliers, the organization objects created for vendors and suppliers must be in a domain that allows the users read access to the organization objects.

A simple approach to setting this up is the following:

1. Create a domain that will be used specifically for this purpose. For example create the Vendors domain using the Policy Administrator from the Site container.

2. Create the access control rule in the Vendors domain that grants READ permission on WTOrganization objects to ALL users.

3. When creating the organization objects that represent external vendors and suppliers using the Principal Administrator, select to the Vendors domain for the **Windchill Domain** field.
### Setting Internal Organization Properties

Use the xconfmanager utility to set the following properties in the wt.properties file:

- `wt.org.OrganizationOwned.displayOrganization`
- `wt.org.InternalOrganization`

The `wt.org.OrganizationOwned.displayOrganization` property value can be set to either True or False and the default value is False. To turn on the display of internal organizations when users create and update parts and documents, set this property to True.

The `wt.org.InternalOrganization` property value is a comma-separated list of internal organizations, where each internal organization is represented by its `organizationIdentifier` attribute of its organization directory entry. The details on the format of this attribute are described in the preceding section. If you are unsure of the value to use for an organization, you can view the `organizationIdentifier` attribute value of the organization from the Aphelion LDAP browser. Using this browser is described in the *Windchill Info*Engine Installation and Configuration Guide.

For details on how to set properties, see [Using the xconfmanager Utility](#).

### Administering Desktop Integration

The Windchill Desktop Integration feature allows you to create and edit Windchill objects in Microsoft Office applications. As an administrator, you must edit the `wtSearch.xml` file in order to configure the Desktop Integration Enterprise Search settings.

The `wtSearch.xml` file is stored in the following directory where your Windchill solution is installed:

```
codebase\com\ptc\windchill\enterprise\nativeapp\msoi\client\xml
```

Changes to the file must be made on the server and these changes are automatically propagated to clients the next time a user connects to the server.

### Enterprise Search Settings

If your site does not use Enterprise Search in Windchill PDMLink, you must remove the keyword field element from the `wtSearch.xml` file. This removes the **Keyword** field from the Desktop Integration search (the **Keyword** field is used only for Enterprise Search functions).

To make this change, locate the search form element, which begins with the following tag:

```
<form key="search">
```
Within the search form element, delete the **keyword** field key element. This element is in bold type in the following sample search form:

```xml
<form key="search">
  <searchfields key="criteria_string">
    <field key="number"></field>
    <field key="name"></field>
    <field key="keyword"></field>
  </searchfields>
  <resultfields>
    <field key="name"></field>
    <field key="number">0:asc</field>
    <field key="versionDisplayIdentifier"></field>
    <field key="lifeCycleState"></field>
    <field key="checkedOutBy"></field>
    <field key="modifyTimestamp"></field>
    <field key="obid" hidden="true"></field>
  </resultfields>
</form>
```

The `<resultfields>` element determines which fields appear in the search results, as well as the sort order of the fields. You can add and remove fields as necessary. To change the sort order, add numbers to the appropriate field key elements, followed by "asc" or "desc" to indicate whether the order is ascending or descending. For example, in the preceding sample, the `<field key="number">` element is assigned the number 0 so it will appear first in the list of fields. The "asc" value indicates that the results will be sorted by number in ascending order.

**Running the Windchill ProjectLink Usage Report Utility**

The Windchill ProjectLink Usage Report utility allows you to collect information about Windchill ProjectLink usage. For example, you can use this utility to collect usage data for billing purposes.

**Creating Usage Reports**

To create Windchill ProjectLink usage reports, use the following procedure:

1. Type the following on a command line:

   ```
   windchill com.ptc.netmarkets.report.ProjectAuditingUI
   ```
2. Log on at the authentication prompt.

3. Enter the name of the organization for which you are searching. The **Organization** field defaults to ALL. ALL returns all organizations to which you have access.

4. Set the dates for which you want to report. The end date defaults to today’s date and the start date defaults to one month prior to the end date.

5. Click **Search**.

6. Only those organizations for which you have access display. From the list at the top right, select the organization(s) for which you want to report.

7. Click **Generate Report**.

8. Click **Print to File**. A dialog box displays for choice of output: **text** or **html**. The best output is **html**.
Sample HTML Report

The following is a sample HTML report:

Windchill ProjectLink Project Summary Report

Report Period of: 2003-09-24 19:00:00.0 to 2003-10-24 19:00:00.0
Report run at 2003-10-24 13:40:20.031 by wcadmin with email WindchillAdministrator@ptc.com
Warning: Report was run before the report period expired.
Logins between 2003-10-24 13:40:20.031 and 2003-10-24 19:00:00.0 will not appear on this report.

<table>
<thead>
<tr>
<th>Organization</th>
<th>User ID</th>
<th>Name</th>
<th>EMail</th>
<th>Last Access</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptc</td>
<td>smorris</td>
<td>Sarah Morris</td>
<td><a href="mailto:smorris@ptc.com">smorris@ptc.com</a></td>
<td>2003-09-28 10:43:53.0</td>
<td>Test Project 2</td>
</tr>
<tr>
<td>ptc</td>
<td>Hernandez</td>
<td>Nick Hernandez</td>
<td><a href="mailto:Hernandez@ptc.com">Hernandez@ptc.com</a></td>
<td>2003-10-22 16:17:52.0</td>
<td>Test Project 1</td>
</tr>
<tr>
<td>ptc</td>
<td>amyers</td>
<td>Austin Meyers</td>
<td><a href="mailto:amyers@ptc.com">amyers@ptc.com</a></td>
<td>2003-10-22 13:25:43.0</td>
<td>Test Project 1</td>
</tr>
<tr>
<td>ptc</td>
<td>dvalder</td>
<td>Dave Valder</td>
<td><a href="mailto:dvalder@ptc.com">dvalder@ptc.com</a></td>
<td>2003-10-18 11:21:56.0</td>
<td>Test Project 2</td>
</tr>
<tr>
<td>ptc</td>
<td>jpark</td>
<td>Jane Park</td>
<td><a href="mailto:jpark@ptc.com">jpark@ptc.com</a></td>
<td>2003-10-12 15:24:52.0</td>
<td>Test Project 2</td>
</tr>
<tr>
<td>ptc</td>
<td>jwilson</td>
<td>James Wilson</td>
<td><a href="mailto:jwilson@ptc.com">jwilson@ptc.com</a></td>
<td>2003-10-21 14:18:59.0</td>
<td>Test Project 2</td>
</tr>
</tbody>
</table>

Total Billing for ptc: 6 Users

Overall Billing: 6 Users

The time shown in the report is Greenwich Mean Time (GMT).
Administering User Preferences

The following sections describe how to define preference scopes and manage user preferences.

Defining Preference Scopes

Windchill administrators with appropriate permissions can change a wide variety of user preferences. Out of the box, the four scopes defined are:

- The 'default' which provides values for preferences that do not have any values defined in the other scopes. These values cannot be changed at runtime.
- The container level scopes, which are identified by the container name, are used to set preferences for all of the containers that are defined.
- A division level scope called Windchill Enterprise, which administrators can use to define preferences for all users.
- The user (bottom level) scope which individual users can use to set their preferences.

Adding additional scopes involves two procedures

- Creating a preference structure, which is the data structure used to store preferences.
- Creating a preference hierarchy, which is the order that defines which preferences takes precedence over others.

Creating a Preference Structure

A preference consists of the following three attributes:

- **Parent node**, or root node if it is at the top of the hierarchy.
- **Preference node**, usually represents a group of similar preferences.
- **Preference key**, a string name for the preference.

Together these attributes form a unique key structure of parent/node/key. This unique key structure is known as the **fully qualified preference key**. To separate individual user and group preferences for the same fully qualified preference key, a context is applied to the preference.

The context consists of the following elements:

- **Macro**, a constant that defines the type of context (see below).
- **Descriptor**, the text that defines the name of the context.

These elements are separated by colons to form the preference context.
The fully qualified preference key is combined with a context to form a unique row in the database table. This makes it possible for users and other divisions to have separate preferences.

Preference Macros

The wt.prefs.WTPreferences class defines the following types of preference context macros:

- USER_CONTEXT - the context for individual users.
- DEFAULT_CONTEXT - the context for the system default (shipping) values.
- CONTAINER_CONTEXT - a context used in the container hierarchy.
- CONTAINER_POLICY_CONTEXT - a container context that is enforced as a policy.
- DIVISION_CONTEXT - the context used for any scopes defined in addition to the default, container, and user scopes.
- DIVISION_POLICY_CONTEXT - a division context that is enforced as a policy.

Creating a Preference Hierarchy

A familiar example of a hierarchy is a modern corporation, which is composed of divisions. The divisions are composed of departments, which are composed of teams consisting of users.

This hierarchical structure could be managed by preference delegates for defining user preferences, as in the following structure:

- The bottom level in the hierarchy (users) would be managed by the wt.prefs.delegates.UserDelegate, which implements the Macro USER_CONTEXT.
- The top level (above the corporation) would be the DEFAULT_CONTEXT, managed by the wt.prefs.delegates.DefaultSystemDelegate.

These delegates can be replaced by customized delegates; however, the customized delegate must implement the DEFAULT_CONTEXT and USER_CONTEXT macros for the preferences framework to operate properly. See the Windchill Javadoc for details on the implementation of these delegates.

For the rest of the hierarchy, delegates must be written to extend PreferenceDelegate and implement the abstract methods. The number of delegates that perform this task can be variable, each level may have its own delegate, or a single delegate can be used to handle the body of the tree.

Use the following procedure to add a delegate to the preference hierarchy:

1. Add the delegate in the appropriate place in the delegates.properties file, located in <Windchill>codebase/wt/prefs/delegates.
2. Modify the `wt.prefs.delegates.DelegateOrder` value, if needed, to set the correct hierarchy.

Although this basic corporate organization has a tree structure, preference delegates are not limited to tree structures.

### Setting the Hierarchy

The `delegates.properties` value `wt.prefs.delegates.DelegateOrder` controls the hierarchy in which delegates are called. For each level in the hierarchy there should be an entry in this property. The customized entries should appear as `DIVISION_CONTEXT`. For example, in the out-of-the-box hierarchy, there is a division scope called Windchill Enterprise, and the out-of-the-box `wt.prefs.delegates.DelegateOrder` property value is:

```
SDEFAULT,$CONTAINER,$DIVISION:WindchillEnterprise,$USER
```

In this value, there is no `DIVISION_POLICY_CONTEXT` defined since `DIVISION_POLICY_CONTEXT` and `DIVISION_CONTEXT` are related and are at the same level in the preference hierarchy. Similarly, the `CONTAINER_POLICY_CONTEXT` need not be included. Entries are designated differently only when storing and retrieving preferences internally. For more details on correctly naming delegates, see the `delegates.properties` file.

**Note:** If `wt.prefs.delegates.DelegateOrder` has been removed from the `delegates.properties` file, Windchill uses the following:

```
SDEFAULT,$CONTAINER,$USER
```

A class that corresponds to the delegate is also required to be specified in the `delegates.properties` file. For the above Windchill Enterprise division this might be:

```
wt.prefs.delegates.class.$DIVISION/u003aWindchillEnterprise=wt.prefs.delegates.WindchillEnterpriseDelegate
```

The value for this preference is the full package path name to the appropriate delegate class. For example:

```
wt.prefs.delegates.WindchillEnterpriseDelegate
```

In the example, the `wt.prefs.delegates.WindchillEnterpriseDelegate` class would handle the Windchill Enterprise division. If these divisions are handled properly within the delegate, the delegate class is responsible for multiple divisions. When completed, this preference would be added to `preference.txt`, located in `<Windchill>/loadFiles/`, which loads the file each time the database is initialized (when installing Windchill).
Creating Delegates

All preference delegates should be extended from the abstract class wt.prefs.PreferenceDelegate. This class defines the following required methods which each delegate must implement within the preferences framework:

- public String getLocalized(String division, Locale aLocale)
- public boolean isAdministrator(String division, WTUser user)
- public ArrayList getDivisionsAsAdministrator(WTUser user)
- public ArrayList getDivisions(WTUser user)

The getDivisions methods differ in that the getDivisionsAsAdministrator(user) method returns a list of all levels in this delegate that the given user is responsible for or able to administer. The getDivisions(user) method returns all the divisions of which the given user is a member.

There are no restrictions on how the delegate decides which users are members or administrators of divisions; however, the getDivisions methods should return well-formed results. If a user is not a member of any division managed by the delegate, then the ArrayList should be empty, rather than null. For additional information on the getDivisions methods, refer to the Windchill Javadoc.

Managing User Preferences

User preferences can be set throughout Windchill. Out of the box, users can set preferences for their own workstations, and administrators can set preferences for each scope, where the preferences can be available to all lower level scopes.

The out-of-the-box hierarchy forms the beginning of a hierarchy that can include multiple levels, each defining a scope below the enterprise level. Preferences can be set for each scope defined in the system, including the user scope. Preferences (other than user) are set to either allow or forbid preference values to be overridden by scopes defined below the current scope.

The Preference Administrator preference table (located in ../wtcore/jsp/wt/prefs/admin/PrefHelp.jsp) displays the system default preferences and their description and default value.

Windchill includes four types of preference scopes:

- Default scope -- The top level, which defines the initial settings for all preferences at installation. This scope is not listed in the Selected scope drop-down list.
- Container scopes -- Container scopes for which you are an administrator are visible, starting with the specified container and going up to Site container. The name of a container scope is the same as the name of the container.
- Division scopes -- These are one level below container scopes. Out of the box, the Windchill Enterprise division scope is defined so that preferences can be set by Windchill administrators to apply to all users.
• User scope -- This scope is the lowest level in the hierarchy, and allows individual users to tailor preferences that allow overrides.

Each scope is controlled by a preference delegate. Documentation on preference delegates can be found in the wt.prefs.delegates javadoc.

Neither the default delegate, nor the user delegate should be removed from the system or changed; however, the Windchill Enterprise delegate can be changed or removed, and additional delegates can be defined.

The preference values at the default scope cannot be removed or changed at runtime; however, preferences at other scopes can be changed or removed (where applicable) by a member of that scope.

Using the Preference Administrator, you can create, edit, and remove user preferences, as described in the following sections. For detailed instructions on how to create, edit, and remove a user preference, click the Help button on the Preferences Administrator page. The help also contains a link to a table that contains a list of the system default user preferences.

Creating a User Preference

You can create any preference by giving it a unique name and a value; however, it will have no effect unless there is code written to use the preference.

Editing a User Preference

You can edit preferences at any level that are defined as overrideable at that scope.

Before you change a user preference, you should understand the type of values that the preference requires (for example, numbers and boolean).

Removing a User Preference

Preference values can be removed from a preference scope. When the value is removed, the value of existing preferences for lower-level scopes (including the user) either change or remain the same, depending on the following conditions:

• If overrides are allowed and preferences are set within the scope of the removed preference, then those preferences retain their values.

• If overrides are allowed and preferences are not set within the scope of the removed preference, then those preferences are reset to the values of the next highest-level preference scope.

• If overrides are not allowed and there is a scope that does not allow overrides, then all preference values below the scope that does not allow overrides are reset to the values of that scope.

• If overrides are not allowed and the overrides are allowed at all other scopes, then all preference values are reset to the lowest scope in which a preference is set.
Ensuring Proper Backup and Recovery

It is important that you either implement or request appropriate backup processes, such as the following:

- In a production environment, Windchill's Oracle database should be backed up on a regular basis. Oracle documentation provides additional information about backup procedures.

- At the time of installation, the Windchill installation directory must be backed up to preserve various configuration files.

- A given installation of Windchill and all source code should be backed up each time the system is regenerated.

- The following Windchill directories should be backed up on a regular basis: /db/sql, /codebase, and /src.

  You do not have to back up the entire /codebase and /src directories each time. However, you must back up the subdirectories containing Java packages that have been changed at your site.

- The RetrievalWare windchill_indexes working directory should be backed up on a regular basis. The RetrievalWare documentation provides additional information about backup procedures.

- Depending upon the user authentication mechanism at your site, you may need to ensure appropriate backup of files relevant to access control.

- Content files stored in an external file vault must be backed up using standard operating system tools and procedures.

- The Aphelion Directory should be backed up on a regular basis. How often you do this backup should be determined by how much activity is done in the solution that relates to the LDAP entries stored in the directory. For example, if user objects are stored in the directory, then you may want to back it up more often than if only group and organization objects are stored there.

Maintaining Log Files

Windchill log files contain exception tracebacks and other information that can be used for debugging code.

Each log file is enabled or disabled in the wt.properties file. If logging is enabled, you will also need to determine the appropriate settings for related properties. For example, the wt.name.log.append properties, which specify whether a log file is appended to or overwritten when the associated application is started.

Log file names and locations are controlled by Windchill properties. The $DATE(format) macro can be used to construct date-dependent file names. See the Javadoc for the class wt.util.WTProperties for information about the $DATE macro.
Configuring Your Windchill Environment

Where your Windchill server components reside should be based on the type and number of machines you have available. The following configurations are possible:

- Each Windchill server component can reside on a separate machine.
- Multiple components can be on the same machine.
- All components can be on a single machine.

The Windchill server components include the following:

- The Windchill client
- The Windchill application server (consisting of the server manager and one or more method servers)
- An HTTP Web server
- A J2EE servlet container
- A relational database server
- The Convera RetrievalWare search engine
- An LDAP server
- A reverse proxy server (optional)
- An authentication server (optional)

**Tip:** Many of these components can be deployed multiple times for load balancing purposes or to facilitate improved response times.

Configuring a Single Method Server

Method server startup and monitoring is provided by the StandardServerMonitor class, which runs on the corresponding server manager.

The following default wt.properties configuration starts up a single method server:

```
#Services to be monitored by the StandardServerMonitor
wt.manager.monitor.services=MethodServer

#Number of servers to start
wt.manager.monitor.start.MethodServer=1
```
Configuring a Method Server for Background Queues

The Windchill property settings described below allow you to configure a separate method server dedicated to running background queues.

The following \texttt{wt.properties} configuration start up two method servers. One is dedicated to running background queues:

\begin{verbatim}
#Services to be monitored by the StandardServerMonitor
wt.manager.monitor.services=MethodServer BackgroundMethodServer

#Number of Servers to start
wt.manager.monitor.start.MethodServer=1
wt.manager.monitor.start.BackgroundMethodServer=1

#Queue default execute setting
wt.queue.executeQueues=false
\end{verbatim}

Background queues can also be grouped to execute on specific method servers; see the \textit{Configuring and Administering Background Queues} chapter.

Load Balancing for Multiple Method Servers

Multiple method servers can be started on a single host to distribute and balance loads across multiple operating system processes. This may be beneficial if you are running a multiprocessor system that does not have native thread support.

In a multiple method server environment, the default setup performs a simple round-robin balancing on initial client connections only. Load balancing takes the balancing idea one step further to the actual method call. This gives the method server the opportunity to switch servers on clients at the level of individual method calls if server load is excessive.

Code changes are not required to use the load-balancing capabilities within Windchill. Properties within the \texttt{wt.properties} file control load-balancing behavior.

Server Selection

Selection of the next available server is performed by classes implementing the \texttt{wt.manager.ServerSelector} interface. Windchill provides the following two server selectors: \texttt{StandardServerSelector} and \texttt{BalancedServerSelector}.

Setting \texttt{wt.manager.serverSelector.<server name>} specifies the server selector to be used (for example, \texttt{wt.manager.serverSelector.MethodServer=wt.manager.BalancedServerSelector}).

The \texttt{getServer(String service)} and \texttt{getNextServer(String service, Remote server)} methods in the server selector interface deal with load balancing. The \texttt{getServer()} returns the server with which the client interacts upon initial connection. While,
the getNextServer() returns the failover method server to which the client switches when the current server surpasses a threshold.

<table>
<thead>
<tr>
<th>Server Selector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.manager.StandardServerSelector</td>
<td>Returns a server in a round-robin fashion on a getServer(). Returns the next server and wraps to the first server if called from the last server on a getNextServer(). This is the default selector.</td>
</tr>
<tr>
<td>wt.manager.BalancedServerSelector</td>
<td>Returns the first server on a getServer(). Returns the least recently used server on a getNextServer.</td>
</tr>
</tbody>
</table>

Threshold Detection

When a request is made on a method server, the current server checks to determine if any thresholds have been surpassed. If so, a wt.method.ServerLoadExceptions is thrown from this server and is caught by the remote method server. Within the exception is a reference to the next server. The remote method server then redirects the request to that server. The wt.method.loadbalance.maxRedirects property specifies the maximum number of times a single method call is redirected. The default setting is 1. A setting of 0 causes method calls to be redirected until a server that falls below the thresholds is identified.

The following thresholds are checked when load balancing is used:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.method.loadbalance.RMISockets</td>
<td>Defines the number of RMI sockets the server allows to be active before a ServerLoadException is thrown. Default is 0.</td>
</tr>
<tr>
<td>wt.method.loadbalance.activeContext</td>
<td>Defines the maximum number of currently active contexts allowed within the server before a ServerLoadException is thrown. Default is 0.</td>
</tr>
</tbody>
</table>

If a threshold is set to zero, or is not defined within wt.properties, the threshold is ignored.

Changing Authentication Text Between Servlet and Windchill Adapter

As part of the installation, Windchill solutions automatically generate a unique value for the secret.text2 and secret.text properties in the ie.properties file to establish a more secure authentication process between your servlet and the Windchill adapter. Both properties serve the same purpose, except that the secret.text2 property provides more secure connection using different underlying
code. The secret.text2 property is recommended and designed for use with Windchill 7.0, while secret.text property is recommended for prior releases.

The instructions in this section specifically reference secret.text2, however, you can use the same instructions to change the secret.text property.

Setting the secret.text2 property in the ie.properties file provides an arbitrary text string that is used to sign outgoing requests and validate incoming requests. When an out-of-process Windchill request is made to execute an adapter webobject, the adapter name specified in the INSTANCE parameter must identify an LDAP entry that has the secret.text2 property set. Using an LDAP entry created for the adapter that does not have the property set to the same value as is set in the ie.property file will not give the request access to the adapter.

The secret.text2 value set as part of the installation process is unique for each installation. If you want multiple adapters and Windchill installations to work together, you need to set the secret.text2 property to the same value for all adapters and installations.

Use the following procedure to change the value of the secret.text2 property in the ie.properties file:

1. Determine the value to assign the secret.text2 property and the location (fully-qualified) of the property file.

   PTC recommends using a secret.text value between 6 and 18 characters.

   The ie.property file is located in the <Windchill>/codebase/WEB-INF directory.

2. Use the xconfmanager to change the secret.text2 property to a value of your choice and to update the site.xconf file.

   From a windchill shell, execute the following command:

   xconfmanager -s secret.text2=<your_secret_value> -t <Windchill>/codebase/WEB-INF/ie.properties -p

   Where <your_secret_value> is an arbitrary text string and <Windchill> is the location where Windchill is installed.

   Any change you make to the secret.text2 property value must also be made to the adapter LDAP entries for all out-of-process adapters that send requests to Windchill. Use the Info*Engine Property Administrator to modify adapter LDAP entries.

**Administrering the Authentication Process**

The Windchill architecture is designed to rely on Web server authentication to provide authenticated user IDs. Therefore, access controls maintained on the Web server determine access privileges to an authenticated Windchill URL or SysAdm URL based on a user ID and password obtained by the Web browser.
The HTTP authentication implementation, described in more detail in the *Windchill Application Developer’s Guide*, results in the following Windchill configuration requirements:

1. Authenticated user names are Web server user names.

2. Windchill's authenticated HTTP gateway (defined by the `wt.httpgw.url.authenticated` property in the Windchill `wt.properties` file) must be subject to access control by the Web server, allowing only authenticated users to access it.

3. On the Web server, the Windchill HTTP gateway URLs must be aliased to the provided Windchill gateway servlet implementations.

Windchill's internal access controls are applied through the Windchill Administrator, as described in this guide. The Windchill Administrator application, in turn, associates each Windchill user ID with an authentication ID maintained by the Web server. The procedures that follow show you how to create user accounts and implement access controls for data residing on the file system:

**Note:** In the `wt.properties` file, the property `wt.auth.toLowerCase` is set to true by default, which forces authentication IDs to become lowercase. Therefore, you should not rely upon case to distinguish user IDs, unless you have changed the value of this property to false.

See the *Windchill Application Developer’s Guide* for information about customizing Windchill's authentication mechanism. See the *Windchill Installation and Configuration Guide* for more information about specifying anonymous access.

**Troubleshooting User Authentication**

Two tools included in the Windchill base product help identify user authentication configuration problems by exercising the authentication mechanism to verify that it is working and then reporting the user identities.

The `wt.auth.Authentication` class is the focal point for user authentication within Windchill. This class includes a main method so that it can be run as a stand-alone application. It exercises the configured login scheme and reports the resulting authenticated user name as seen by the Windchill method server. The following is an example of output for a failed HTTP authentication (canceled login), followed by a successful Null authentication:

```html
<html><head>
<title>401 Authorization Required</title>
</head><body>
<h1>Authorization Required</h1>
This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply
```
the credentials required.<p>
</p></BODY></HTML>
HTTP Login failed: java.io.StreamCorruptedException:
InputStream does not contain a serialized object
Reading user.name system property
Authentication.getUserName() jhs

To test authentication from within a browser, or to re-authenticate your current Windchill session to change the user name, use the login applet wt.clients.login.LoginApplet. The applet can be accessed through the HTML page wt/clients/login/Login.jsp, located in the Windchill codebase on the Web server. This tool displays the current authenticated user name (Web ID) associated Windchill user ID.

**Windchill Scheduler**

The Windchill Scheduler is an internal service used by different Windchill services to schedule execution of certain tasks. Tasks can be run once or periodically, and can be scheduled to for a particular time or immediately after the scheduling takes place. Typical scheduled tasks involve External Vaulting, Content Replication, and Product Replication.

Scheduled tasks are executed using the Windchill queue service, which allows the inheritance of the advantages of the background processing. For instance, if the Background Method Server is used, scheduled tasks will be running in it.

The Windchill Scheduler service keeps the log of each executed task in history objects that contain the current and historical status information. For example, if you are scheduling content replication and you select schedule items and click Log, history objects supply the data that appears in the Replication History window.

**Windchill Scheduler Automatic Removal of History Items**

The Windchill Scheduler periodically removes history items older than a specified number of days. For example, old history items no longer appear in the Revaulting History and Replication History windows, and data about them is not stored.

The following properties in the wt.properties file control the cleanup process and the lifetime of the history items:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.scheduler.purgeHistoryItems</td>
<td>Controls whether history items are periodically cleaned up. Its value is true or false. The value true enables the cleanup, and the value false disables the cleanup.</td>
</tr>
<tr>
<td>wt.scheduler.purgeHistoryItemsInterval</td>
<td>Specifies the number of days between the cleanups of history items thirty days old.</td>
</tr>
</tbody>
</table>
Other Properties

Other properties related to Windchill Scheduler operations are the following:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.schedular.verbose</td>
<td>Specifies whether to run the Windchill Scheduler in verbose mode. The default is false.</td>
</tr>
<tr>
<td>wt.schedular.log.properties</td>
<td>Specifies whether to print out the Scheduler-specific properties on startup.</td>
</tr>
<tr>
<td>wt.schedular.log.filename</td>
<td>Specifies the file to which the log file is written. The default is $(wt.logs.dir)\Scheduler.log.</td>
</tr>
<tr>
<td>wt.schedular.log.enabled</td>
<td>Specifies whether to save the log of Windchill Scheduler in a separate file. The default is false.</td>
</tr>
<tr>
<td>wt.schedular.log.append</td>
<td>Specifies whether to append to the end of the existing log file. The default is true.</td>
</tr>
</tbody>
</table>

Windchill Software Maintenance and Best Practices

Normal maintenance corrections and updates to the products of Windchill 7.0 are delivered primarily through a single cumulative installation image known as the Windchill Service Pack. Updates to a smaller subset of the products are delivered through a replacement CD image. Both the Windchill Service Pack and any replacement CDs can be ordered in CD form or downloaded from the PTC Software Update Web site (http://www.ptc.com/cs/doc/swupdate.htm). Each release of maintenance is identified with its own datecode, which is clearly visible from the Software Update Web site and through the installer programs. The datecode values, which are always increasing, identify a version within the overall 7.0 release. Datecode values for maintenance releases are of the form Mnnn where nnn is numeric, typically increasing by 10. Thus the first maintenance release is M010, the second is M020, and so on. In the event that a high priority problem must be delivered quickly, a temporary patch can be delivered in the form of an executable JAR file that will deliver and install the updates. Temporary patches are also uniquely identified, but not with a datecode value.

For many customers, part of implementing and deploying a Windchill solution involves adding customizations and possibly modifying some files delivered by PTC. Because some of the files changed by the site may also be updated in a
Maintenance Release, its important to carefully manage your changed files. Be aware that each time you install a Windchill Service Pack you will have to manually incorporate any PTC updates into your modified files. For detailed best practices on managing your site modifications, see the Windchill Solutions R7.0 Maintenance Information paper that is available from the PTC Technical Support Web site. The remainder of this section gives a brief overview of the Windchill Service Pack installation process and provides notes on areas where best practices will be provided.

By carefully managing site modifications following best practices, you can greatly improve the efficiency of the Windchill Service Pack installation and decrease production down time. It should be noted that just as in past releases, PTC recommends that you use a separate test system to prepare and validate the updates before installing them into production. This will help ensure minimal downtime during the installation into the production system. It is also advisable to perform a backup of the product installation directory prior to performing the installation.

Installation Process for the Windchill Service Pack

The Windchill Service Pack installation image is the delivery vehicle for updates to over half of the Windchill 7.0 products, including Windchill Foundation & PDM, Windchill ProjectLink, Windchill PDMLink, Info*Engine, and Information Modeler. It also delivers updates to a subset of the Workgroup Managers and gateway products. For these products, there are updates for both English and language specific versions. The specific list of products covered is provided through the documentation accompanying the Maintenance Release.

Updates are cumulative meaning that once a correction has been delivered in one version of the Windchill Service Pack, it is included in all future versions of the Windchill Service Pack. Each version of the Windchill Service Pack is identified with a different datecode.

When the Windchill Service Pack executes1, it copies new and updated files from PTC and performs various housekeeping operations (such as registering the installation of the updates, propagating XCONF file updates, re-building class files for enumerations that have changed, re-building client JAR files, and so on). In order to prevent the Windchill Service Pack installer from simply over-writing your site-modified files with updates from PTC, the installer runs two modes:

- Preliminary Installation. In the first execution, you can install the Windchill Service Pack into your test system where you can examine PTC updates and incorporate them into PTC files that you have modified. On the test system, you can then validate that the PTC updates, site modifications, and

---

1. This description applies to when the Windchill Service Pack is executed for the Windchill Installation directory where the Windchill Method Server is hosted. For other products, the installer primarily just copies in updated files and there are none of the special processing regarding site modifications are required.
customizations operate together. After the validation is complete, you collect all the site customizations together for easy deployment to production.

- **Install and Deploy.** In the second execution, the Windchill Service Pack installer executes on your production system. During this execution you direct the installer to pick up your collection of previously prepared site customizations and install them. This is done after PTC files are installed, but before the Windchill Service Pack housekeeping operations. After the Windchill Service Pack completes, the system is ready to be returned to production.

In order for the Windchill Service Pack to be used in this fashion, you must manage your modifications to the PTC files as prescribed by the maintenance best practices.

When you execute the Windchill Service Pack installer, it first determines which files should be installed onto your system. It does this by finding out which products are installed in the installation directory its being executed on and what datecode versions are already present. This results in the following behavior:

- If you do not have a product for which there are updates, the updates are not installed.

- Previously installed updates are not re-applied at every execution of the Windchill Service Pack. This means that if there were no changes for a product between different datecodes of the Windchill Service Pack, and you've already installed the Windchill Service Pack from the earlier datecode, they will not be re-installed on the next Windchill Service Pack installation.

- The installer will only update locale specific resources on your system if it finds that those locales were previously installed and registered through an installation of the Windchill Language Pack.

These features are intended to minimize the time it takes to install the Windchill Service Pack\(^2\). In particular this avoids re-installing updates to site modified files when you have previously incorporated those changes.

**Note:** Depending on what Windchill products you have installed and how they are deployed across one or more computer systems, you may have to execute the Windchill Service Pack installer in multiple installation directories on multiple computers.

Each execution of the Windchill Service Pack updates all the products it finds in a single installation directory, but it can only address one installation directory at a time. For this reason, one of the best practices is to keep a list of all the systems on

---

\(^2\) After you perform a first time installation any 7.0 product covered by the Windchill Service Pack, you should re-execute the Windchill Service Pack to install any recent updates and ensure the product is at a compatible level with the other products on your system. You should also repeat the Windchill Service Pack installation if add a new locale to your system through the Windchill Language Pack.
which a Windchill 7.0 product is installed. This list would identify the products and into which directories they are installed. Maintaining the list ensures that updates are applied to all the correct locations.

Instructions for installing the Windchill Service Pack are included in the documentation accompanying it at each maintenance release.

**Best Practices for Managing Windchill Installation and Maintenance**

As many customers implement and deploy Windchill, they find it necessary to modify some of the files provided by PTC. These changes include simple tuning of properties file entries, adding and modifying enumerated lists, altering displayed values for business types, attributes, modifying HTML or JSPs, and so on. Sites also add new classes, JSPs, HTML pages, extend the model, and so on. Some types of changes can conflict with updates provided by PTC during the maintenance cycle. Managing these updates properly will greatly simplify the installation of updates. It is important that those developing these modifications and building new customizations at your site understand how the maintenance process works and coordinate their updates with those responsible for installing software updates.

The details on best practices is provided through the *Windchill Solutions R7.0 Maintenance Information* paper that is available from the PTC Technical Support Web site. The topics in the paper include:

- Use of the xconfmanager utility and XCONF files for managing modifications to PTC property files, and for defining new property files for site customizations.

  For general information on the xconfmanager utility, see *Using the xconfmanager Utility*.

- Properly managing changes to enumerations (valid value lists), messages and displayed values for modeled business classes, attributes and associations by using RBINFO files. General information on this capability, including the enumCustomize utility for modifying enumerations is covered in the *Windchill Customizer’s Guide*.

- Properly managing modifications to existing HTML templates and creating new ones to replace standard PTC templates. Reference information is in the *Windchill Customizer’s Guide*.

- Properly handling modified JSPs, where it is permissible to do so.

- Ensuring that applet JAR files are updated properly as site modifications and customizations are added. These JARs must be updated with changes that occur on the server.

  **Note:** Windchill 7.0 has restructured these JARs to minimize the client download impact as updates through maintenance and customization occur.
About the windchill Command

PTC has provided a command, windchill, to invoke Windchill actions. For example, the command can be used to stop and start Windchill, check the status of the Windchill server, and create a new shell and set the environment variables. It can also be used as a Java wrapper. In that regard, it can accept a Class file as an argument, just like Java, and execute it without a predefined environment (Windchill classes in CLASSPATH, Java in PATH, and so on).

The windchill command should be used to execute any server-side Windchill Java code. This will insure that the environment that the command is executed in is properly setup. The environment that actions are executed within, including the windchill shell action, is defined by the wt.env properties in the wt.properties file. For example, the wt.env.CLASSPATH property will set the CLASSPATH environment variable for the action that is being invoked.

The windchill command is a Perl script that has also been compiled into a Windows binary executable. For UNIX systems, Perl 5.0 or greater must be installed. The windchill script assumes that Perl is installed in the standard install location of /usr/bin/perl. If Perl is not installed at this location, you can either create a symbolic link (recommended method) to the Perl install location or edit the windchill script to reference the Perl install location. To modify the windchill script, edit the `#!` entry (for example, `#!/usr/bin/perl -w`) and change the Perl directory to the location where Perl is installed.

The windchill command is located in the `<Windchill>`\bin directory. If you receive a command not found message when you execute the windchill command, add the `<Windchill>`\bin directory to your PATH environment variable. The syntax of the windchill command is:

```
windchill [args] action
```

You can display the help for the windchill command by executing windchill with the -h argument or with no argument.

The following tables list some of the arguments and actions applicable to the windchill command. To see a complete list of the arguments, use the report generated from the help (argument).

**windchill Arguments:**

<table>
<thead>
<tr>
<th>Arguments (optional)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>- h, --help</td>
<td>Displays help and exits.</td>
</tr>
<tr>
<td>-v, --[no]verbose</td>
<td>Explains what is being done when a command is executed. Default is noverbose.</td>
</tr>
</tbody>
</table>
### Arguments (optional)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w, --wthome=DIR</td>
<td>Sets the Windchill home directory. Default is the parent directory containing the windchill script.</td>
</tr>
<tr>
<td>--java=JAVA_EXE</td>
<td>The Java executable. Default is the wt.java.cmd variable value specified in the $WT_HOME/code-base/wt.properties file.</td>
</tr>
<tr>
<td>-cp, --classpath=PATH</td>
<td>Java classpath. Default is the wt.java.classpath variable value specified in the $WT_HOME/code-base/wt.properties file.</td>
</tr>
<tr>
<td>--javaargs=JAVAARGS</td>
<td>Java command line arguments.</td>
</tr>
</tbody>
</table>

### windchill Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shell</td>
<td>Sets up a Windchill environment in a new instance of the currently running shell.</td>
</tr>
<tr>
<td>start</td>
<td>Starts the Windchill server.</td>
</tr>
<tr>
<td>stop</td>
<td>Stops the Windchill server.</td>
</tr>
<tr>
<td>status</td>
<td>Retrieves the status of the Windchill server.</td>
</tr>
<tr>
<td>version</td>
<td>Displays the Windchill install version.</td>
</tr>
</tbody>
</table>
About the windchill shell

The windchill shell brings up a new command shell, from the parent shell that is setup for the Windchill environment. This includes setting all environment variables defined in wt.env property in the wt.properties file.

To execute the windchill shell, at the command prompt enter the following command:

```
windchill shell
```

When you are finished using the windchill shell, you can exit the shell and return to the parent shell.

PTC recommends running all server-side Windchill applications, tools, and utilities from the windchill shell. Also, you can use the windchill shell to set up your development environment to use javac or Java directly.

---

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties &lt;resource&gt;[,...][?key[&amp;key2]...]</td>
<td>Displays the properties as seen by Windchill for the given resource with substitution, etc. executed. It can be limited to a given set of keys. For example: windchill properties wt.properties — lists all wt.properties  windchill properties wt.properties?wt.server.codebase — lists server codebase  windchill properties wt.properties?wt.env.* — lists all the environment variables use by windchill shell  windchill properties — with no arguments generates the help report</td>
</tr>
<tr>
<td>CLASS [CLASS_ARGS]</td>
<td>Run a Windchill class with optional class arguments. For example: windchill wt.load.Developer -UAOps</td>
</tr>
</tbody>
</table>
2

Administering the Bootstrap Client

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>2-2</td>
</tr>
<tr>
<td>About the Bootstrap Feature</td>
<td>2-2</td>
</tr>
<tr>
<td>Java Plugin Cache</td>
<td>2-3</td>
</tr>
<tr>
<td>Bootstrap Configuration File</td>
<td>2-3</td>
</tr>
<tr>
<td>Bootstrap Package Versioning</td>
<td>2-7</td>
</tr>
<tr>
<td>Downloading JAR Files</td>
<td>2-8</td>
</tr>
<tr>
<td>Administering Codebases</td>
<td>2-8</td>
</tr>
</tbody>
</table>
Overview

The Windchill bootstrap loader is intended to make Java applets and applications usable over the Internet and on wide area networks, such as enterprise intranets and extended enterprise extranets.

When direct RMI socket connectivity is not possible from a client, installing the bootstrap loader will enable the client to tunnel RMI over other protocols. There are several reasons a client may not be able to make direct RMI connections to the Windchill server: a firewall sitting between the client and the Windchill server is blocking the Windchill RMI ports (5001-5010), client only has HTTP access through a client-side proxy, or the Windchill application server is on a different host than the applet's codebase (for example, reverse proxy or split web server/servlet engine).

This chapter provides background information on the bootstrap feature of Windchill, and information related to administrative responsibilities for creation and maintenance of JAR files when the bootstrap feature is enabled.

About the Bootstrap Feature

The bootstrap feature of Windchill allows Java applets and applications that would normally be downloaded from a server to be loaded from locally cached JAR files. This improves performance by eliminating the need to load Java class files and other resources from across the network.

The bootstrap feature automatically manages a cache of local JAR files that correspond to remote server codebases. (A codebase is the URL to the root of a directory tree containing Java class and resource files.) The bootstrap feature provides the following functionality:

- Preserves namespace separation between codebases
- Preserves the security of the sandbox to which code from each remote codebase is subject
- Does not add codebase JAR files to the Java system class path (the CLASSPATH environment variable) of the client system

A major benefit of using the bootstrap feature is that maintenance of each server codebase remains centralized, and no additional per-client administrative responsibilities are incurred. Even if a codebase undergoes frequent changes, the bootstrap feature recognizes the existence of new JAR files, and allows you to download the files.

To use the bootstrap feature, clients must have both the Windchill bootstrap package installed, and JAR files contained on their servers’ codebases. (If a client has the bootstrap feature installed, but a server codebase does not contain the required JAR files, the bootstrap feature is ignored. Similarly, the existence of JAR files in a server codebase does not affect clients that do not have the bootstrap feature locally installed.)
Java Plugin Cache

The JAR caching feature of the bootstrap loader, when used with applets, is similar to the Java plugin caching scheme. All Windchill supplied applets use the Java plugin and the plugin JAR caching mechanism. Although the Bootstrap loader may continue to be used with applets, it is recommended to use the plugin caching mechanism. Until other technologies are sufficiently mature, the Bootstrap loader is still recommended for remote Java applications. The Java plugin cache and bootstrap cache use the same JAR files available in the Windchill codebase directory. The Administering Codebase section later in this chapter is pertinent for both mechanisms.

Bootstrap Configuration File

The bootstrap package maintains a properties configuration file named .wtboot.properties in the directory identified by the Java user.home system property. The following sections show how you can use this file to control JAR file location and Java system properties.

Specifying JAR File Cache Location

The first time you use the bootstrap feature, you are prompted for the location in which to cache JAR files.

![Setup](image)

Windchill Bootstrap Setup

Saving bootstrap.properties. Please select local cache location.

Cache Location: [ ]

Bootstrap Properties File: C:\wtboot.properties

[Ok] [Cancel] [Help]

The location you specify is stored for future use in the .wtboot.properties configuration file that is located in the user’s home directory.

When browsing for the JAR file location, the Cache Location dialog box starts in the user.home location. The file name in the dialog box is a placeholder, as only the directory location is important. The open directory that is displayed in the Cache Location box is the actual location that is returned.
The following sample **Cache Location** dialog box shows that a directory named `cache` has been chosen:

Within the location you choose, subdirectories are created for each site from which JAR files are downloaded. These subdirectories correspond to codebase locations on each site. Each downloaded JAR file is accompanied by a properties file that contains information obtained when the file was downloaded.

Although you are required to specify only the cache location for JAR files, the following table lists the full set of properties supported by the bootstrap package:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowUserInteraction</td>
<td>Turns bootstrap user interaction on or off. Set to false, the autoDownload/autoInflate becomes important. Default is true.</td>
</tr>
<tr>
<td>autoDownload</td>
<td>Automatically downloads JAR files being cached by bootstrap. The user will be prompted. Default is false.</td>
</tr>
<tr>
<td>autoInflate</td>
<td>Automatically inflates the downloaded JAR files. The user will be prompted. Default is false.</td>
</tr>
<tr>
<td>cacheDir (required)</td>
<td>Defines the location where the bootstrap package maintains its cache of JAR files. This is a required property and has no default. If not specified, the user is prompted to choose a location.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>captureFile</td>
<td>Specifies the fully qualified class names of all classes loaded by a bootstrap loader. This is useful in determining how many classes are being loaded and can be used to build a list of class names for later use in the <code>[boot_preload]</code> parameter when bootstrapping an applet or application.</td>
</tr>
<tr>
<td>captureFileStackTrace</td>
<td>Writes stack traces to the capture file as a debugging aid to determine when and why classes are being loaded. This property has no effect unless captureFile is set. Default is false.</td>
</tr>
<tr>
<td>checkVersion</td>
<td>Accesses a particular server codebase and checks for a more recent version of the bootstrap package. Default is true.</td>
</tr>
<tr>
<td>enabled</td>
<td>Enables and disables bootstrap. Default is true.</td>
</tr>
<tr>
<td>rmiFailoverTimeout</td>
<td>Specifies, in milliseconds, the length of time for which the <code>wt.boot.WTRMIMasterSocketFactory</code> class waits before asynchronously launching alternate connection attempts. This property affects how quickly RMI fails over from direct socket connections to HTTP tunneling. This property is communicated to the socket factory by being set as a system property named <code>wt.boot.rmiFailoverTimeout</code>. Default is 10,000 (10 seconds).</td>
</tr>
<tr>
<td>rmiSocketFactory</td>
<td>Specifies the fully qualified class name of an RMI socket factory to be installed. Default is <code>wt.boot.WTRMIMasterSocketFactory</code>.</td>
</tr>
<tr>
<td>setProperty.xxx</td>
<td>Triggers the setting of arbitrary Java system properties, through a property naming convention. Any bootstrap property that has a name starting with <code>setProperty</code> specifies a setting for the system property identified by the remainder of the name. For example, <code>setProperty.user.language=FR</code> sets the <code>user.language</code> property to French.</td>
</tr>
<tr>
<td>showClasspath</td>
<td>Displays the Java class path. This can be used when debugging applets to see where the classes are being loaded from.</td>
</tr>
<tr>
<td>showMissingFiles</td>
<td>Displays resource files requested by the classloader that are not available within the cached JAR files. Resources are requested from the web server. Default is false.</td>
</tr>
</tbody>
</table>
Controlling Java System Property Settings

As described in the preceding table, you can use the configuration file to control Java system property settings. Because the bootstrapper is signed and trusted, it can be used to set system properties before applets are started. The bootstrapper searches the .wtboot.properties file and uses any properties with names of the form of setProperty.xxx to set the system property specified by xxx.

Because the bootstrap package can be granted special privileges, it can be used to set Java system properties before other classes are loaded and initialized. This makes it useful as a single, consistent mechanism to control your Java system properties across several Web browsers. This is especially handy when you want to work around default settings that do not produce the desired results.

Whenever a new JAR class loader is instantiated, the bootstrap properties (from the .wtboot.properties file) are examined. Java system properties are set when a property naming convention is triggered. Any bootstrap property that has a name starting with setProperty results in the setting of a system property using the remainder of the name. For example, the following bootstrap property sets the user.language system property:

```
setProperty.user.language=FR.
```

Some Java system classes are initialized at load time, using system properties, and are not normally affected by later changes to the system properties if the class is already loaded. Therefore, special support has been added to reset the default

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>useFullHostNames</td>
<td>Specifies if hostnames should be fully qualified for use within the bootstrap. Default is false.</td>
</tr>
<tr>
<td>verboseInstaller</td>
<td>Writes to System.out trace messages reflecting class loader reuse, downloading and installation of new JAR files, and construction of new bootstrap loaders. The output is short and can be used to confirm that bootstrap class loaders are being used. This property is used to debug bootstrapping problems. Default is false.</td>
</tr>
<tr>
<td>verbose loader</td>
<td>Writes trace messages to System.out reflecting classes and resources used by bootstrap class loaders. The output can be very large if many files are loaded. This property is used to debug bootstrapping problems. Default is false.</td>
</tr>
<tr>
<td>version</td>
<td>Specifies the version number for the currently installed bootstrap. This is the version number used by checkVersion.</td>
</tr>
</tbody>
</table>

Property | Description
---|--------------------------------------------------
useFullHostNames | Specifies if hostnames should be fully qualified for use within the bootstrap. Default is false.
verboseInstaller | Writes to System.out trace messages reflecting class loader reuse, downloading and installation of new JAR files, and construction of new bootstrap loaders. The output is short and can be used to confirm that bootstrap class loaders are being used. This property is used to debug bootstrapping problems. Default is false.
verbose loader | Writes trace messages to System.out reflecting classes and resources used by bootstrap class loaders. The output can be very large if many files are loaded. This property is used to debug bootstrapping problems. Default is false.
version | Specifies the version number for the currently installed bootstrap. This is the version number used by checkVersion.
locale in java.util.Locale if user.language or user.region properties are set. Similarly, running in Sun JVM, which is Sun's default implementation of HTTP URL connections, sun.net.www.http.HttpClient is reset when http.proxyHost or proxyHost properties are set. Because the properties that control SOCKS proxying in java.net.PlainSocketImpl are read each time, you do not have to do anything when those properties are changed.

Bootstrap Package Versioning

To ensure that the most current version of the Windchill bootstrap loader is available, the bootstrap package automatically tries to determine if the remotely installed bootstrap loader is newer than the locally installed version. The level of the current version is stored in a boot.properties file contained in the wt.boot package. During the check for a newer version, the version property from the local resource is compared to the version property available in the remote codebase. If the remote package identifies a newer version, the following dialog box opens:

![Windchill Bootstrap Download dialog box](image)

If the remote property file includes a downloadURL property, that value is displayed on the dialog box as the location from which you can download the newer boot.jar file.

**Note:** The wt.boot package in the Windchill codebase specifies a relative URL that points to the installation directory in the Windchill codebase. You can cut and paste the URL from this dialog box into a browser.
### Downloading JAR Files

When a remote JAR file is available, but not cached locally, you are prompted to download the file. Similarly, before a cached JAR file is reused, the remote codebase is checked to see if a newer version of the file is available. If a newer version is available, you are prompted for the action that you want to take. You can download the file immediately, continue using the old file, or dynamically download classes like a normal applet class loader would.

**Note:** The option to use an old file is enabled only if a previous JAR file exists locally.

When downloading the new JAR file (which is normally compressed), you can inflate the file. Inflating the JAR file after download makes the file bigger, but it avoids the processing time that is required to inflate entries when they are loaded later. Whether this CPU cost savings is worth the increased disk access to read bigger entries, depends on the hardware. A user with a fast CPU but very slow disk (laptop) might choose to leave the JAR file compressed.

If the specified JAR file is not available in the remote codebase, no local JAR file is used, even if a previous one is available locally. The bootstrap loader downloads classes like a regular applet class loader. To benefit from local JAR files, the remote codebase must contain up-to-date JAR files reflecting its content. The assumption is that, if a JAR file is no longer found, the codebase has undergone some sort of change that would invalidate the previously cached JAR file.

### Administering Codebases

It is the responsibility of the Windchill administrator to create and recreate codebase JAR files whenever any files in a codebase are changed.

The cached JAR files are standard JAR files. You can create them by using the Jar utility included in the Java Developer's Kit (JDK), or by using other Zip utilities as long as the resulting file names match those specified in bootstrap tags. They should be created to contain all the Java class files and resource files from the codebase that are required by the applet or application being bootstrapped. Any files referenced that are not in the system class path or the specified JAR file are not found.
**MakeJar Script for Client JAR Files**

Included with Windchill is a MakeJar script that contains the ant task for creating client JAR files.

From a windchill shell, enter the following command to update the contents of all client JAR files:

```
ant -f MakeJar.xml
```

**JAR Files and Security**

The JAR class loaders of the bootstrap package guarantee that classes loaded from cached JAR files are subject to the same security policies as if they were downloaded from the remote codebase by the normal applet class loader. This is usually a policy assigned to unsigned code or one associated with the remote codebase (site).

The local JAR file is merely a substitute for the remote codebase, so that all class and resources loaded by the class loader can be retrieved quickly without accessing the remote codebase. In all other respects, including security policies, behavior is identical to loading from the network. There is no benefit to having signed classes in the cached codebase JAR file because the classes are treated as if they were loaded over the network. (This would be the case for users without the bootstrap loader or a local file that was being ignored).
## Administering External File Vaults

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of Storing and Moving Data in Windchill</td>
<td>3-2</td>
</tr>
<tr>
<td>Overview of External File Vaults</td>
<td>3-2</td>
</tr>
<tr>
<td>The Central Cache Vault</td>
<td>4-4</td>
</tr>
<tr>
<td>Windchill Properties for File Vaulting</td>
<td>3-6</td>
</tr>
<tr>
<td>Windchill External Storage Administrator</td>
<td>3-8</td>
</tr>
<tr>
<td>Creating a Vaulting Policy</td>
<td>3-13</td>
</tr>
<tr>
<td>Managing Revaulting</td>
<td>3-16</td>
</tr>
<tr>
<td>Changing the Location of Files in External Vaults</td>
<td>3-23</td>
</tr>
<tr>
<td>Moving a Server Using External Vaulting to a Different Host</td>
<td>3-24</td>
</tr>
</tbody>
</table>
Overview of Storing and Moving Data in Windchill

Windchill offers several methods to increase the accessibility of data. The following brief summaries present the key features of these methods:

- **External File Vaulting** -- File Vaulting allows you to store Windchill data outside the Windchill database in logical containers called *vaults*, each of which can refer to multiple physical memory locations called *folders*. Multiple hosts can work together in file vaulting to form *sites* or *clusters*. You can create rules to upload specified data into vaults and folders. File Vaulting reduces the time for uploading and downloading data, and allows Windchill data access control, indexing, and notification policies for Windchill domains, while providing a transparent interface for the user. This chapter provides detailed information about file vaulting. You can accomplish many of the operations explained in this chapter through a command line interface that is explained in another chapter, *Configuring External File Vaulting or Replication With FvLoader*.

- **Content Replication** -- Windchill Content Replication allows you to compose rules that copy specified data from file vaults or Windchill databases to more rapidly accessible vaults known as *replica vaults*. Sites in Content Replication are of two types: *replica sites* to store data for rapid access and *master sites* which send data to replica sites. One site can play both roles. Security measures ensure that the data on replica sites is genuine. The data sent to replica sites does not include metadata. See the chapter *Administering Content Replication* later in this guide for detailed information about Content Replication.

- **Export and Import** -- Windchill Import and Export functions facilitate the exchange of content and metadata between Windchill sites and ProjectLink portals. Windchill Import and Export are available to software developers through an API. The Windchill user can access export functions to package in JAR files the data in the following top-level Windchill objects: folders, product structures, and documents. The Windchill user can import data from the JAR files produced by the export functions and place the data in local Windchill, free of change controls. See the chapter *Windchill Import and Export* later in this guide for detailed information about Windchill Export and Import.

Overview of External File Vaults

When a Windchill user creates information, such as a part or a document, content files can be associated with that object. Using file vaulting, you can specify that, for a particular type of object in a specific life cycle state, content files should be stored in a logical container called a *vault* on a system within your network, rather than in the Windchill database.

Each file vault contains folders which correspond to physical storage locations (for example, directories) on the host system. Based on the vaulting policy you...
establish, an uploaded file is stored in the file system location represented by the vault and folder to which it is assigned.

A Windchill site, or cluster, is a group of hosts with one URL. The hosts can be accessed individually or as a single unit. File vaults function as elements of sites. The following graphic summarizes the relationships between the entities that compose a site.

The symbols in the preceding graphic are identified by letters and correspond to the following components:

A. Windchill Method Server -- The Windchill Method Server that manages the Windchill database processes data and queries that pass between the Windchill database and the external file vaults.

B. Windchill Database -- The Windchill Database provides an interface for an Oracle database.

C. BLOB -- The Oracle database stores binary large objects (BLOBs).

D. File Vault -- The file system consists of multiple folders. The folders are located in one or more file vaults, which are logical constructs unassociated with particular locations.
E. Folder -- The folder represents a physical location.

F. Mount Path -- A mount path records the physical mounting that connects each folder and each host.

G. Hosts -- Multiple hosts form a cluster. One or more Windchill method servers can run on each host.

H. Database Communication -- The Windchill Method Servers can read the Windchill Database and write to it.

The Windchill File Vault Administrator displays a tree view of the site object, which includes all the hosts, vaults, and folders.

There are several benefits to file vaulting, which you should consider when making a decision about how content is to be stored:

- Uploading and downloading content files, which are common Windchill operations, are significantly faster when the files are stored in a vault rather than in the database.

- The storage location of content files is transparent to the user, so no user operations have to be modified.

Additional administration required to implement file vaulting is described in this chapter. Specific administrative tasks are described in the help files associated with the Windchill File Vault Administrator and the Windchill Administrator clients.

**The Central Cache Vault**

The Central Cache Vault is created on the local system during the startup of a new or migrated Windchill system in order to enable faster file upload for certain applications without, or prior to, the system administrator setting up a custom cache vault.

Initially located in a temporary directory [vault name "defaultcachevault", folder "defaultcachedir", mounted from $9wt.temp)/CacheVlt1 on localhost], this cache vault can be relocated by relocating its folder as follows:

1. Update the folder an assign it read-only status to prevent additional files from being uploaded to the current location.

2. Copy the existing files to the new storage location.

3. Update the mount with the new path specification.

4. Update the folder again to clear the read-only status.

For more information on folders and mounts see “Adding and Updating Folders” and “Creating and Updating Mounts” later in this chapter.
Though the Central Cache Vault can be used as any normal vault for file storage, you typically designate a different vault or the Windchill database for long term file storage. In that case, files uploaded to the Central Cache Vault will be revaulted to their designated storage location. The value of the property `wt.fv.uploadtocache.revaultOnCommit` can be set true or false, with the following results:

- true -- All documents modified during the transaction are added to the queue (RevaultCacheQueue) for revaulting to their designated storage area upon completion of the transaction
- false -- You must configure a Revaulting Schedule to periodically revault files to their designated storage area. For more information see the section “Managing Revaulting” later in this chapter.

If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

Because a cache vault accumulates unreferenced files more quickly than other vaults on a site, regular file cleanup is necessary. For more information on vault cleanup see the section “Maintaining Your Vaults” later in this chapter.

A scheduler item created at startup will periodically execute to clean up unreferenced database information. Two `wt.properties` control the timing of the vault purging (default is daily) and the age of unreferenced items to be purged (default is 30 days). You can modify the values of these properties to suit your requirements. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.
Windchill Properties for File Vaulting

Set the following Windchill properties, defined in the wt.properties file to configure your file vaulting environment. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.fv.verbose.properties</td>
<td>Determines whether or not file vault properties are to be output upon startup. The default is false.</td>
</tr>
<tr>
<td>wt.fv.verbose</td>
<td>Specifies whether the system outputs log information specific to the file vault feature. The default is false.</td>
</tr>
<tr>
<td>wt.fv.read.buffer_size</td>
<td>The size of the buffer for uploading files directed to a vault. The default is 8192 (8 KB).</td>
</tr>
<tr>
<td>wt.fv.cleanup.buf_size</td>
<td>The size of the buffer for file vault clean-up operations. This buffer is used to read the file names of the folder from which unreferenced content files are to be removed. The default is 10,240 file names (80 KB).</td>
</tr>
<tr>
<td>wt.fv.log.enabled</td>
<td>Enables the file vault log file. The default is false.</td>
</tr>
<tr>
<td>wt.fv.log.append</td>
<td>Specifies whether or not file vault logging information is to be appended to the log file (rather than overwriting the file content). The default is true.</td>
</tr>
<tr>
<td>wt.fv.log.filename</td>
<td>The name of the file vault log. The default is $(wt.logs.dir)\FileVault.log.</td>
</tr>
<tr>
<td>wt.fv.log.mountInfoFile</td>
<td>The name of the log file for mounting information. The default is $(wt.logs.dir)\MountInfo.log.</td>
</tr>
<tr>
<td>wt.fv.revaultOnChange</td>
<td>Determines whether or not the revaulting is performed in the background for the objects changing their domain and/or life cycle state. The default is true.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wt.fv.revaultQuerySize</td>
<td>Maximum size of the bucket to be used during revaulting processing. Increasing this parameter decreases the time revaulting takes but increases memory use on the method server. Decreasing this parameter decreases the memory use on the method server, but increases the time revaulting takes. The default is 1000.</td>
</tr>
<tr>
<td>wt.fv.uploadtocache.revault OnCommit</td>
<td>Determines whether or not the documents modified during the transaction will be added to the RevaultCacheQueue upon the completion of the transaction. The default is true. If set to false, a Revaulting Schedule must be implemented to revault files.</td>
</tr>
<tr>
<td>wt.fv.purgeUnreferencedFvItemsInterval</td>
<td>Determines the periodicity in days of the execution of the cleanup of unreferenced items. The default is 1.</td>
</tr>
<tr>
<td>wt.fv.purgeUnreferencedFvItemsOlderThan</td>
<td>Determines the age in days of unreferenced items which will be subject to cleanup. The default is 30.</td>
</tr>
<tr>
<td>wt.fv.forceContentToVault</td>
<td>Determines whether a single vault will be used for all content vaulting. The default is false. See Forcing Content to Vault.</td>
</tr>
<tr>
<td>wt.fv.useFvFileThreshold</td>
<td>If true, the property wt.fv.fvFileThreshold is effective. If false, wt.fv.fvFileThreshold has no effect.</td>
</tr>
<tr>
<td>wt.fv.fvFileThreshold</td>
<td>Value of this property sets the maximum number of files that each folder associated to a vault can hold. For example, consider the property to have the value &quot;n.&quot; At the moment that n files are in a folder, the folder becomes read only, and the next content file is vaulted to the next folder mounted to the vault.</td>
</tr>
</tbody>
</table>
Windchill External Storage Administrator

In order to implement file vaulting, you use the *Vault Configuration* window to define the following items:

- **Site** (also known as *cluster*) -- A group of hosts with one URL that can be accessed independently but also as a single unit. The site of interest in file vaulting is the automatically generated master site, which has the default name Master.

- **Host** -- a system on your network that can be used to store content files

- **Vault** -- a logical container for folders

- **Folder** -- a representation of a physical storage location on a host system

- **Mount** -- a folder is associated with a host by a mount

You display the *Vault Configuration* window by clicking *Vault Configuration* in the *External Storage Administrator* window, which you access from the Windchill home. The following figure shows the *Vault Configuration* window:

![Vault Configuration Window](image)

The left panel of the *Vault Configuration* window displays a tree view of the site object, which includes all of the hosts, vaults, and folders that have been defined for the site. The *Vault Configuration* window shows icons only for the site to which you are connected and for Content Replication replica sites. The content of
the right panel depends on whether or not you have requested a display of all possible mounts.

When the radio button labeled **Show only existing mounts** is selected, the right panel displays all mounts associated with the selected host, vault, or folder.

When the radio button labeled **Show all possible mounts** is selected, the right panel displays the following:

- If the site is selected, all possible mounts for the hosts on that site.
- If a host is selected, all possible mounts to that host. Double-click a potential mount (a mount with no defined path) to invoke the **New Mount** dialog box. Double-click an existing mount to open the **Update Mount** dialog box.
- If a vault is selected, all folder and host combinations possible or already defined for that vault. Because they are logical entities, vaults cannot be mounted.
- If a folder is selected, all hosts on which that folder is or could be mounted. Double-click a potential mount (a mount with no defined path) to invoke the **New Mount** dialog box. Double-click an existing mount to open the **Update Mount** dialog box.

You can use the **Vault Configuration** window menus and toolbar buttons to perform such actions as the following:

- Create and update vault components
- Schedule revaulting
- Generate backup information
- Enable and disable the status of objects
- Clean up your vaults by removing unreferenced files.
- Validate a single mount or all mounts of an object.

**Adding and Updating Hosts**

Select **File > New > New Host** to add a host. Select **Object > Update** or double-click a host icon to modify the selected host.

Hosts are identified by the Domain Name Service (DNS) name or IP address you specify. If you enter a DNS name, ensure that no blank values are included in the name specification.

The External Storage Administrator does not verify the validity of the host name you enter.
Adding and Updating Vaults

Select **File > New > New Vault** to add a vault. Select **Object > Update** or double-click a vault icon to modify the selected vault. Vault attributes you can specify include the following:

- **Vault name.** All vault names must be unique.
- The vault's access status. If **Read Only** is selected, the vault is not available to store uploaded files, but it continues to be accessible for download requests. For example, you may decide to mark a vault as read-only when moving files from one vault to another, to prevent the upload of additional files.
- When you are updating an existing vault, the **Update Vault** dialog box displays the sequence number of the folders within the vault. The sequence numbers determine the order in which Windchill searches the vault for a writable folder in which to store content. Click **Show Only Writable** to restrict the display to folders that allow write access. Use the available buttons to change a folder's sequence number.

Deleting a Vault

If a vault does have a folder or folders, those folders should be either deleted or moved to another vault before you delete the vault. Execute the following steps to delete an external file vault or replica vault:

1. Stop and delete all scheduled activities associated with the vault to be removed, such as revaulting or replication.
2. Delete all the policy rules associated with the vault.
3. For external file vaults only, run revaulting on the vault to move all the content from it to another location.
4. Remove all of the vault’s folders.
5. Remove the vault by selecting it from the tree on the left side of the Vault Configuration dialog and selecting **Delete** in the **File** menu.

Adding and Updating Folders

Select **File > New > New Folder** to add a folder. Select **Object > Update** or double-click a folder icon to modify the selected folder.

Folder attributes are as follows:

- **Unique folder name.**
- **Vault location.** All folders must belong to a vault, which you select from the drop-down list on the **New Folder** dialog box. You can move a folder to another vault when updating it. This may be desirable if, for example, the folders in the current vault are becoming full. If information is stored in the folder, the following steps are the best way to change its location:
a. Select the folder from the tree on the left side of the Vault Configuration dialog.

b. From **Object** menu select **Update**.

c. In the Update Folder dialog select the new vault location for the folder.

d. Click **OK**.

- **Access status.** If **Read Only** is selected, the folder is not available for storing uploaded files, although it remains accessible for download requests. A folder is automatically marked read-only when the physical storage location it represents becomes full. You may also want to mark a folder as read-only to prevent uploading of additional files when you are changing a folder's mount location.

- **Enabled status.** A folder must be enabled before it can be used to store content files, and it must be mounted before it can be enabled.

### Deleting a Folder

1. Select the folder from the tree on the left side of the Vault Configuration dialog.

2. Select **Delete** from the **Object** menu.

   If the selected folder is empty, the preceding steps delete it. If the selected folder is not empty, you can make it possible to delete by changing rules to move files from this vault to another vault and by performing revaulting.

### Creating and Updating Mounts

A mount is the association between a folder and a host. When you create or update a mount, you specify how the Windchill method server running on the host accesses a specific file storage location on a host.

Select **Object > Mount** to mount the selected object. Select **Mounts > Update Mount** to modify an existing mount. **Mounts > Unmount** removes an association between folder and host.

**Note:** All folders must be mounted to all available hosts. Otherwise, a method server running on a host without a mount is unable to access content files when a download operation is requested. When a folder is mounted to more than one host, all mount paths must point to the same physical location.

When defining or updating a mount, you must specify the following:

- **The folder to be mounted on the specified host system.**

- **The path to the physical storage location represented by the folder.** If a mount is directed to a nonexistent path, uploads to and downloads from this folder will fail.
Changing the path value associated with a mount makes all files stored in the previous location inaccessible until they are moved to the new path location. Use the following update procedure to avoid download failures when you are changing a mount location:

1. Update the folder and assign it read-only status to prevent additional files from being uploaded to the current location.
2. Copy the existing files to the new storage location.
3. Update the mount with the new path specification.
4. Update the folder again to clear the read-only status.

Maintaining Your Vaults

To free up disk space, you may want to perform periodic maintenance on vaults and folders to remove unreferenced files. An unreferenced file is one that no longer has a valid association to a Windchill object. Select Object > Remove Unreferenced Files to perform this clean-up operation.

When you request removal of files, those files are permanently deleted from the host system. Therefore, the system issues a message suggesting that you request backup information before continuing with the clean-up operation. When you select New > Generate Backup Info, the Windchill method server writes to the log file identified by the $(wt.fv.log.mountInfoFile) property in the wt.properties file. This file contains file vault mounting information, in the following format:

```
<hostname><SPACE><mount path>
```

You can use this file to configure your system back-up tool for effective protection of your file vaults. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

In addition, the following rules govern deletion of vault objects:

- When a host with mounts is deleted, all the mounts associated with that host are also removed. Consequently, folders associated with this host are no longer mounted to it, but may remain mounted to other hosts.
- You cannot delete a vault that contains folders.
- You cannot delete a folder that contains content files.
- When a folder is deleted, all of its mount connections are also removed.
Creating a Vaulting Policy

As described in the *Windchill Business Administrator’s Guide*, the Windchill Administrator application can be used to establish access control, indexing, and notification policies for specific Windchill domains. Similarly, you can establish a file vaulting policy that identifies the vault and folder to which content files are to be uploaded, based on the domain location and life cycle state of the object with which they are associated.

Windchill domains can be created in a hierarchical fashion, with some domains being children of other domains. It is important to note that a domain does not inherit the vaulting rules of its parent domain. Vaulting rules must be explicitly defined at each level of a domain hierarchy.

Examining Existing Rules

Before creating, modifying, or deleting existing rules, you may want to examine these rules. Begin by clicking the **Vaulting Rules** icon on the **External Storage Administrator** page. This brings up the **Administrative Domains** selection window, from which you must select a domain. Full domain paths are shown in the **Administrative Domains** selection window, beginning with a root domain represented by a slash (/). Click **Update** to display the **Domain Vaulting Rules** window for the selected domain, with the **Vaulting** tab selected. If you want to see the vaulting rules already in place for the domain, click **Retrieve** to get the information from the database and populate the display with existing rules, each consisting of a class, a state, and a file vault. The following example shows the
Domain Vaulting Rules window for the Parts domain, with the Vaulting tab selected.
Creating New Rules

To create new vaulting rules, click Create to open the Vaulting Rule window, on which you can make the necessary selections:

![Vaulting Rule Window]

Note that the Classes pane contains a hierarchical tree showing the classes in the domain for which you can create vaulting rules. To create a new rule, select an object class to which the rule will be applied. Because the classes are hierarchical, a rule created for the class you select is extended to its subclasses as well.

The classes that are displayed may not include some abstract classes, but the classes shown are the complete set of classes that can appear in valid rules.

Next, select a state type from the list of life cycle states. Finally, select a file vault from among the list of vaults you defined by using the Windchill External Storage Administrator. Note that there can be only one class, life cycle state, and vault specified within a single rule. Additionally, a single object type and life cycle state combination can be linked to only one file vault.
When determining the vault to which to direct content files when an upload operation is requested, the file vault service applies the most specific, valid rule. For example, consider the following rules:

- Rule 1: <User, WTDocument, All> Vault1
- Rule 2: <User, WTDocument, InWork> Vault2

Assume that a given document object (WTDocument) is associated with the User domain and is in the InWork life cycle state. Rule 1 would direct its content to Vault1, regardless of its life cycle state. However, Rule 2 indicates that content files should go to Vault2 when the document is in the InWork life cycle state. So, in that case, the most specific rule would be applied, and any content associated with the document would be stored in a folder within Vault2.

**Note:** Currently, content files are moved into a vault only when an object is checked into the Windchill database and its content files are uploaded. Therefore, a file does not automatically move to a new vault when the life cycle state of the object changes. Rather, the file is moved to the appropriate vault the next time it is uploaded.

When you are satisfied with your selections, click **OK** to save the rule and exit the window. Click **Cancel** to exit the window without saving the rule.

If you return to the **Domain** window, the list of vaulting rules will include the rules you created in this session.

### Modifying and Deleting Rules

To update a vaulting rule, select it from the list displayed and click **Update**. When a rule is updated, only the selected vault may be changed. The class and life cycle state remain constant.

To delete a rule, select it from the list displayed and click **Delete**.

### Managing Revaulting

When a vaulting rule is created, modified, or deleted, it is necessary to relocate the files to their new home. This process is called revaulting.

Revaulting is necessary when a vaulting rule is modified to use another file vault or when a vaulting rule is deleted, which is equivalent to designating the object storage to be in a BLOB. Revaulting may also be needed when a change occurs in the domain or life cycle state of an object that holds content files. The revaulting process for such object changes can be done in the background, which is administered by a property, wt.fv.revaultOnChange. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.
Revaulting has the potential to be a resource-intensive activity. Therefore, it needs to be managed. To designate a vault for revaulting, it is necessary to schedule when it is to be done. Creating a schedule item for the vault does this. At the scheduled time, the revaulting process is launched and the contents of the vault are either relocated to another vault, moved from a vault to BLOB storage, or moved from a BLOB to a vault automatically. Use the Remove Unreferenced Files menu item to clean up the vault storage after the revaulting process.

Managing revaulting is primarily the routine of scheduling when the revaulting should occur for each vault and periodically monitoring its progress.

**Examining Existing Revaulting Schedule Items**

Before creating, modifying, or deleting existing revaulting operations, you may want to examine the existing schedule items. Begin by clicking the Revaulting Scheduler icon on the External Storage Administrator window.

The External Storage Scheduling window opens.

A list of vaults is displayed, for which revaulting has been scheduled. Each vault displays the frequency of revaulting, the time of the next revaulting run, and the status of the current run. A status of READY means that the run has been scheduled.
Click **Refresh** to update the contents of the list box.

**Viewing the Results of Revaulting**

To review the results of the revaulting operations, select a revaulting schedule item from the list box and click **Log**.

The **Revaulting History** window opens.

![Revaulting History Window](image)

The vault that the history is for is displayed, with the time it was submitted for execution, its completion time, and the status of all revaulting runs. The completion time of a given run should be earlier than the submission time of the next revaulting run. If this is not the case, you should increase the period length.

**Creating a Revaulting Schedule Item**

There are two ways to schedule an item for revaulting.

- You can select a vault in the **Vault Configuration** window and click **Object > Revaulting** to display the **Revaulting Scheduler** window. Complete specifications in the window and click **Apply**.
You can click **Create** in the **External Storage Scheduling** window to display the **Revaulting Scheduler** window. Complete specifications in the **Revaulting Scheduler** window and click **Apply**.

Revaulting should be done on a regular basis. Since it can be a resource intensive operation, PTC recommends that the revaulting be scheduled for a time period with the least system activity.
A revaulting schedule item can be in four modes of operations. You can set these modes by setting the radio buttons to the appropriate state.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate/Once</td>
<td>The revaulting begins immediately and runs only once.</td>
</tr>
<tr>
<td>Immediate/Periodic</td>
<td>The revaulting on this vault occurs at the period specified by the spin boxes. The base time is when the screen is dismissed.</td>
</tr>
<tr>
<td>On Start/Once</td>
<td>The revaulting begins when scheduled and runs only once.</td>
</tr>
<tr>
<td>On Start/Periodic</td>
<td>The revaulting runs regularly on the selected vault.</td>
</tr>
</tbody>
</table>

If the edit timing window was accessed from the Revaulting Scheduler window, the Apply button is enabled. This enables an administrator to schedule revaulting on all the vaults in the system in one session.

Click OK to save the changes to the repository, and close the window. Click Clear to reset the window to its original state. Click Cancel to close the window without saving any current changes. Schedule items saved to the repository with the Apply button can not be undone.

**Updating a Schedule Item**

You may update the timing parameters of a revaulting schedule item by double-clicking the item in the list. You can update certain parameters, depending on the details of the mode, as well as the status of the revaulting process.

The system keeps track of these rules for you by enabling/disabling the widgets in an appropriate fashion.

**Viewing a Schedule Item**

To view a schedule item without updating it, select the item for which you wish to view the timing information, and click View.

**Canceling a Schedule Item**

To cancel a schedule item, select the schedules item on the list and click Cancel. The schedule item will no longer run, but the history of the schedule items executions will be retained.
Deleting a Schedule Item

To delete a schedule item, select it and click Delete. Deleting a schedule item will cause the schedule item to be deleted and its history destroyed.

Forcing Content to Vault

With the increasing capability of certain applications comes a possibility that the increased number of vaults and file vault policy rules may become unmanageable. To control this situation, the property wt.fv.forceContentToVault has been introduced. If set to false (default), the system functions as if the property did not exist. If set to true, the property forces vaulting to be accomplished through a single vault in the following way:

- The MethodServer will not start if there is more than one vault present in the system. Therefore, you need to remove all vaults but one before enabling this property. A message will appear in the MethodServer log describing the problem.
- If users attempt to create more than one vault, they receive an error message stating that they cannot create more than one vault if the property is set to true.
- If re-vaulting is scheduled on the existing vault, all vaultable content will be moved to this vault.
- On content upload, all vaultable content will go to the existing vault. Therefore, this vault must be marked as Designated for Cache.
- File vaulting policy rules will be ignored.

If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

Avoiding Storage of Content in Oracle Blobs

Forcing content to a single vault increases performance by preventing content files from being stored in Oracle BLOBs, and also reduces the need for a proliferation of vaulting rules.

**Note:** When you use file vaults, the database and file system backups are no longer synchronized. Vaulting is recommended only in situations where the storage is fault tolerant, such as by use of RAID (Redundant Array of Independent Disks) or mirroring, to minimize the risk of data loss in the event of a single drive failure.

Converting from Multiple Vaults to a Single Vault

To convert from a multiple vault to single vault external storage configuration, use the following procedure:

1. Delete scheduled revaulting entries.
2. Use the following SQL statements to delete existing revaulting rules:
   
   ```sql
   delete FvPolicyRule
   delete FvPolicyItem
   ```

3. Reassign all storage folders to the vault designated as your cache vault.

4. Delete all vaults except the designated cache vault.

5. Add the following to the wt.properties file. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.
   
   ```properties
   wt.fv.forceContentToVault=true
   ```

6. Restart the method server

7. Ensure that the designated cache vault is properly configured for a large number of data sets.

8. Initiate revaulting to the designated cache vault so that any remaining data is moved from BLOBs to the vault.

**Changing from a Multiple Vault to a Single Vault Architecture**

The following steps show a method composed of deleting rules and revaulting. Another approach is to move all folders from vaults that you will delete to a vault that will remain.

1. Stop all scheduled revaultings.

2. Use the following SQL statements to delete existing revaulting rules:
   
   ```sql
   delete FvPolicyRule
   delete FvPolicyItem
   ```

3. Create a csv file with new external storage rules and a cache vault chosen to be the storage location. Comment out the final rule.

4. In wt.properties, comment out the following FvService entry and restart the method server. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.
   
   ```properties
   wt.services.service.33 = wt.fv.FvService/wt.fv.StandardFvService
   ```

   **Note:** Services numbering has to be sequential, so take the last service and assign it whatever number FvService had previously;

5. Load the csv file with rules.

6. In wt.properties, restore the FvService entry and restart the method server. If you are not setting properties through a graphical user interface or in a
mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

7. In the rules csv file, comment out or delete all entries and uncomment the final entry.

8. Load the rules csv file.

**Note:** This may take a considerable amount of time.

9. Revault the cache vault;

10. Remove all vaults except the cache vault;

11. In wt.properties, add the entry, wt.fv.forceContentToVault=true, and restart method server. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

### Changing the Location of Files in External Vaults

You may wish to change the location of files from one external vault to another. The following procedure shows how to move files from one vaulted folder, for example, /opt/windchill/vaulting/folder-001) to a folder in a different vault, for example, /usr/vaulting/folder-001) without affecting Windchill.

To change the location of vaulted files, perform the following steps:

1. Log into Windchill as an administrator.

2. Click **System Administration > External Storage Administrator > Revaulting Scheduler**. In the External Storage Scheduling window, either verify that no revaulting is currently underway or cancel all revaulting which is (or is about to be) in progress.

3. Click **External Storage Administrator > Vault Configuration**.

4. In the Vault Configuration window, click on the **Folders** node and select the folder that you want to move.

5. Click **Object > Update**. In the **Update Folder** dialog box, check **Read-Only**, then click **OK**.

6. Copy all the files stored inside /opt/windchill/vaulting/folder-001 to /usr/vaulting/folder-001.

7. Select **Object > Mount** and update the mount location from /opt/windchill/vaulting/folder-001 to /usr/vaulting/folder-001.

8. To test the success of the location change, rename opt/windchill/vaulting/folder-001 and try to access the content of a Windchill object stored in the external vault.
9. In the Vault Configuration window, select the folder you have moved and click Object > Modify. In the Update Folder dialog box, clear the Read-Only checkbox you checked in Step 5.

10. In the External Storage Scheduling window, restore any rescheduling operations you canceled in Step 2.

Moving a Server Using External Vaulting to a Different Host

You may wish to move a Windchill server using external vaulting to a different host. The following example gives a procedure for moving a Windchill server on Host A with file vaulting in f:\vaulting\folder-001 to Host B with file vaulting in g:\vaults\folder-001, assuming that both hosts have running Windchill installations.

1. Terminate the Windchill server on Host A.

2. Modify db.properties on Host B to identify it as the same oracle user as Host A. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

3. Do one of the following:
   – Copy all the vaulted files from Host A onto Host B. (i.e. from f:\vaulting\folder-001 to g:\vaults\folder-001).
   – Make sure the vaulted files (in their original location) are accessible from Host B.

4. Start the Windchill server on Host B. A warning appears, stating that the master site was modified and that you should make sure the modification is correct by checking Replication Administrator or Site Manager.

5. Go to External Storage Administrator > Vault Configuration. The Vault Configuration window appears.

6. Expand the tree until you can see Host A under the Hosts node.

7. Double click on Host A. In the Update Host window change the host name to Host B.

8. Click on Folders and select the one you want to modify.

9. Click Objects > Mount and update the mount location from f:\vaulting\folder-001 to g:\vaults\folder-001.

10. To test if you’ve been successful, try to access the content of a Windchill object stored in the external vault.
## Administering Content Replication

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>4-2</td>
</tr>
<tr>
<td>Setting Up Sites and Keys</td>
<td>4-4</td>
</tr>
<tr>
<td>Editing the wt.properties File</td>
<td>4-17</td>
</tr>
<tr>
<td>Content Replication and Windchill Visualization Service</td>
<td>4-19</td>
</tr>
<tr>
<td>Replication and Compression</td>
<td>4-21</td>
</tr>
<tr>
<td>Improving Content Replication Performance</td>
<td>4-22</td>
</tr>
</tbody>
</table>
Overview

Windchill Content Replication increases the productivity of Windchill users by reducing their time to access data. The users access data stored on more rapidly accessible external vaults known as replica vaults. Replica vaults store data that has been replicated from less rapidly accessible external vaults or from the Windchill Oracle Database. The Windchill user's experience in accessing replicated and non-replicated information is identical except for the improved access time. The Windchill user's only explicit interaction with Windchill Content Replication is setting preferences in a graphical interface.

A Windchill cluster or Windchill site is a group of hosts with one URL. For the purpose of Content Replication, a site can play the role of master site or replica site or both. When a site is playing the role of a master site, content can be replicated from Oracle LOB storage or from external storage or both to one or more replica sites. When a site is playing the role of a replica site, content can be replicated to it from master sites.

A master site stores configuration information about the replica sites that receive replicas of its data. A replica site provides Windchill users faster access to data in replicated vaults. The data in each replicated vault can come from only one master site, and attempts to disregard this rule result in the loss of data.

The method servers of sites that are playing the role of master or the roles of both master and replica must run off a Windchill Oracle repository. A replica site can run in a lightweight mode that requires only minimal Windchill services that support the receipt of configuration information and the processing of requests to replicate or download content. The advantage of running in this lightweight mode is that no Oracle instance is needed.

The security of data sent by Windchill Content Replication is assured by a pair of keys associated with each master site server. A request sent by a master site is digitally signed using a private key, and the public key is a vehicle for authenticating that the private key used by the request is genuine. The replica service verifies that all the URLs from which to download originate from the master site by using the master site's public key. The same checking procedure is used during the replication process to insure that the replicated items came from a registered master site. The public key copied to a replica site must be genuine, and permissions should protect it from alteration.

The clocks at master and replica sites must be synchronized to insure correct key validation. A difference between the clocks of more than five minutes prevents validation. The URL of a replicated document expires five minutes after its creation. The five minute period is a default that you can alter on replica sites.

Content rules for replication can be defined on the basis of domain, class, and life cycle state. The targets of these rules are replica vaults located on specific replica sites. For example, consider two replica sites named site1 and site2. The engineers at site1 are collaborating on the generation of the design models of a part, while the personnel at site2 will sell the part. The sales personnel do not need to see the incomplete designs for the part, so two vaulting rules would be appropriate:
1. WTPart, all-states, collab-domain > Vault_on_site1
2. WTPart, complete, collab-domain > Vault_on_site2

These rules allow the engineers to get the content for all life cycle states of the part, while the sales personnel see the content only after the part is complete.

Windchill domains can be created in a hierarchical fashion, with some domains being children of other domains. It is important to note that a domain does not inherit the replication rules of its parent domain. Replication rules must be explicitly defined at each level of a domain hierarchy.

Content Replication can be scheduled by creating a schedule item for a replica vault. A schedule item describes an operation, for example, "Replicate to vault A at 4:00 PM." Each schedule item is independent, so one needs to be created for each replica vault. By creating a schedule item, you can assure that synchronization occurs on a regular basis. Schedule items are independent of each other. For example, you can force a replication to happen sooner than it was originally scheduled to happen by creating a schedule item on the vault in question in the immediate-once mode. Schedule items are maintained on the master site.

The Windchill master contains information about the files that exist on the replica site, and no replication occurs unless content is missing and needs to be replicated. Each Windchill user can specify a preferred site from which to attempt downloading of replicated data. If the data requested does not exist at the preferred site, the data is downloaded from the master site where it resides in an Oracle database or an external vault. If data is not available at a replica site, it is because the rules controlling content for the vault do not include the data or the data has not yet been replicated to the replica site.

Content Replication is ready to run as soon as you have set up a master site, entered the necessary configuration information for the replica site, exported the public key of the master site to a file, and copied the file to the replica site.

The major topics explained in Windchill Content Replication documentation are the following:

- Setting up the sites -- Setting up master and replica sites and creating and placing key files are the first tasks.
- Editing the wt.properties file -- The properties control the behavior of the master site and replica site.
- Creating hosts, vaults, and folders -- In the second stage of setting up the system you define hosts, vaults, and folders.
- Mounting Folders -- Mounting folders makes them available for replication.
- Final Site Setup Activities -- Several small configuration tasks are required to make the sites completely usable.
• Creating Content Replication rules and schedule items -- After the sites are set up, the usual sequence is to create Content Replication rules to control the types of content that is replicated to replica sites. These rules can be created or modified at any time. You create scheduling items to specify when replication from master sites to replica sites occurs. These rules can be created or modified at any time. Content Replication is a resource-intensive process.

• Troubleshooting -- If configuration included errors, examining log files can help in finding the errors.

• Performance improvement -- The Content Cache Server and Local Upload are features that improve Content Replication performance experience. Small configuration tasks are needed to gain the benefits of these features.

This documentation describes both the properties and services in the wt.properties file that are relevant to Windchill Content Replication. If replication configuration contains an error, log files created by the services provide information for troubleshooting. The log files show all the interactions between master sites and replica sites. In the case of some errors, the log files list suggestions to solve the problem.

You can accomplish many of the operations explained in this chapter through a command line interface that is explained in another chapter, Configuring External File Vaulting or Replication With FvLoader.

**Setting Up Sites and Keys**

Starting file vault service generates a site named Master, and modifying the name or URL of that site is a possible, but not normally needed procedure. This procedure is discussed in the following section, Modifying the Local Site.

You create sites that relate to the local site as its master site or as its replica sites. Creating a master site is explained in the later section, Creating the Master Site. Then you generate replica sites as described in the later section, Creating a Replica Site. After creating the replica sites, you make public keys from the master site available to each replica site, as explained in the later section, Creating and Placing Security Keys.

The installation of replica sites is discussed in the later section, About Installing Two Types of Replica Sites and two sections that follow it, Installing a Full-scale Replica Site and Installing a Lightweight Replica Site.

After master and replica sites have been installed, they require some configuration. The configuration of replica sites includes importing keys. Site configuration is discussed in the later sections Master Configuration and Replica Configuration.
Modifying the Local Site

When file vault service starts, a default master site is automatically generated, so you do not need to create a master site. You can modify the parameters of the automatically generated master site. The automatically generated master site has the name *Master* and its URL is the value of the property \texttt{wt.httpgw.url.anonymous} in the \texttt{wt.properties} file. In the Site Management window, the site name for the Windchill site to which you are currently connected is followed by the label (This Installation). You do not select roles such as Master or Replica for the site to which you are currently connected.

Windchill software insures that the automatically generated site labelled *This Installation* can continue serving its role in the event of a change in the value of the property \texttt{wt.httpgw.url.anonymous} in the \texttt{wt.properties} file. If the value changes, Windchill assigns the new URL to it, and a warning message is printed on the master site console.

After vaulting or replication is functioning, you can update name or URL data for any site, but you should consider the results of this action which could result in configuration problems.

Creating the Master Site

Execute the following steps on a machine that will be a master site server to modify a master site.

1. In the Windchill home window click **Administrator** to display the **Administrator** window.
2. Click **Replication Administrator** to display the **Replication Administrator** window.
3. Click **Site Management** to display the **Site Management** window. Note that you are creating a site that will play the role of master relative to the local site. The roles of sites in Content Replication are identified by plus signs (+) and minus signs (-) in the **M** and **R** columns. The local site is identified by two minuses. A site that is a replica for the local site displays a plus in the **R**
column. A site that is a master for the local site displays a plus in the M column.

4. Click **Create** to display the **Create New Site** window.
5. Enter the values for the master site data.
   - **Site Name** -- The site name must be unique. The string is case-insensitive and cannot include a space.
   - **URL** -- The URL of the anonymous gateway for the Windchill system that will use this installation as a replica. The URL is the value of the property named wt.httpgw.url.anonymous in the wt.properties file. The anonymous URL must have anonymous access specified.
   - **Site Type** -- Select **Master**. A site can play the role of replica or master or both. Leaving the boxes for both these roles clear disables the site except in the case of the local site.
   - **Description** -- You can enter a description of the master site.

6. Click **OK**. The **Site Management** window displays a row showing the site's data, URL, and a space for the initial date of the access key after it is generated. The **Vault Configuration** window displays an icon for the site in its left-hand pane. You can display the **Vault Configuration** window by clicking **Administrator** in the Windchill home page, clicking **Replication Administrator**, and clicking **Vault Configuration**.

**Creating a Replica Site**

Execute the following steps to create a replica site.

1. In the Windchill home window click **Administrator** to display the **Administrator** window.

2. Click **Replication Administrator** to display the **Replication Administrator** window.

3. Click **Site Management** to display the **Site Management** window. Note that you are creating a site that will play the role of replica relative to the local site. The roles of sites in Content Replication are identified by plus signs (+) and minus signs (-) in the M and R columns. The local site is identified by two minuses. A site that is a replica for the local site displays a plus in the R
column. A site that is a master for the local site displays a plus in the **M** column.
4. Click **Create** to display the **Create New Site** window.

![Create New Site Window]

5. Enter the values for the replica site data.

   - **Site Name** -- The string is case-insensitive and cannot include a space. A replica site must be known by the same name to all master sites.
   
   - **URL** -- Enter the URL of the replica site anonymous gateway URL. The URL is the value of the property named wt.httpgw.url.anonymous in the wt.properties file.
   
   - **Site Type** -- Select the **Replica** check box. A site can play the role of replica or master or both. Leaving the boxes for both these roles clear disables the site except in the case of the local site.
   
   - **Description** -- You can enter a description of the replica site.

6. Click **OK**. The **Site Management** window displays a row showing the site's data and URL. The **Vault Configuration** window displays an icon for the site in its left pane. You can display the **Vault Configuration** window by clicking **Administrator** in the Windchill home page, clicking **Replication Administrator**, and clicking **Vault Configuration**.

You can display a dialog box for modifying the data for an existing replica site by selecting the site and clicking **Update**.
Creating and Placing Security Keys

To create and place keys, execute the following steps.

1. In the Windchill home window click Administrator to display the Administrator window.

2. Click Replication Administrator to display the Replication Administrator window.

3. Click Site Administrator to display the Site Management window. Master sites show a plus (+) sign in the M column. Selecting a master site activates the Update Key button, and after a key is generated, the Export Key button becomes active.

4. Select the master site.

5. Click Update Key to initiate the creation of the key. Generating the key requires several seconds.

6. When the key is complete, click Export Key.

7. In the Save As dialog box, name the key, navigate to a storage location for the key, and save it as a file.

8. Transfer key to the replica site.

9. Make the key file known to the replica site by editing the replica site's wt.properties file. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

   If only replica site services are running, edit the parameter
   wt.intersvrcom.masterSite.<number of master site>=<URL of master site>,<location of key file>
   to reflect correct values. Multiple instances of this property can exist to serve different master sites, and each instance is distinguished by a separate number. The numbers are integers beginning with 1 and they are sequential. A comma separates the master site URL from the location of its key file. If other services are running in addition to replica services, make changes by performing site administration on the replica site. After transferring the exported key file to each replica site, import it to each. Click Import Key in the Site Management window.

10. Stop and restart the replica site method server.

Whenever you update the public key, it must be copied again to every replica site, and you must restart the replica site method server. If replica sites do not receive the correct public key, replication will not occur. Concealing the public key is unnecessary, but permissions should prevent its alteration.
About Installing Two Types of Replica Sites

Two types of replica sites to allow you to maximize performance and the use of resources: the full-scale replica site and the lightweight replica site:

- Full-scale replica site -- A complete Windchill installation, with database access, that can also manage replicated content from another Windchill installation.
- Lightweight replica site -- A partial Windchill installation, without database access, which can only manage replicated content from another Windchill site.

The installation of a full-scale replica site is similar to a regular Windchill installation; but you need to pay attention to one setting and note one piece of data that is displayed.

The installation of a lightweight replica site differs from the installation of a full-scale replica site in several ways: you do not need to work with a database tool, you are concerned with more variables, and some properties settings must be present.

The installation and configuration of both full-scale replica site and lightweight replica site are detailed in the following sections.

Installing a Full-scale Replica Site

Follow the installation process defined in the Windchill Installation and Configuration Guide — Windchill. Do not select the checkbox Disable All Non-replication Services during the installation, which results in a server that serves as a lightweight replica site. Note the data in the box Name of Site Hosting Data for Other Services, because that information will be needed to connect the master sites and replica sites.

Before using the full-scale replica site, you can create folders or confirm the existence of folders that will serve as Replica Folders.

See the section Replica Configuration later in this document for information on configuring the full-scale replica site.

Installing a Lightweight Replica Site

In lightweight mode, the replica Windchill server will not have access an Oracle instance and will be running with a minimal set of services. Only those services required for Content Replication will be started. The replica site will requires the installation of a servlet engine and web server; it is recommended that this is completed prior to the ultralight server install.

To set up a lightweight Content Replication server, you must modify some of the properties during the execution of Pro/Setup. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties
with the xconfmanager utility, which is discussed elsewhere in this guide. The installation process follows:

1. From the Windchill Foundation CD, run setup.exe.

2. Select Windchill and PDM Foundation only. As a lightweight replica server does not require a database, you do not need to install the Windchill Database Tool.

3. Continue as a normal installation, selecting Configure Replication Services on the Optional Configuration Steps page. The selected or non-selected status of options other than Configure Replication Services in the following graphic is irrelevant to the creation of the replica site.

Because a lightweight Windchill server will not connect to a database, clear the Configure Oracle Server Properties checkbox may be cleared.

4. On the Replication Services page, select the following options:
   – Replicate data from this server' should not be selected
   – Select Host replica data for other servers.

   Set the value of Name of site hosting replica data to indicate this replica site. For example the machine name for the replica server is often used. This value will be inserted into the property wt.fv.replica.mysite in the wt.properties file.

   Make a note of the Name of site hosting replica data value. This information is required when configuring the master to site to recognize this Windchill installation as a replica site.

   – Select Disable all non-replication services. This is not selected by default. Making this selection ensures that only a minimum set of Windchill services will be configured in the wt.properties file.
5. Continue with the installation, as documented in the Windchill Installation and Configuration Guide.

After the installation is complete, edit the wt.properties file. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide. Rather than the usual 67(+) services only the 6 services pertaining to replication will have been configured.

Verify that the services section contains the following:

```
wt.services.service.1=wt.intersvrcom.InterSvrComService/wt.intersvrcom.StandardInterSvrComService
wt.services.service.2=wt.fv.replica.ReplicaService/wt.fv.replica.StandardReplicaService
wt.services.service.3=wt.fv.replica.ReplicaServiceSvr/wt.fv.replica.StandardReplicaService
wt.services.service.4=wt.wrmf.delivery.ShippingService/wt.wrmf.delivery.StandardShippingService
wt.services.service.5=wt.wrmf.delivery.ReceiverService/wt.wrmf.delivery.StandardReceiverService
wt.services.service.6=wt.wrmf.transport.GenericTransportService/wt.wrmf.transport.StandardGenericTransportService
```
The Method Server and Server Manager should now launch successfully. The POM messages normally seen when the Method Server starts will not be displayed and registering with the Server Manager should be significantly quicker than in a full Windchill installation.

The Lightweight site should display the Windchill Homepage fully if accessed via a browser, but all actions requiring a database should fail with an appropriate error message.

See the section Replica Configuration later in this document for information on configuring the lightweight replica site.

Master Configuration

You should already have created the replica site through the Site Administration window. This was discussed in the earlier section, Creating a Replica Site.

You should already have generated the key from the Master site and saved it for later use. This was already discussed in the earlier section, Creating and Placing Security Keys.

To configure the master site, perform the following steps:

1. In the Vault Configuration window, create hosts, vaults, and folders.
2. Mount folders to the directories specified in the installation section.
3. Enable the folders.

Replica Configuration

To configure a full-scale replica site PTC recommends that you perform the configuration through the graphical user interface, as explained in the following section Configuring a Full-scale Replica Site. You could perform this configuration by adding a property with the xconfmanager utility, which is discussed elsewhere in this guide, but that would be less convenient.

To configure a lightweight replica site, you must add a property. This configuration is discussed in the following section, Configuring a Lightweight Replica site.

Configuring a Full-scale Replica Site

Configuring a full-scale replica site through the graphical user interface is similar to configuring a master site.

See the section earlier in this document, Creating and Placing Security Keys, to make sure that you have performed the required actions regarding creation and export of keys. Make sure you perform the actions for the master site that are described in the preceding section Master Configuration before you start using the master site.
1. In the Windchill home window, click Administrator to display the Administrator window.

2. Click Replication Administrator to display the Replication Administrator window.

3. Click Site Administrator to display the Site Management window. The roles of sites in Content Replication are identified by plus signs (+) and minus signs (-) in the M and R columns. The local site is identified by two minuses. A site that is a replica for the local site displays a plus in the R column. A site that is a master for the local site displays a plus in the M column.

4. Create a master site using the instructions in the previous section in this document, Creating the Master Site, if you have not already done so.

Selecting the master site that in the Site Management window makes the Import Key button active. Click Import Key in the Site Management window to import the key to the local site that is serving as a full-scale replica site.

**Configuring a Lightweight Replica site**

See the section earlier in this document, Creating and Placing Security Keys, to make sure that you have performed the required actions regarding keys. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

To configure the replica site, perform the following task:

In Windchill\codebase\wt.properties, add the property
`wt.intersvrcom.masterSite.1` to reference the master site. The structure is as follows:

`wt.intersvrcom.masterSite.1=<masterGatewayUrl>,<master public key location>`

For example, If the master gateway Url, which can be obtained from Site Administration window, is
`http://abcdef.com:9999/Windchill/servlet/WindchillGW`, and the location of the master key on replica is `C:\masterConfig\master.key`, the resultant string is:

`wt.intersvrcom.masterSite.1=http://abcdef.com:9999/Windchill/servlet/WindchillGW,C:\masterConfig\master.key`

**Setup Check**

To check the setup of either type of replica site, perform the following steps:

1. Enable verbose for both fv and fv.master packages on master site and fv.replica package on replica site.
2. Restart the replica site MethodServer. Right after start-up, in the logs, you should see a line stating that replica has requested configuration from Master. Several lines below, there should be a response message specifying received configuration. Do a sanity check on the configuration.

3. Restart the master site MethodServer. Right after start-up, in the logs, you should see a line stating that master site has attempted to refresh the configuration of the replica site. Check the replica site MethodServer.log to see that the configuration was actually received.

Replication Security

To enable secure transactions, Content Replication requires replication sites to share a common, trusted certificate authority (CA). If a client experiences a Java secure socket link exception (for example, "javax.net.ssl.SSLException: untrusted server cert chain"), the client needs to import the CA of the server to which it is making a request. See the section on Importing Certificates into Sites for more information.

Importing Certificates into Sites

Use the following commands to import certificates into master and replica sites:

```
keytool -import -alias someAliasName -file path/to.certificateAuthority.cert
-storetype jks -keystore /path/to/keystore.jks
```

certificateAuthority.cert is the certificate of the certificate authority (CA), not the web server. In the case of a self-signed web certificate, the CA and the web server are the same.

keystore.jks is the file that the trusted CA will be imported into. The Java secure socket extension (JSSE) provider has a truststore located at:

```
$JAVA_HOME/jre/lib/security/jssecacerts
```

The commands listed above install the CA to be trusted by all invocations of the given virtual machine. Alternatively, the CA can be imported into any file, and then referenced on the command line.

The argument to java to use a trust store file is:

```
-Djavax.net.ssl.truststore=fileName
```

For example:

```
keytool -import -alias Acme_CA -file /tmp/acme_ca.cert
-storetype jks -keystore /home/jlk/wgm_for_proe/conf/cacerts.jks
java -classpath /home/jlk/wgm_for_proe/lib/foo.jar:/...
   -Djavax.net.ssl.trustStore=/home/jlk/wgm_for_proe/conf/cacerts.jks
   com.ptc.foo.jar
```
Editing the \texttt{wt.properties} File

Setting properties in the \texttt{wt.properties} file is an essential activity in configuring Content Replication. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the \texttt{xconfmanager} utility, which is discussed elsewhere in this guide.

The following list describes the properties that are most relevant to Content Replication, categorized by their area of influence. All of these properties are present in all \texttt{wt.properties} files present on master and replica servers, but many of the properties are relevant only to master site or replica site servers. Some of the properties control the placement or behavior of log files, which can be important in troubleshooting. If you have not planned the details of the Content Replication sites, you may be unable to provide correct values for some properties until you have completed the setup procedures.

Master and Replica Properties

The following table shows the properties that are set on master sites, replica sites, and both.

<table>
<thead>
<tr>
<th>Property</th>
<th>Master</th>
<th>Replica</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{wt.fv.master.verboseProperties}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.master.verbose}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.master.log.enabled}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.master.log.append}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.master.log.filename}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.master.replicateQuerySize}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>\texttt{wt.fv.replica.verbose}</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>\texttt{wt.fv.replica.log.enabled}</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>\texttt{wt.fv.replica.log.append}</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>\texttt{wt.fv.replica.log.filename}</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>\texttt{wt.fv.replicationFileSizeThreshold}</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

\textbf{Note:} The \texttt{wt.fv.replicationFileSizeThreshold} may not appear in the file, but it is nonetheless existing and effective. For an explanation of this property, see \textit{Content Replication and Windchill Visualization Service} on page 4-19.
### Basic Properties

The following properties affect Content Replication and other functions as well, unlike the properties in the preceding table, which have a more limited effect. For example, the roles of sender and receiver are related to content and to the IntraLink-to-Windchill gateway. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the `xconfmanager` utility, which is discussed elsewhere in this guide.

<table>
<thead>
<tr>
<th>Property</th>
<th>Sen-der</th>
<th>Receiv-er</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.intersvrcom.verbose</td>
<td>X</td>
<td>X</td>
<td>Set in installation. Do not alter. default: false</td>
</tr>
<tr>
<td>wt.intersvrcom.ultraLight</td>
<td>X</td>
<td>X</td>
<td>Set in installation. Do not alter. default: true</td>
</tr>
<tr>
<td>wt.intersvrcom.security.graceTimePeriod</td>
<td>X</td>
<td>X</td>
<td>If the time difference between the time that the URL is signed at the sender site and the time that it is received at the receiver site is more than N seconds, the signed URL will be invalid. default: 300 (seconds)</td>
</tr>
<tr>
<td>wt.intersvrcom.security.URLAuthentication</td>
<td>X</td>
<td>X</td>
<td>Do not set to false unless in debug mode. default: true</td>
</tr>
<tr>
<td>wt.intersvrcom.security.useProxyForClients</td>
<td>X</td>
<td></td>
<td>This value must be set to true if the sender connects internet through a proxy. default: true</td>
</tr>
<tr>
<td>wt.wrmf.verbose</td>
<td>X</td>
<td></td>
<td>default: false</td>
</tr>
<tr>
<td>wt.wrmf.delivery.deleteDeliveredItem</td>
<td></td>
<td>X</td>
<td>If set to true, all delivered Shipping Item will be deleted from the database. default: true</td>
</tr>
<tr>
<td>wt.wrmf.transport.httptransport.supportInterruption</td>
<td>X</td>
<td></td>
<td>If true, upload or download will resume the http connection due to IOExceptions such as temporarily unavailable networking problems. This is useful for uploading or downloading large content files. default: true</td>
</tr>
<tr>
<td>wt.wrmf.transport.outbox.pipe.&lt;1 or 2 or 3&gt;</td>
<td>X</td>
<td></td>
<td>Sets a transportation type. See the values of this property in the <code>wt.properties</code> file to for the correspondence of integers with transportation types.</td>
</tr>
</tbody>
</table>
### Maximum Queue Values

The following table shows properties that set the maximum number of process or schedule queues performed by the queue service. Depending on your requirements, you may need to reset these properties from their default values. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

In theory, the values of these properties have no upper limit, but increasing their size decreases performance. If you see errors that are presented as **Max ProcessQueues Exceeded**, increase the value of the `wt.queue.max.processQueues` property. If you see errors that are presented as **Max ScheduleQueues Exceeded**, increase the value of the `wt.queue.max.scheduleQueues` property.

Both types of errors are displayed in the method server log. Most errors of the **Max ScheduleQueues Exceeded** type appear in graphical messages, while most errors of the **ProcessQueues Exceeded** type do not appear as graphical messages.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wt.queue.max.processQueues</code></td>
<td>Sets the maximum number of process queues that the queue service creates before throwing an exception. Default is 15.</td>
</tr>
<tr>
<td><code>wt.queue.max.scheduleQueues</code></td>
<td>Sets the maximum number of schedule queues that the queue service creates before throwing an exception. Default is 15.</td>
</tr>
</tbody>
</table>

For more information on background queues see the chapter, *Configuring and Administering Background Queues*, later in this guide.

### Content Replication and Windchill Visualization Service

You need to configure properties and to specify rules to make Content Replication function with the Windchill Visualization Service. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.
Configuring Properties

To enable Content Replication for viewables it is recommended that the following properties be set. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

• In wt.properties:

  wt.fv.replicationFileSizeThreshold=0

  The wt.fv.replicationFileSizeThreshold in the wt.properties file sets the minimum size file that Content Replication will handle. The value of this property sets a number of bytes. The property’s default value is 10K, which could exclude very small files.

• In wvs.properties:

  publish.service.enabled=true
  wvs.enabled=true

  The default value of publish.service.enabled property is false. You must change the default value, or the replication of viewables will fail. The default value of the wvs.enabled property is true.

To Replicate WVS Viewables

To specify that the viewables are replicated, you select the DerivedImage class or the WTMarkUp class in the Content Replication Rules window while specifying the Content Replication rules. You can set up separate rules for each class, and the viewables will be replicated.

1. You begin to display the Content Replication Rules window by clicking the Content Replication Rules icon on the Replication Administrator page. This displays the Administrative Domains selection window.

2. You select a domain in the Administrative Domains selection window. Full domain paths are shown in the Administrative Domains selection window, beginning with a root domain represented by a slash (/). Click Update to display the Domain Vaulting Rules window for the selected domain. The Replica Vaulting tab is selected by default.
3. To create a rule, click **Create** to display the Content Replication Rules window. In this window you create rules, each consisting of the DerivedImage class or the WTMarkUp class, one state, one site, and one replica vault. When you select a site from the pull-down menu, only the vaults for that site are displayed in the window.

### Replication and Compression

By default, Content Replication uses compression. To stop the compression, you set false as the value of the property `wt.intersvrcom.transport.site` in the `wt.properties` file. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the `xconfmanager` utility, which is discussed elsewhere in this guide.

The following line shows the syntax for setting the property false:

```properties
wt.intersvrcom.transport.site.<Site Name>.useGzip=false
```

For example if site name is `replica1`, the following line in the `wt.properties` file would configure replication to the site without compression:

```properties
wt.intersvrcom.transport.site.replica1.useGzip=false
```
Improving Content Replication Performance

Windchill offers two technologies enabled by the same option that accelerate the handling of content data and expedite collaborative development.

- Local Upload -- Places a user's uploaded content in the local cache vault as an intermediate location prior to transfer to a master site upon checkin.
- Content Cache Server (CCS) -- Creates a location for rapid access to frequently requested content data.

Both technologies depend on a designated cache vault in the replica site. These technologies are transparent to the Windchill user and can be incorporated in applications. The technologies deliver the following benefits:

- Faster checkin for the user
- Faster and earlier access to cache content data for users with shared download preference
- Availability of data to all users at all times
- Ability to determine the checkin status on the master site
- Data mirroring on cache vault site to safeguard data against failure
- Conformation of replica site structure to rules
- Data searchable by all users after indexing on the master site
- Java Bean for easy incorporation of Local Upload functionality in applications
**How the Local Cache Works**

The explanation of the local cache vault in this section refers to the following graphic. The characters in the graphic are keys to explanations in the list that follows the graphic.

A. Vault on master site.

E. Local cache vault on cache server site.

E2. A physical path on the file system corresponding to a folder intended for reading and writing in the local cache vault.

E1. A physical path on the file system corresponding to a folder intended for mirroring in the local cache vault. Mirrored paths for folders cannot be used for downloads from the replica site.

G1 and G2. Hosts whose users share the same site preference for downloads. The site preference is set to the replica site that also contains the local cache vault.

G3. A host whose users have the site preference for downloads set to a replica site that does not contain a cache vault.

g1e2 and g2e2. Mounts to the readable folder in the local cache vault.
g1e1 and g2e1. Mounts to the mirroring folder in the local cache vault.

H. A vault that is not a local cache vault that is in another site that is the preferred download site for the user of host G3.

sA. Master site for sites sE and sH.

sE. Replica site that includes a local cache vault and that is the preferred site for download for users of hosts G1 and G2.

sH. Replica site that does not include a local cache vault and that is the preferred site for download for the user of host G3.

The time required for a user's checkin, create, read, and update processes associated with the upload and download of files is reduced because these interactions involve data on the rapidly accessible cache vault, rather than the more slowly accessible vaults on the master site. In the absence of an earlier request for them, the content files are replicated to the master site under the control of a replication schedule. For example, when a user of host G1 checks in data, the checked in copy is in local cache vault E rather than master site vault A, which would be the checkin vault if the local cache enhancements were not enabled. The data will be copied from vault E to vault A, either when an applicable replication schedule becomes active or when a request for the data arrives at site sA.

Users who have a content cache preference set to the replica site that holds the local cache vault can access data placed there more rapidly than if they could access it only from the more slowly accessible master site. For example, a user on host G2 who accesses content data checked in by the user on host G1 deals with local cache vault E as the source of the content data, rather than the less rapidly accessible master site vault A. Not only is time for access reduced, in addition the data is available earlier due to the reduced time for checkin to local cache vault E relative to the longer time that a checkin to master site sA would require.

If the master site receives a request for data that exists only in the local cache vault, the data moves immediately to the master site to enable it to serve the request. For example, if the user on G3 requests content data that exists in vault E and does not exist in vault A, the content is copied to A, and the data is then downloaded to G3. The content data is not transferred automatically to site sH unless an appropriate replication rule is created to transfer the data.

The transparency of the technology to the Windchill user may create a need for clarification about whether data checked in to the local cache vault E has been copied to the master site sA. A utility that runs on the master sA site supplies information about files not yet copied to site sA. The utility is discussed later in this document as the "Utility to Assist Backups."

Maintaining two copies of data within the local cache vault protects it from loss or damage. Each local cache vault folder accessed by read and write operations can be associated with a folder that mirrors it when mount paths for both folders are specified in the same entry during configuration. If the folder accessed for read and write operations cannot be read, the contents of the mirroring folder can be
copied to the readable folder so that the read operation can continue. A later section of this guide, Establishing Mirroring in the Local Cache Vault, explains the technique for establishing mirroring. For example, the mount path g2e2 associates the read folder E2 with the host G2, while the mount path g2e1 associates the mirroring folder E1 with the host G2.

When content has been replicated from the local cache vault to the master, it exists in both locations. If its structure in the local cache vault violates a rule, the violation is corrected when the rule becomes active.

Indexing is the collecting of keywords from data to make the data searchable. Data in the local cache is not indexed. Data is indexed as soon as it moves from the local cache vault to the master site.

A download and upload Java Bean is available to implement the feature in applications. The *Windchill Application Developer's Guide* describes this bean.

### Creating a Local Cache Vault

You can specify one vault in a site to perform the local cache vault role. To enable the local cache function in a vault, select the **Designated for Cache** checkbox while creating or updating a vault in the replica site. When you are creating a vault, the checkbox appears in the **New Vault** window. To update a vault, double-click its icon in the **Vault Configuration** window or select its icon and select **Object > Update**.

### Establishing Mirroring in the Local Cache Vault

When you are defining mounts that associate hosts and folders on the replica site that holds the local cache vault, you can create a backup to protect against loss of data. Each local cache vault folder accessed by write operations can be associated with a backup storage location that mirrors it when the mount paths for folders and the backup storage location are specified in the same entry during configuration. If loss of data occurs in a folder that is read, you can copy the data in the backup storage location to the folder that is read.

To configure the backup, perform the following steps:

1. In the **Vault Configuration** window open the cabinet that holds mounts.
2. Select the folder that will be readable.
3. Click **Object > Mount** to display the **New Mount** dialog box.
4. Specify two paths to different folders separated by a semicolon (;) in the **Path** box. The storage location specified by the first path will be the folder that is written to and read under normal conditions. The storage location identified by the second path will be used for mirroring the content.
5. Duplicate the mounting on all the hosts in the replica site, providing paths to the same physical folders.
6. Select the folder in the Vault Configuration window and click Object > Toggle Enabled.

Setting the Preferred Content Cache Site

To benefit from the technology described in this chapter, users must set their content cache site preference to the replica site that includes the local cache vault. Because this is a personal preference that can easily be changed, explaining the benefits and location of the local cache vault to users may be advisable. The preferred content cache site is the same as the replica site preference for Content Replication. The Windchill User's Guide explains setting this preference.

Scheduling Moving Data from Local Cache to the Master Site

You can schedule the replication of content from the local cache vault to the master vault. If you do not schedule, the data is copied to the master site when a request for the data is made to the master site. If data is not copied to the master site, it is not indexed and it is therefore not searchable. The data is not automatically backed up in a central location, but you can schedule Content Replication for the cache vault. Scheduling of data in the cache vault is the same as scheduling for other content data, except that you do not need to create a rule for moving the data from the cache server to the master. See Administering Content Replication in this guide for an explanation of scheduling.

Utility to Assist Backups

You can run a utility at the master site to distinguish between files that are currently copied on both the master site and the replica site and other files that have been checked in to the replica site but have not been copied to the master site. The utility is intended to guide backup processes. You invoke the utility from the command line with the following syntax:

```
windchill -cp <path_to_codebase>
wt.fv.uploadtocache.CCS_BackupFilesList
```

The `<path_to_codebase>` is the path to the codebase for the master site.

The utilities output is an ASCII file in the log directory that lists files on replica sites that are not on the master site. Files are listed by site and by folder within each site. The output file's name has the following syntax:

```
ccs_backup_<timestamp>.log
```

Log Files

The standard master site and replica site log files show the interactions between master and replica sites. See Editing the wt.properties File in this chapter for an explanation of the log file properties.
## Configuring External File Vaulting or Replication With FvLoader

### Overview of Configuration With FvLoader
- Page 5-2

### Configuring External File Vaults or Rules
- Page 5-3

### Supporting Replica Site Vaulting
- Page 5-3

### Removing External File Vaulting Rules or Replication Rules
- Page 5-4

### Listing Domains
- Page 5-5

### Listing Vaulting Rules
- Page 5-7

### Supporting Local Replication
- Page 5-8
Overview of Configuration With FvLoader

PTC supplies a class that you can use as a utility for the following purposes. The utility is known as FvLoader, which is a shortened version of File Vault Object Loader. Some FvLoader actions are controlled by command line arguments, and other actions are controlled by data in files.

- to create and configure an external file vault and vaulting rules -- accomplished through data specified in a file. You can create vaults, folders, hosts, mounts, and rules, and you can enable the folders. See Configuring External File Vaults or Rules.

- to support replica site vaulting -- accomplished through data specified in a file. You can create vaults, folders, hosts, mounts, and rules, and you can enable the folders. See Supporting Replica Site Vaulting.

- to remove external file vaulting rules or replication rules -- accomplished through data specified in a file. See Removing External File Vaulting Rules or Replication Rules. You may need to use FvLoader to list domains to perform the removal actions efficiently.

- listing domains -- accomplished through command line. This is a two-step process. See Listing Domains.

- listing vaulting policy rules -- accomplished through command line. You can use the output for batch deletion or recreation of policy rules. See Listing Vaulting Rules.

- supporting local replication -- -- This is a procedure that provides accelerated Content Replication through a series of steps. The actions to perform local replication require a file to specify FvLoader’s action, and it is probable that most users will invoke FvLoader with command line to get information to set up the local replication. The steps relocate content to the local site, move content to the destination replica site manually, and update the database to reflect the move. See Supporting Local Replication.

You can create data in files to specify the action of FvLoader in two ways.

- Modify fvloader.csv file in the directory /loadFiles directory to specify the task. Running FvLoader with no arguments loads data from this file. The command that you type in the command window is the following:

  java wt.fv.FvLoader

- Create your own comma-separated value (.csv) file. The command that you type in the command window is the following:

  java wt.fv.FvLoader <Full File Name>

The syntax is the same for the fvloader.csv file or the .csv file that you write.
Configuring External File Vaults or Rules

The following are the arguments that you can supply for the fvloader.csv file or the .csv file that you write to configure external file vaults or rules:

- **V,vaultName** -- Creates external file vault with the name vaultName.
- **H,hostName** -- Creates a file vault host with the host name hostName.
- **F,folderName,vaultName** -- Create file vault folder with the name folderName and attaches that file vault folder to the file vault with the name vaultName.
- **M,folderName,hostName,path,local(0/1)** -- Create a file vault mount between the file vault with the name Folder folderName and the file vault host with the name hostName. The mount has the following characteristics:
  - path given in the path argument
  - local flag turned off by the value 0 or turned on by the value 1
- **R,vaultName,fullClassName,fullDomainPath,lifeCycleStateName** -- Creates a file vaulting policy rule that concerns the following:
  - the file vault with the name vaultName
  - the class with the name fullClassName -- Only classes listed in the graphical user interface for creating rule may be included in the .csv file. Abstract classes that are content holders are not permitted. If you use only the classes displayed in the graphical interface for making rules, you will obey this guideline.
  - the domain with the full external path fullDomainPath. For an explanation of where to get the fullDomainPath please see the section, Listing Domains.
  - the life cycle state with the name lifeCycleStateName.
- **FE,folderName** -- Enables the folder with name folderName. If a folder does not have at least one mount, it should not be enabled.

Note: Life cycles states are case-sensitive. Consequently, verify how a life cycle state name is written, including the case used, before writing the name in the FvLoader file.

Supporting Replica Site Vaulting

The following are the arguments that you can supply for the fvloader.csv file or the .csv file that you write to support replica site vaulting:

- **RV,vaultName,replicaSiteName** -- Creates replica file vault with the name vaultName and attaches it to the replica site with replicaSiteName.
- **RVE,vaultName** -- Enables replica file vault with the name vaultName.
• RF, folderName, replicaVaultName -- Creates a replica file vault folder with the name folderName and attaches that replica file vault folder to the replica file vault with the name replicaVaultName.

• RH, hostName, siteName -- Creates a file vault host with the host name hostname and attaches it to the replica site with the name siteName.

• RR, replicaVaultName, fullClassName, fullDomainPath, lifeCycleStateName -- Creates a Content Replication policy rule that concerns the following:
  – the replica file vault with the name replicaVaultName
  – the class with the name fullClassName -- Only classes listed in the graphical user interface for creating rules may be included in the .csv file. Abstract classes that are content holders are not permitted. If you use only the classes displayed in the graphical interface for making rules, you will obey this guideline.
  – the domain with the full external path domainPath. For an explanation of where to get the fullDomainPath please see the section, Listing Domains.
  – the life cycle state with the name lifeCycleStateName

• RFE, replicaFolderName -- Enables the replica file folder with name replicaFolderName. If a replica file folder does not have at least one mount, it should not be enabled.

• RM, replicaFolderName, hostName, path -- Creates a file vault mount between the replica file vault folder with the name replicaFolderName and the file vault host with the name hostName. The mount has the following characteristic:
  – path given in the path argument

Note: Life cycles states are case-sensitive, and the use of lowercase letters could corrupt the rules table. Consequently, use only capital letters for life cycle states to load vaulting rules with FvLoader.

Removing External File Vaulting Rules or Replication Rules

The following are the arguments that you can supply for the fvloader.csv file or the .csv file that you write to remove external file vaulting rules or replication rules.

• RemoveReplicaR, replicaVaultName, fullClassName, fullDomainPath, lifeCycleStateName -- Removes an existing Content Replication rule (same arguments as for rule creation)
  – the replica file vault with the name replicaVaultName
  – the class with the name fullClassName -- Only classes listed in the graphical user interface for creating rules may be included in the .csv file. Abstract classes that are content holders are not permitted. If you use only
the classes displayed in the graphical interface for making rules, you will obey this guideline.

- the domain with the full external path domainPath. For an explanation of where to get the fullDomainPath please see the section, Listing Domains.

- the life cycle state with the name lifeCycleStateName

• RemoveLocalR,vaultName,fullClassName,fullDomainPath,lifeCycleStateName -- Removes an existing external vaulting rule (same arguments as for rule creation)

- the file vault with the name vaultName,

- the class with the name fullClassName -- Only classes listed in the graphical user interface for creating rule may be included in the .csv file. Abstract classes that are content holders are not permitted. If you use only the classes displayed in the graphical interface for making rules, you will obey this guideline.

- the domain with the full external path fullDomainPath. For an explanation of where to get the fullDomainPath please see the section, Listing Domains.

- the life cycle state with the name lifeCycleStateName.

**Listing Domains**

To list containers and domains several command line arguments can be appended to the command java.wt.fv.FvLoader. They may be invoked by typing the following syntax at the command prompt:

java.wt.fv.FvLoader -argument [options]

Listing containers or domains requires two invocations of FvLoader, which are discussed in the next two sub-topics.

**The -listContainers Argument Obtains Data**

The first FvLoader invocation produces output that includes the container and domain information. Use the argument -listContainers to print a list of external container paths to the console. The output may be redirected to a file using piping. The output may be used only as input for the -listDomains argument, which is explained after the following example.

Example:

C:\> java wt.fv.FvLoader - listContainers
/ /wt.inf.container.OrgContainer=PTC
/ /wt.inf.container.OrgContainer=PTC/wt.inf.library.WTLibrary=Windchill PDM
C:\>
The -listDomains Argument Presents Data

Use the argument -listDomains to accept the output of -listContainers in order to list domains and format the list. A couple of formatting options allow you specify the list. The argument has the following syntax:

- listDomains <containerPath> includeDescendentContainers

This invocation of FvLoader prints a list of domain paths to the console. The output may be redirected to a file using piping. The two arguments, explained below, are optional. If none are specified, the command prints all domains in the system.

- containerPath -- If specified, only the domains which reside in a specified container print to the console. If the argument contains spaces, place double quotation marks around it. You type the path to complete the specification.

- includeDescendentContainers -- If the argument is specified, the domains residing in the descendent containers of the containerPath are printed as well.

For example, the command would take the following form if you want to include domains residing in the descendent containers and use the container path /wt.inf.container.OrgContainer=PTC:

```
java wt.fv.FvLoader - listDomains /wt.inf.container.OrgContainer=PTC
includeDescendentContainers
```

Examples with output:

```
C:\> java wt.fv.FvLoader - listDomains /wt.inf.container.OrgContainer=PTC
[/wt.inf.container.OrgContainer=PTC]/PTC
[/wt.inf.container.OrgContainer=PTC]/Default/Project
[/wt.inf.container.OrgContainer=PTC]/Default/Project/Administration
C:\>
C:\> java wt.fv.FvLoader - listDomains "/wt.inf.container.OrgContainer=PTC/
wt.inf.library.WTLibrary=Windchill PDM"
[/wt.inf.container.OrgContainer=PTC/wt.inf.library.WTLibrary=Windchill PDM]/
ChangeItems
C:\>
C:\> java wt.fv.FvLoader - listDomains /wt.inf.container.OrgContainer=PTC
includeDescendentContainers
[/wt.inf.container.OrgContainer=PTC]/PTC
[/wt.inf.container.OrgContainer=PTC]/Default/Project
[/wt.inf.container.OrgContainer=PTC]/Default/Project/Administration
```
Listing Vaulting Rules

Use the arguments -listFvPolicyRules <site name> to print a list of existing vaulting policy rules for given site to the console. This output can be redirected to a file using piping. You can use the output for batch deletion or recreation of policy rules through FvLoader. See the prefixes R, RR, RemoveLocalR, and RemoveReplicaR in the preceding discussions. Note that output of -listFvPolicyRules and the input for these prefixes is almost identical.

The specification of site name is required. If the name of an existing site is specified, only rules related to the file vaults on that site are printed to the console. To print the rules for all sites, specify the following constant argument:

ALL_SITES

Example:

Imagine that we have three sites in the system. There is a master site with name master, a replica site with name replica_11, and a replica site with name replica_99. File vaults on the sites master and replica_11 have rules associated with them. File vaults on the site replica99 do not have rules associated with them.

C:\> java wt.fv.FvLoader -listFvPolicyRules master
###Current Policy rules for site [master]
LocalPolicyRule,v1,wt.doc.WTDocument,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
LocalPolicyRule,v1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL

C:\> java wt.fv.FvLoader -listFvPolicyRules replica_11
###Current Policy rules for site [replica_11]
ReplicaPolicyRule,replica_vault_1,wt.doc.WTDocument,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
ReplicaPolicyRule,replica_vault_1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL

C:\> java wt.fv.FvLoader -listFvPolicyRules replica99

C:\> java wt.fv.FvLoader -listFvPolicyRules ALL_SITES
###Current Policy rules for site [master]
LocalPolicyRule,v1,wt.doc.WTDocument,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL

Configuring External File Vaulting or Replication With FvLoader 5-7
In the output, each line has one of the following prefixes which specifies the type of the rule:

- **LocalPolicyRule** -- Rule is used for external vaulting.
- **ReplicaPolicyRule** -- Rule is used for Content Replication.

If you take any line of output and change prefix to the appropriate prefix for rule creation or deletion, you get a command, which is ready to be used in the FvLoader batch execution. Be careful not to mix the prefixes for rules used in external vaulting and Content Replication.

Original output:

```
LocalPolicyRule,v1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
```

Example: delete command for the same rule:

```
R,v1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
```

Example: create command for the same rule:

```
RemoveLocalR,v1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
```

**Supporting Local Replication**

You can accelerate Content Replication with a series of steps that relies on FvLoader. The steps replicate content to the local site, move content to the destination replica site manually, and update the database to reflect the move. The process can be regarded as having four phases, the first two of which rely on FvLoader:

1. **External Vaulting or Replication Preparation Phase**
2. **Replication Rule Creation Phase**
3. **Replication Phase**
4. **Finalization**

The steps are united in the example at the end of this chapter, [Full Example of Local Replication](#).
External Vaulting or Replication Preparation Phase

1. Create a replica site through the Site Administration graphical user interface. Make sure that master-replica communication is working.

2. Create a file vault host, FvHost, on the replica site.

3. Create a replica vault, rv, on the master site.

4. Create replica folders, ReplicaFolders, for the replica vault, rv.

5. Mount the replica folders, ReplicaFolders, to the file vault, FvHost, on the master site. For this example, the mounts you make are called FvMounts. Create the mounts with attention to the following:

   In order for the local replication to work properly, paths of each of the FvMounts of the FvHosts on the Master site must exactly match the path of the FvMounts of the FvHost on the Replica site for all the folders.

   For example: If the mount path for one of the folders on the host at a local site is `C:\tmp\1`, after completing the fourth phase of local replication, Finalization, the content must be found on the replica site at path `C:\tmp\1`.

6. Enable the replica folders, ReplicaFolders.

7. Enable the replica vault, rv.

**Note:** At this time, all of the objects you have created (vaults, folders, and mounts) are not visible through External Vaulting Configuration dialog.

Replication Rule Creation Phase

Create file vault policy rules, FvPolicyRules, for the replica vault, rv, either through FvLoader or the rule creation graphical user interface.

Replication Phase

1. In wt.properties set wt.fv.localReplica=true. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

2. Stop and restart the system.

3. Launch Replication on the replica vault, rv, through the Content Replication Scheduler dialog.

4. Wait for the completion of the process.

5. Physically move the content of the folders of the replica vault, rv, to the replica site, into the predefined place.
Finalization

1. Launch FvLoader to move the replica vault, rv, with its underlying structure, to the destination replica site. You can perform local replication by supplying the following arguments in the fvloader.csv file or the .csv file that you write:
   - LRV,vaultName -- Creates a remote vault for the local site with the name vaultName.
   - LRMV,vaultName,destinationSiteName,masterHost, replicaHost -- Moves the replica vault from the master site to the replica site. masterHost has the same mount paths as the replicaHost on Replica site

2. Set wt.fv.localReplica=false, or comment or delete the entry.

3. Stop and restart the system.

Full Example of Local Replication

1. Create replica site object through the Site Administration graphical user interface.

2. Set up the replica.

3. Exchange keys

4. Create a host for the replica site.

5. Execute FvLoader with the following arguments in the file that specifies its action:
   LRV,replicaVaultOnMaster
   RF,rrr_folder1,replicaVaultOnMaster
   RF,rrr_folder2,replicaVaultOnMaster
   RF,rrr_folder3,replicaVaultOnMaster
   RM,rrr_folder1,eshilmayster03d.ptc.com,C:\tmp\000\1
   RM,rrr_folder2,eshilmayster03d.ptc.com,C:\tmp\000\2
   RM,rrr_folder3,eshilmayster03d.ptc.com,C:\tmp\000\3
   RFE,rrr_folder1
   RFE,rrr_folder2
   RFE,rrr_folder3
   RVE,replicaVaultOnMaster

   The result is the replica vault, replicaVaultOnMaster, with three folders mounted on the master site.

6. Create rules for the replica vault, replicaVaultOnMaster, through the graphical user interface or by using FvLoader with the RR argument.

8. Copy content from folders to the replica site preserving paths with one host on master.

9. Execute FvLoader with the LRMV argument, specifying replicaVaultOnMaster, destination replica site name, host on master with identical paths to replica, and replica host. The arguments look like this:

```
LRMV,replicaVaultOnMaster,replica,127.0.0.1,replicahost.ptcnet.ptc.com
```

7. The process is now complete. The replica vault, replicaVaultOnMaster, must behave as a regular replica vault on the replica site. All transferred content must be accessible for download. All later scheduled Content Replication on the given vault should proceed as usual.
Windchill Import and Export

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>6-2</td>
</tr>
<tr>
<td>Context Considerations in Import and Export</td>
<td>6-4</td>
</tr>
<tr>
<td>Configuration Specification Settings</td>
<td>6-9</td>
</tr>
<tr>
<td>Import and Export of EPMDocuments</td>
<td>6-9</td>
</tr>
<tr>
<td>Exporting with the Export User Interface</td>
<td>6-11</td>
</tr>
<tr>
<td>Importing with the Import User Interface</td>
<td>6-16</td>
</tr>
<tr>
<td>Additional Export and Import Actions</td>
<td>6-19</td>
</tr>
<tr>
<td>Windchill Properties for Export and Import</td>
<td>6-21</td>
</tr>
<tr>
<td>Windchill Export Properties</td>
<td>6-21</td>
</tr>
<tr>
<td>Windchill Import Properties</td>
<td>6-22</td>
</tr>
<tr>
<td>Access Control for Export and Import</td>
<td>6-22</td>
</tr>
</tbody>
</table>
Overview

Windchill Import and Export can assist you in moving Windchill content and metadata to and from Windchill PDM, Windchill PDMLink, and Windchill ProjectLink sites by placing the data in JAR files.

Windchill Export places in JAR files on your file system all the data held in high-level Windchill objects in the local Windchill database. Windchill Import extracts such JAR files to the local Windchill database.

Windchill Export allows you to compress the data in any of the following Windchill objects into a JAR file:

- **Objects in Cabinets and Folders** -- The content of objects supported by Windchill Export that is located in cabinets and folders is compressed. Folders and cabinets are not supported, but for each supported object, data is included in the JAR file about the cabinets and folders that held it.

- **Product Structure (built with active configuration specification)** -- A WTPart serves as the seed object for a complete product structure, which includes its dependent child WTParts and associated WTDocuments, built with the active configuration specification. The following can serve as seed objects: Subclasses of WTParts, serial numbered parts, instances of soft typed WTParts, and End Items which are also known as WTProducts.

- **CAD Document Structure (built with latest configuration specification)** -- A CAD Document serves as the seed object for a complete CAD structure generated with the latest configuration specification. No WTParts or links between WTParts and CAD Documents are exported.

- **Product Structure with CAD Documents (built with active configuration specification)** -- A product structure of WTParts with CAD Documents combines the two preceding options. It supports exporting the product structure of the WTParts and CAD Structure of the CAD Documents along with the build rules between the two structures. This leads to the export of WTParts, product structure, CAD Documents, CAD structure, and content files.

- **Document** -- Ordinary or Soft Type instances of Documents can be included in the JAR file.

- **Soft Type Definition** -- WTDocuments and WTParts can have soft type definitions. You cannot check out type definitions on export.

The JAR suffix is automatically appended unless you specify another suffix. You can filter objects by their time of last modification to control what is included in a JAR file. Any software that expands ordinary zip files can also expand the JAR files produced by Windchill Export.

Windchill Import allows you to specify the placement in the Windchill database of content that was exported.
Windchill Import will not import an object that already exists in the local Windchill database. Uniqueness is evaluated on the basis of an object identifier. For most business objects such as WTPart, WTDocument and EPMDocument instances, the uniqueness identifier is known as the UFID (unique federated identifier) that is composed of the local ID, the domain, and the site. The UFID is assigned to an object at the time that it is exported. Changing an object's revision, version, or iteration results in a UFID change, but changing the life cycle state does not. Different object types may have different uniqueness identifiers, for example, an instance based attribute (IBA) or soft type definition object can be identified by its name and its path.

An object’s business identity is derived from the value of certain attributes, which are as follows:

- For a WTPart -- Number, Version, Iteration, and View
- For a WTDocument -- Number, Version, and Iteration
- For a CAD document -- CADName, Number, Version, and Iteration

If an object to be imported has the same UFID, but a different business identity than a database object, the import will fail unless the Resolve Overridable Conflicts functionality is selected in the graphical user interface, or a policy or rule file is used to change either the UFID or the business identity of the import object.

Both the export and import processes can refer to mapping rule files that transform or block attribute data on the interface to the JAR file. In addition context mapping files enable the specification of object context during import or export.

The way objects in the database can be created or modified during import and export operations is governed by the use of policy files or selected actions available in the user interface during import or export. If this is not supplied from the user interface, import or export software attempts to find the appropriate actions from server registry files. These policy files or actions are applied after any mapping rule files are applied.

A Preview feature shows the expected results of importing from a specific file. The Preview feature may not report every detail of the results of performing the import operation.

See the appendix, Import and Export Policies, Mapping Rules, and Conflict Messages, for more detailed information about conflicts and policies and mapping rules.
Context Considerations in Import and Export

The software manages objects within logical entities called containers. The container concept is used to separate objects that belong to different working contexts.

This topic discusses the following in its sub-sections:

• Access to objects and the import or export of objects at the appropriate context level
• Controlling context in import operations

Export Container Availability

This topic explains export container availability for Product, Library, Project, Organization, and Site.

Product, Library, or Project Level Container Availability

At the Product/Library/Project level, the export action is available to end users (with read access) and administrators of the context itself, its parent context (the Organization) or a Site Administrator.

The Export action is available on the property (details) page of any of the following types of objects:

• WTPart, WTDocument, Reference Document, and End Item which is also known as WTProduct.

  Any IBA values and IBA definitions must be exported with the object instances. The associated type definitions, either soft types or hard types, must be exported as well.

• All soft type and hard type definitions. In most cases these are instances of modeled subclasses of WTPart, WTDocument, and End Item which is also known as WTProduct.

• Supported objects in a folder.

  The export must include the definition of the modeled subclass as if it were a soft-type.

You can export EPMDocuments through the Import/Export Manager.

Additionally, the Export user interface will be available to administrators of site, organization, product, and library contexts through the Export/Import Manager on the Utility tab. This user interface allows administrators to search within a context for objects to export. The search returns objects created within the context container. To export data from a Project Context, the user should use the export action from the Project Details page.
Organization Level Container Availability

On the Utilities tab, the Export action is available to administrators of the Organization or its parent (Site) context. In this case, the Export only considers type definitions and folder contents that are visible from the given Organization container.

**Note:** Users should exercise care not to import product, library, or project level objects to the Organization or Site level.

Site Level Container Availability

On the Utilities tab, the Export action is available to administrators of the Site container.

Import Container Availability

This section explains import container availability for Product, Library, Project, Organization, and Site.

Product, Library, or Project Level Container Availability

At the product/library/project level, the Import action will be available to the context or parent context (organization or site) administrators. The imported objects will be created in the contexts as specified by the mapping rules, provided that the administrator has write access to the mapped contexts.

The following conditions apply to Import at this level.

- Only business objects may be imported into this level.
- When Type instances are being imported, if the type equivalent is not found in the destination system, the type will be created in the target container's organization container, provided the organization container has a properly configured internet domain, otherwise the type will be created at the site container. The user will be assumed to have the permission to create types at the appropriate container levels, otherwise the import fail. See the following section for **Type Definition Equivalence**.

- Do not use the **Import/Export Manager** to import data into a project. Instead, use the import action available from the Project Details page or the folder import action. There is a slight difference in how these two actions work. The import action from the Project Details page will provide the default behavior for the folder structure of all the objects in the target set. When the import into folder action is performed, all foldered objects are placed into the target folder.

Type Definition Equivalence

For Import purposes, a type definition in an import file is considered equivalent to a local type definition if all of the following criteria are met:
• It has the same name or the name is mapped to a local name, as well as mapped to the same parent type, unless this type is a root-level type. The names of hard types cannot be changed.

• It has the same values for the following attributes: instantiable, userAttributeable, deleted.

• The two types have the same number of attribute.

• The two types have the same set of soft attributes. Two soft attributes are considered the same if they are of the same IBA type have the same value.

• The two types have the same set of constraints on their attributes as well as the same set of constraints on the soft type itself.

If a Type matching the above criteria is found in the system, it must be visible to the context into which the import is being performed.

Organization Level Container Availability

The import action is available to the Organization or Site administrator. Folder contents can be imported into an organization context. Type Definitions can be mapped to locally defined type definitions.

Site Level Container Availability

The import action is available to the Site administrator. Folder contents can be imported into a Site context. Type Definitions can be mapped to locally defined type definitions.

Controlling the Destinations of Imported Objects with Context Mapping Files

The preceding sections discussed working contexts, which correspond to logical entities called containers, within which the software organizes objects. The containers separate objects in different working contexts. The following are examples of containers:

• Each installation of Windchill PDM has a different context from others, so each Windchill Foundation installation has its own ClassicContainer.

• In Windchill ProjectLink, each project has a different context from other projects, so each project is its own Project Context.

• Each organization has a different context from others, so each organization has its own OrgContainer. For Windchill PDMLink the namespace that is used is the Site level, and for Windchill ProjectLink the namespace that is used is the Organization level.

In light of the preceding examples, you can expect objects exported from the context of a system, project, or organization to be present after import in new contexts within the system to which they were imported.
You can import objects to destination contexts that have specified correspondences to the contexts from which the objects were exported. The specifications that control the destination contexts for imported objects are called context mapping files. You may select a context mapping file in the import graphical user interface.

When objects are imported without a context mapping file in control, all the objects are imported into the target context where the import process launched. This context file is ignored during import into a Project because to do so would violate Windchill ProjectLink security guarantees. If you wish to override this behavior, you must use the Export/Manager for Windchill ProjectLink by importing into the Organization container level.

The context mapping file allows the distribution of imports into multiple targets. The context mapping file is intended only for advanced users who cannot find another resolution. The context mapping file hard-wires the container paths in your import set, so this approach requires detailed synchronization between the source and target system which is typically only achievable via pilot to production export scenarios. A better approach is to analyze what your transport needs really are, and then to streamline the creation of appropriate export sets. PTC does not recommend extensive use of the context mapping file functionality at your site.

The context mapping file has the following syntax:
There can be more than one <container> elements in the mapping file, as shown in the following example:

```xml
<container-info>
  <container>
    <target-container>/wt.inf.container.OrgContainer=Windchill_RD/wt.inf.library.WTLibrary=Windchill PDM</target-container>
  </container>
</container-info>
```
Configuration Specification Settings

An export operation refers to a configuration specification to determine the data to include in the JAR file, and the selection of configuration specification is made in the following manner:

If the user performing an export operation for the first time does not select a configuration specification, the current preference for configuration specification determines the objects exported. In this case the current preference for configuration specification is saved as a modifiable default for the future.

If the user performing an export operation for the first or any other time selects a configuration specification, the selection determines the objects exported and is saved as a modifiable default for the future.

Import and Export of EPMDocuments

This section discusses some limitations related to EPMDocuments and the behavior related to exporting and importing EPMDocuments as checked out.

Attribute Limitations

Because the attributes of CAD documents are tightly related to content files, there are limitations on which attribute can change outside the workgroup manager clients. The following import actions are not supported for CAD documents:

- Create a new object with new identities
- Substitute for an existing object
- Ignore object

Mapping rules allow a user to change any attribute specified in an import or export XML file. When working with CAD documents, only the following attributes should be changed by mapping rules:

- name
- number
- CADName
- description
- folderpath
- versionInfo
- lifecycleInfo
- teamIdentity
When importing CAD documents, mapping rules should not be used to change the following attributes:

- ownerApplication
- authoringApplication
- epmDocType
- epmDocSubType
- extentsValid
- contentItem
- iba

Mapping rules should not be used to change any attributes on other EPM link objects, including:

- EPMMemberLink
- EPMReferenceLink
- EPMVariantLink
- EPMContainerLink
- EPMDescribesLink
- EPMBuildLinksRule
- EPMBuildHistory

**Exporting and Importing EPMDocuments as Checked Out**

When you export or import EPMDocuments as checked out, they are located in a workspace whose name is the name of the import jar file with its extension removed. This behavior is different from what occurred in previous releases of the software.

For example, if the jar file abc.jar includes EPMDocuments exported as checked out, they are located in the workspace abc, and an import operation abc.jar with the EPMDocuments as checked out results in their being checked out to the workspace abc. The workspace is automatically generated if it does not already exist.
Exporting with the Export User Interface

If you have read access to an object, are an administrator of the object’s context (for example, product, library, or project), are an administrator of the parent context (the organization), or are a site administrator, you can perform an export. Administrators can access the Export/Import Manager from the Utilities sub-tab of the context. Export from a Project should be done via the export action on the Project Details page, or through the export action on the object/folder action list.

Exporting from the Export Window

To display the Export window, perform either of the following two sets of actions:

- Click System Administration on the Windchill home page to display the System Administration page. Click Import/Export Manager.
- Click Site Map on the Windchill home page. In the System Administration list in the Site Map click Import/Export Manager.

1. Click Export to display the Export window. Perform the following steps in the Export window to export data,
2. You may optionally specify a folder and file name in the local file system for the exported data jar file: Click **Browse** for the **Export File Name** box.

3. You may optionally specify a mapping rule file in the local file system to control the export process: Click **Browse** for the **Export Rule File** box. Specify the folder and file in the dialog box that appears.

4. You may optionally specify export policies by one of the following two methods.

   - Select the Export Policy File radio button and clicking **Browse** for the **Export Policy File** box. Specify the folder and file in the dialog box that appears. Export actions in the file will be combined with ones found in the system’s registry files in
     `<Windchill>/codebase/registry/ixb/export_settings/defaultExportPolicy.xsl`

   - Select the **Export Action** radio button and then select from the export action drop-down list (actions will be applied to all objects in the export
file). The Lock action is not shown as a selection, but it is applicable through an export policy file or the system registry.

- None -- If you are sure no actions such as checkout and lock are needed, this is an appropriate selection.
- System default -- Actions specified in the system registry will be applied.
- Check out -- Upon export, the database object will become checked out by you. The software attempts to check out objects that are necessary to describe an object that you are exporting, such as a document that describes a part that you are exporting. You cannot check out type definitions on export.

5. Set the configuration specification in the Configuration Specification section of the window to specify a configuration specification, baseline, or effectivity for the exported object. Setting the configuration specification is optional.

6. Click Add in the Objects section of the window to select data for export. You can remove objects from the list by selecting them and clicking Delete. You select a type of object and then display a window for selecting the object. The following graphic shows the appearance of the window if you selected a Document as the type of object to add.
7. Click **Add** in the Filters section of the window to specify a time period that defines the objects for export. Adding filters is an optional step. You can filter objects to be exported by their time of last modification in all languages, but Time Range user interface is available for English locale only. For other languages, the only user interface option is "during previous .. days/hours/months". This variation affects the features available in the Filter by Time window.

8. If you need to, click **Preview** to understand what will be exported. With **Detailed Log** not checked, you can see how many objects will be exported and how many XML files will be processed. With **Detailed Log** checked, you can see what files will be exported.

9. Click **Submit**.

Messages in the **Export Status Log** section of the **Export** window show progress or problems that you can resolve. See the appendix that explains mapping rules, policy files, and conflict messages in the Windchill System Administrator's Guide for information that can help in resolving conflicts.

### Exporting an Object from its Properties Page

A part, document, or product, or a folder's contents, can be exported from its properties page, by performing the following steps. When exporting from the properties page, you cannot add additional objects for export.

1. Navigate to the properties page of the object you want to export.

2. In the Pick an Action drop-down list, select **Export** and click **Go**. The **Export Manager** window appears, followed by the **Export** page with the object already selected in the **Objects for Export** field.
3. Specify a folder and file name in the local file system for the exported data jar file: Click Browse for the Export File Name box. Specify the folder and file in the dialog box that appears.

4. You may optionally specify a mapping rule file in the local file system to control the export process: Click Browse for the Export Rule File box. Specify the folder and file in the dialog box that appears.

5. You may optionally specify export policies by one of the following two methods.
   - Select the Export Policy File radio button and clicking Browse for the Export Policy File box. Specify the folder and file in the dialog box that appears. Export actions in the file will be combined with ones found in the system’s registry files in <Windchill>/codebase/registry/ixb/export_settings/defaultExportPolicy.xsl
   - Select the Export Action radio button and then select from the export action drop-down list (actions will be applied to all objects in the export file). The Lock action is not shown as a selection, but it is applicable through an export policy file or the system registry.
     - None -- If you are sure no actions such as checkout and lock are needed, this is an appropriate selection.
     - System default -- Actions specified in the system registry will be applied.
     - Check out -- Upon export, the database object will become checked out by you. The software attempts to check out objects that are necessary to describe an object that you are exporting, such as a document that describes a part that you are exporting. You cannot check out type definitions on export.

6. Click Set Config Spec in the Configuration Specification section of the window to specify a configuration specification, baseline, or effectivity for the exported object.

7. In the Navigator Ids section of the window, select either Product Structure (built with active configuration specification) or Product Structure with CAD Document (built with active configuration specification).

8. Click Add in the Filters section of the window to specify a time period that defines the objects for export. Adding filters is an optional step. You can filter objects to be exported by their time of last modification in all languages, but Time Range user interface is available for English locale only. For other languages, the only user interface option is "during previous .. days/hours/months". This variation affects the features available in the Filter by Time window.
9. If you need to, click Preview to understand what will be exported. With Detailed Log not checked, you can see how many objects will be exported and how many XML files will be processed. With Detailed Log checked, you can see what files will be exported.

10. Click Submit.

Messages in the Export Status Log section of the Export window show progress or problems that you can resolve. See the appendix that explains mapping rules, policy files, and conflict messages in this document for information that can help in resolving conflicts.

**Importing with the Import User Interface**

The Import window allows you import data. During an import, mapping rules are applied first to modify the content of the import source XML file. You can specify with context mapping rules the context into which you want to import objects from various source contexts. Then, if an import object exists in the target database, import policies or import actions selected from the Import window are applied to determine how that object is modified.

**Note:** The Access Control List (ACL) applies to import operations. For example, a user without Revise privileges for a particular object type cannot defeat ACL control by using the import action Import as a New version. If an import attempt specifies an object action for which you do not have privileges, the entire transaction will fail.

The Import action is available to administrators of a context or its parent context (for example, organization or site). Imported objects are created in containers as specified by context mapping rules, provided that the administrator has write access to the mapped context. Type definitions should only be imported at the site or organization levels. Product or library objects should not be imported into the site or organization levels.

Do not use the Import/Export Manager to import data into a project. Instead, use the import action available from the Project Details page or the folder import action. There is a slight difference in how these two actions work. The import action from the Project Details page will provide the default behavior for the folder structure of all the objects in the target set. When the import into folder action is performed, all foldered objects are placed into the target folder.

To display the Import window, perform either of the following two sets of actions:

- Click System Administration on the Windchill home page to display the System Administration page. Click Import/Export Manager.

- Click Site Map on the Windchill home page. In the System Administrator list on the Site Map click Import/Export Manager.
1. Click **Import**. The **Import** window opens, displaying your current context at the top of the window. Perform the following steps in the **Import** window to import data.

2. Specify the exported data JAR file in the local file system to import to the local Windchill database. Click **Browse** for the **Import File Name** box. Specify the folder and file in the dialog box that appears.

3. Specify a mapping rule file in the local file system to control the import process. Click **Browse** for the **Import Rule File** box. Specify the folder and file in the dialog box that appears. Specify a mapping rule file is optional.

4. You may specify a context mapping file in the local file system to identify into which target context the import file objects are placed. If you do not specify a context mapping file, objects will be imported into the context from which the import action was launched, the **Default Target Context** listed at the top of the **Import** window. Click **Browse** for the **Context-mapping File** box. Specify the folder and file in the dialog box that appears. For a more complete explanation see a later section in this document, **Controlling the Destinations of Imported Objects with Context Mapping Files**.
5. You may optionally specify import policies by one of the two following methods.

- Select the **Import policy File** radio button and click **Browse** for the policy file box. Specify the folder and file in the dialog box that appears. Import actions in the file will be combined with ones found in the system’s registry files in `<Windchill>/codebase/registry/ixb/import_settings/defaultImportPolicy.xsl`.

- Select the **Import Action** radio button and then select from the import action drop-down list (actions will be applied to all objects in the import file):
  
  - **Default** -- If the import object matches the full object identity of an existing database object, the existing object is picked and no import takes place. If the import object is new, it will be created with a version.iteration matching the version.iteration in the import XML file.
  
  - **Import as latest iteration** -- If the import object is newer than the existing db object, the import process will create an object with the next available iteration on the latest iteration. If the object does not exist in the target database, applying this action will create a new object. If you are sure that a new iteration should be created for all objects that can be iterated, this option is appropriate.
  
  - **Import as new version** -- If there is a version of the import object in your site's database, the import process will create an object with the next available version and the first iteration. Otherwise, a new object will be created. If you are sure that a new version should be created for all objects that can be versioned, this option is appropriate.
  
  - **Import as checked out** -- If there is a version of the import object in your site's database, the import process will create a checked out (working) copy of the existing object with the same version as in the import XML file. The newly imported object will be kept checked out until you check it in. If you are sure that all existing objects should be checked out for all objects that can be checked out, this is an appropriate option.
  
  - **Modify non-versioned attributes** -- This option will update certain non-versioned attributes (for example, life cycle) of an existing database object without iterating the object.
  
  - **Update checked out object in place** -- This option will replace the content, attributes, and links of the checked out object.

6. Select or clear the "Resolve Overridable Conflicts" checkbox. This checkbox controls the value of the property `wt.ixb.import.overrideConflicts`. Most import operations fail if this checkbox is not selected. There are two types of conflicts in Windchill: overridable and non-overridable conflicts. Whether a
conflict is overridable or not is dependent on the target database, the jar file (containing metadata XML files and content files) to be imported, as well as the import actions. Selecting the checkbox Resolve Overridable Conflicts will only resolve the overridable conflicts and can not resolve the non-overridable conflicts. If there are one or more non-overridable conflicts, the import operation fails. If failure occurs, in order to have a successful import operation, something must be done prior to the next attempt to do the same import operation. For example, apply a mapping rule file to the XML files to ensure no non-overridable conflicts will happen against the target database.

Note: A particular change for the 7.0 release that can produce conflicts involves the RatioDefinition and RatioValue. These types of data, if included in an export from 6.2.6 or earlier, result in an overridable import conflict in the 7.0 release. If you override the conflict, the data is imported as FloatDefinition and FloatValue.

7. You may click Preview to understand what will be imported. With Detailed Log cleared, you can see how many objects will be imported and how many XML files will be processed. With Detailed Log checked, you can see what files will be imported, what conflicts may arise during import, and whether the import process will be completed or will fail. PTC recommends using Preview, especially for checking the effect of policy files, which have the potential of creating significant changes to the database. The Preview operation does not perform actual import, nor does it report all conflicts, especially those from runtime.

8. Click Submit.

Messages in the Import Status Log section of the Import window show progress or problems that you can resolve. See the appendix that explains policy files, mapping rules, and conflict messages in the Windchill System Administrator's Guide for information that can help in resolving conflicts.

Additional Export and Import Actions

The following export and import actions are available, but are not presented as options in the graphical user interface.

Additional Lock Export Action

The Lock action can be imposed through an import policy file or the system registry.

Additional Import Actions

These actions are imposed through an import policy file or the system registry:

- UnlockAndIterate -- This action finds an object in the database with the same UFID or the same name, number, version, and iteration as the object in the XML file. If such an object exists and it is locked, this action unlocks and
iterates it, and then updates it with information from the XML file. Otherwise, the action generates an error.

- **CreateNewObject** -- This action creates a brand new object with new name, new number, new version, and new iteration provided in the import policy file. Other information is extracted from the XML file. This functionality must be used with a policy file that provides new identities for the object.

The format of new information that must be provided in Import Policy file is the following:

```xml
<actionInfo>
  <xsl:choose>
    <xsl:when test="criteria='value'">
      <action>CreateNewObject</action>
      <actionParams>
        <newName>New Name</newName>
        <newNumber>New Number</newNumber>
        <newVersion>New Version</newVersion>
        <newIteration>New Iteration</newIteration>
      </actionParams>
    </xsl:when>
    <xsl:otherwise>
      <action>Some other action</action>
    </xsl:otherwise>
  </xsl:choose>
</actionInfo>
```

Please note that:

- `<actionInfo>` must always exist.
- Criteria can be any valid attribute of the object in XML file.
- Between `<xsl:choose>`, there can be many `<xsl:when test ....>` with different criteria and different action names.
- Only `CreateNewObject` and `SubstituteObject` can have action parameters, and there are only four action parameters: `<newName>`, `<newNumber>`, `<newVersion>`, and `<newIteration>`, and all of them must be provided.

- **SubstituteObject** -- This action substitutes the object in the XML file for an object in the database that has the name, number, version, and iteration provided in the Import Policy file. If such an object doesn't exist, it generates
an exception. The format of tag and parameters for this case is exactly the same with CreateNewObject, but the <action> is SubstituteObject.

- Ignore -- This action does not import the object in the XML file. This action doesn't require any actor.

Windchill Properties for Export and Import

There are some properties to control Windchill export or import operations. These properties can appear in the wt.properties file or in mapping files. If you are not setting properties through a graphical user interface or in mapping files, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

- wt.ixb.logLevel (name in wt.properties file or in mapping files) or logLevel (name in mapping files) -- This property specifies the log level for both export and import operations. The default value is 0, which means least information will be written into the log files, for example MethodServer.log. If this value is set to 4 or larger, it will be in the debug mode.

- wt.ixb.warningAsError (name in wt.properties file or in mapping files) or warningAsError (name in mapping files) -- The default value is false. It specifies whether a warning from either export or import should be treated as an error or not.

Windchill Export Properties

There are some properties to control Windchill export operations. These properties can appear in the wt.properties file or in mapping files. If you are not setting properties through a graphical user interface or in mapping files, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

- wt.ixb.export.objectSetPageSize (name in wt.properties file or in mapping files) or export.objectSetPageSize (name in mapping files) -- This property specifies the page size for export. It deals with an out of memory problem when the number of objects to be exported is very large. If out of memory is still experienced, try to decrease the page size. The default value is 1000.

- wt.ixb.export.validateOnExport (name in wt.properties file or in mapping files) or export.validateOnExport (name in mapping files) -- The default value is false. Usually for performance considerations this property is set to false in wt.properties, and it is recommended to set the value to true from the client, if necessary. When this property is set to true, it will ensures that the export operations generate valid XML files, for virtually any XML parser, when the XML files contain "strange" characters. If this property is set to false, some values in the XML files (mostly for the attributes which are user editable, such as description of WTDocument objects) may not be acceptable by the XML parser even if they are manually wrapped by Cdata Sections.
Windchill Import Properties

Two properties control Windchill Import operations. These properties can appear in the wt.properties file or in mapping files. If you are not setting properties through a graphical user interface or in mapping files, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

- wt.ixb.import.parser.validate (name in wt.properties file) or import.parser.validate (name in mapping files) -- This property specifies whether the imported document is validated by the XML parser. Its possible values are true and false. Its default value is false.
- wt.ixb.import.overrideConflicts (name in wt.properties file) or import.overrideConflicts (name in mapping files) -- This property allows the overriding of overridable folder and other conflicts during importing. Its possible values are true and false. Its default value is false. This property controls import operations for Windchill Import.
- import.default.lifecycleInfo.lifecycleTemplateName -- When the life cycle template name is missing from the xml file, the default name that is the value of this property will be assigned to the object.
- import.default.lifecycleInfo.lifecycleState -- When the life cycle state is missing from the xml file, the default state that is the value of this property will be assigned to the object.
- wt.iba.definition.hierachicaldefinition.enabled -- In Release 7.0 it is suggested that you do not create hierarchical IBA definitions unless this property has the value true in the wt.properties file. Setting the preceding property’s value true allows the import of hierarchical IBA definitions. By default in Release 7.0, the default value of the property is false, and that value allows the creation of hierarchical IBA definitions. A false value for the property prevents the import of hierarchical IBA definitions, except when you use a properly written mapping file, called a mapping file. A mapping file maps hierarchical IBA definitions to non-hierarchical IBAs.

Access Control for Export and Import

The Access Control List (ACL) applies to both Export and Import operations. If an export or import attempt specifies an object action for which you do not have privileges, the entire transaction will fail.

For example, a user without Revise privileges for a particular object type cannot defeat ACL control by using the import action “Import as a New version.” Therefore the access control rule for importing new versions of WTParts includes Revise permission.

As another example, the access control rules for importing IBAs include their specific type to allow non-administrator users to import them: FloatDefinition, BooleanDefinition, IntegerDefinition, RatioDefinition, StringDefinition, and URLDefinition.
You must log in as administrator and set the following access control rules for Windchill export and import operations for non-administrative users.

The rules in the following tables are examples that may meet your needs for Windchill PDM or Windchill ProjectLink. They do not attempt to represent the minimum permissions required for a non-administrator to perform the indicated actions.

**Export Access Control Rules**

You create the WTParts, WTDocuments, EPMDocuments, and Folders as an administrator, and you export them as a non-administrator.

**Export Rule for All Objects**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>Cabinet</td>
<td>All</td>
<td>non-administrator user</td>
<td>Full Control (All)</td>
</tr>
</tbody>
</table>

**Export Rules for WTParts**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Site</td>
<td>View</td>
<td>All</td>
<td>non-administrator user</td>
<td>Full Control (All)</td>
</tr>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTPart</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read</td>
</tr>
</tbody>
</table>

**Export Rule for WTParts With Policy File for Lock and Checked Out**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTPart</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify</td>
</tr>
</tbody>
</table>
### Export Rule for WTDocuments

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read</td>
</tr>
</tbody>
</table>

### Export Rule for WTDocuments With Policy File for Lock and Checked Out

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify</td>
</tr>
</tbody>
</table>

### Export Rule for EPMDocuments

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>EPMDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read</td>
</tr>
</tbody>
</table>

### Export Rule for EPMDocuments With Policy File for Lock and Checked Out

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>EPMDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify</td>
</tr>
</tbody>
</table>
Export Rule for Nested Folders

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>SubFolder</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read</td>
</tr>
</tbody>
</table>

Import Access Control Rules

Import Rule for All Objects

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>Cabinet</td>
<td>All</td>
<td>non-administrator user</td>
<td>Full Control (All)</td>
</tr>
</tbody>
</table>

Import Rules for WTParts

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Site</td>
<td>View</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Creat e</td>
</tr>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTPart</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Creat e</td>
</tr>
</tbody>
</table>

Import Rule for New Versions of WTParts

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTPart</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Cre at/Revise</td>
</tr>
</tbody>
</table>
### Import Rule for WTDocuments

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Create</td>
</tr>
</tbody>
</table>

### Import Rule for New Versions of WTDocuments

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>WTDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Create/Revise</td>
</tr>
</tbody>
</table>

### Import Rule for EPMDocument

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>EPMDocument</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Create</td>
</tr>
</tbody>
</table>

### Import Rule for Nested Folders

<table>
<thead>
<tr>
<th>Domain</th>
<th>Context</th>
<th>Type</th>
<th>State</th>
<th>Principal</th>
<th>Grant Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Windchill PDM</td>
<td>SubFolder</td>
<td>All</td>
<td>non-administrator user</td>
<td>Read/Modify/Create</td>
</tr>
</tbody>
</table>
7

Administering Content Holders and Content Objects

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of Content Holders</td>
<td>7-2</td>
</tr>
<tr>
<td>Content Handling Configuration</td>
<td>7-3</td>
</tr>
<tr>
<td>Adding and Updating Data Formats</td>
<td>7-4</td>
</tr>
</tbody>
</table>
Overview of Content Holders

A number of Windchill information objects, including all document types and change objects (change requests, change orders, and change activities), are modeled as content holders. A content holder is an object to which files and URLs can be attached. For example, when you create a Windchill document object and save it to the Windchill database, files and URLs can be added to the object. The files and URLs are then uploaded to the database or to an external file vault. For information about file vaults, see Administering External File Vaults.

Content can be uploaded to and downloaded from content holder objects in the following ways:

- Through HTML forms/hyperlinks
- Through Java applets making RMI calls
- Through a SOAP implementation such as Windchill Desktop Integration 2.0.

The content attached to a Windchill object can later be viewed, downloaded, removed, or replaced with new or updated content, subject to user permissions and the status of the Windchill object.

All content holder objects can have unlimited content attachments, but only FormatContentHolder objects can have a primary content attachment in addition to their unlimited secondary content attachments.

Content can be replicated to increase the productivity of Windchill users. For information about content replication, see Administering Content Replication.

This chapter describes the following:

- Content Handling Configuration
- Adding and Updating Data Formats (which define MIME types for downloading content objects)
## Content Handling Configuration

The following Windchill properties can be used to configure the content handling capabilities of Windchill. The Property column shows the default setting for each property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.clients.debug=false</td>
<td>When set to true, debug information is printed to the Java Console from Windchill applets. This property is not specifically a content property, but it can be useful when troubleshooting upload or download problems where applets are involved. The value should be changed to true only when there is a specific need to generate applet debug output for troubleshooting purposes.</td>
</tr>
<tr>
<td>wt.content.DEBUG=false</td>
<td>When set to true, enters debug information in the method server log. The value should be changed to true only when there is a specific need to generate content-handling debug output for troubleshooting purposes.</td>
</tr>
<tr>
<td>wt.content.httpClass=wt.content.ContentHttp</td>
<td>Identifies the class that processes HTTP requests for upload and download operations. Currently, this value is not configurable and should not be changed.</td>
</tr>
<tr>
<td>wt.content.temp=$(wt.temp)</td>
<td>Identifies a temporary directory to which files will be written upon upload for intermediate processing. This property is not currently used.</td>
</tr>
<tr>
<td>wt.content.uploadImpl=rmi</td>
<td>Identifies the communication protocol used when uploading content from a Windchill applet. Currently, this value is not configurable and should not be changed.</td>
</tr>
<tr>
<td>wt.content.validEmptyFile=false</td>
<td>Identifies whether a 0-byte file is considered valid for Windchill content (true) or invalid (false). Typically, a 0-byte file is the result of some sort of failure in saving or transferring a file. Therefore, the default value is false. This value should be set to true only if Windchill needs to store files from some other application or process that actively utilizes 0-byte placeholder files.</td>
</tr>
</tbody>
</table>
These properties reside in the `wt.properties` file. Use the `xconfmanager` utility to display existing values or set values for these properties. For details on using the `xconfmanager`, see Using the `xconfmanager` Utility.

See Windchill Configuration Properties for descriptions of all available properties.

### Adding and Updating Data Formats

When content files are added to a content holder object, the format of the file (based on the file name extension) is set automatically upon upload. The available formats are stored as DataFormat objects in the system. In some cases, you may need to augment or change the existing data formats to accommodate additional MIME types associated with your enterprise data.

A data format:

- Sets the MIME type when a file is downloaded.
- Supplies the icon that will represent the object in browser displays.
- Informs the user of the file format.

You can use a command-line utility, provided by Windchill, to maintain data format objects.

### Adding Data Formats

Use the following command to add a data format to the system:

```
java wt.content.DataFormatUtil -add
```

You are then prompted to provide values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format name (required)</td>
<td>Displays this name (for example, Microsoft Word) in the client to identify the file format. The value specified must be unique.</td>
</tr>
</tbody>
</table>
**Note:** After a data format has been added to the system, it cannot be removed.

### Updating Data Formats

Enter the following command to update an existing data format:

```java
java wt.content.DataFormatUtil -update
```

The tool prompts you for the format name of an existing data format object. Once you have identified the data format, you can update its attributes.

**Note:** If you change the MIME type of a data format object, you must stop and restart the method server to implement the change.

### Listing Available Data Formats

Enter the following command to display a list of existing data format objects and their attributes:

```java
java wt.content.DataFormatUtil -list
```
# Configuring and Administering Background Queues

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>8-2</td>
</tr>
<tr>
<td>Queue Entry States</td>
<td>8-3</td>
</tr>
<tr>
<td>Configuring Background Queues and Related Properties</td>
<td>8-4</td>
</tr>
<tr>
<td>Regular Queue Maintenance</td>
<td>8-8</td>
</tr>
</tbody>
</table>
Overview

During day-to-day operation of Windchill, certain system tasks must be completed immediately, while others can wait until a more convenient time. For example, updating RetrievalWare indexes based on events included in the indexing policy must be completed, but you may decide to defer that processing, consigning those tasks to a queue where they can be run at specified intervals. For example, usually many Windchill tasks, including updates to RetrievalWare, e-mail notifications, and many life cycle tasks, can be moved to an ordered background queue rather than being executed immediately.

To keep your system running efficiently, perform regular queue maintenance. For more information, see Regular Queue Maintenance, at the end of this chapter.

You can configure background queues with Windchill property values defined in the wt.properties file. The Queue Manager utility provides you with capabilities for creating and managing background queues. This utility can be accessed from the System Configurator. For information about opening the System Configurator, see Using the System Configurator.

Queues can be distributed among background method servers by using queue grouping. Establish queue grouping by completing two major tasks:

- Assign queues to groups through the Queue Manager utility. The group names can consist of alphanumeric characters. One or more queues can be assigned to the same group.

- Assign groups to background method servers by setting the wt.queue.queueGroup property in each server to one or more groups.

When you assign groups to background method servers, the queues that have not been assigned to any group are automatically assigned to the default queue group and run on the background method server that has the default group assigned. Unless the property wt.queue.queueGroup is set on a running method server to a given group, the queues that form the group are not be executed.

**Note:** Do not set the same group to run on more than one method server. Also, the wt.queue.executeQueues property overrides the wt.queue.queueGroup property, and when wt.queue.executeQueues is set to false, the given method server does not run any queues in spite of setting the queue group. Also, assigning a queue to a group that has not been assigned to a background method server causes the queue execution to halt.

For use with Windchill clusters, Windchill allows you to set the wt.queue.queueGroup property in a wt.properties file to the keyword localhost. Setting wt.queue.queueGroup=localhost establishes the queue group name for the method server as the local host name (in all lowercase characters) of the system where the method server is running. Using this setting in multiple wt.properties files, where each method server is running on a different local host establishes the queue group names as the local host names. For example, assume you have three hosts named appsrvr1, appsrvr2, and appsrvr3. Then setting
wt.queue.queueGroup=localhost in each wt.properties file sets up three queue groups named appsvr1, appsvr2, and appsvr3.

Also, you can set the default queue group to map to another established group by setting the wt.queue.queueGroup.default property. For example, setting the following properties in the wt.properties file on the method server that resides on the appsvr2 host sets the queue group appsvr2 and maps the default group of queues to the group named appsvr1. Additional queues are mapped to appsvr2 and appsvr3:

```
wt.manager.monitor.start.BackgroundMethodServer=1
wt.queue.queueGroup.default=appsvr1
wt.queue.queueGroup=localhost
wt.queue.xxx=appsvr2
wt.queue.yyy=appsvr2
wt.queue.zzz=appsvr3
```

You can change the group to which a queue is assigned by changing queue properties through the Queue Manager utility.

**Queue Entry States**

Each entry in a background queue corresponds to a processing task. An entry can be in one of the following states:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready</td>
<td>Corresponds to the initial state of the task. Only entries in the <strong>Ready</strong> state are selected for execution, based on the order in which they were inserted into the queue (first in, first out).</td>
</tr>
<tr>
<td>Suspended</td>
<td>Indicates that an entry is to remain in the queue, but is not eligible for execution until you change its state to <strong>Ready</strong>. Changing a state can be accomplished from the <strong>Queue Manager</strong> tab in the System Configurator.</td>
</tr>
<tr>
<td>Executing</td>
<td>Indicates that an entry is selected for execution. From the <strong>Executing</strong> state, the entry goes to either the Completed or the Failed state.</td>
</tr>
<tr>
<td>Completed</td>
<td>Indicates that the task was successfully executed. You can purge queues of <strong>Completed</strong> entries so that the size of a queue does not exceed the storage capacity of the database.</td>
</tr>
<tr>
<td>Failed</td>
<td>Indicates that an error occurred during execution. You can purge queues of <strong>Failed</strong> entries so that the size of a queue does not exceed the storage capacity of the database.</td>
</tr>
</tbody>
</table>
Configuring Background Queues and Related Properties

This section describes the properties for background queues, background queue logs, and other related parameters. All the properties in the tables that follow are defined in the wt.properties file.

Background Queue Properties

Use the properties described in the following table to configure the background queues:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wt.queue.&lt;queueName&gt;</code></td>
<td>Assigns the queue named <code>&lt;queueName&gt;</code> to a queue group. This property is set through the Queue Manager utility when it assigns queues to groups. For additional information, see Overview.</td>
</tr>
<tr>
<td><code>wt.queue.defaultInterval</code></td>
<td>Sets the number of seconds in the initial polling interval. A background queue processes all entries in the Ready state and then enters a waiting state, called the polling interval. The queue begins processing again when the polling interval has elapsed. Default is 60.</td>
</tr>
<tr>
<td><code>wt.queue.execEntriesCount</code></td>
<td>Sets the number of entries queried from the queue to be executed. Default is 6.</td>
</tr>
<tr>
<td><code>wt.queue.executeQueues</code></td>
<td>Establishes whether a method server is used to execute background queues. Set this property to false when you do not want a method server to execute any background queues. Setting this property to false overrides any wt.queue.queueGroup property that is set. Default is false.</td>
</tr>
</tbody>
</table>
### Background Queue Log Properties

Use the properties described in the following table to configure the background queue logs. Edit these properties to create a log of queuing events, select the queue log file, and determine whether or not queue entries append to or overwrite the existing log file. Currently, most queue service logging is directed to the method server or background method server so you can view queue events in context of other activities.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wt.queue.max.processQueues</code></td>
<td>Sets the maximum number of process queues that the queue service creates before throwing an exception. Default is 12.</td>
</tr>
<tr>
<td><code>wt.queue.max.scheduleQueues</code></td>
<td>Sets the maximum number of schedule queues that the queue service creates before throwing an exception. Default is 12.</td>
</tr>
<tr>
<td><code>wt.queue.queueGroup</code></td>
<td>Assigns queue groups to a method server. To specify multiple groups, separate the group names using either a comma or a space. For additional information, see <a href="#">Overview</a>.</td>
</tr>
<tr>
<td><code>wt.queue.queueGroup.default</code></td>
<td>Maps the default queue group to an established group. The default queue group consists of all queues that have not been explicitly assigned to a queue group through the Queue Manager utility. For additional information, see <a href="#">Overview</a>.</td>
</tr>
<tr>
<td><code>wt.queue.queueMonitor.sleep</code></td>
<td>Sets the default number of seconds that the queue monitor sleeps before rechecking the integrity of the queues. (The queue monitor also wakes up when certain events occur.) Default is 120000.</td>
</tr>
<tr>
<td><code>wt.queue.removeCompleted</code></td>
<td>Specifies whether successfully completed entries are removed from the Windchill database. If they are not removed, they can overflow the database storage capacity. Default is true (to remove).</td>
</tr>
<tr>
<td><code>wt.queue.&quot;queueName&quot;.removeFailedEntries</code></td>
<td>Determines whether failed entries are automatically removed. Default is false (to not remove).</td>
</tr>
<tr>
<td><code>wt.queue.&quot;queueName&quot;.exceptionRetries</code></td>
<td>Determines whether failed execution entries are retried. These retries occur back to back, with no wait time. Default is 0.</td>
</tr>
</tbody>
</table>
The properties in the table below provide information related to queue architecture and schedule queue type:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.queue.log.enabled</td>
<td>Determines whether queue events are logged. This is a global property, so the log created contains information about all existing queues.</td>
</tr>
<tr>
<td></td>
<td>Default is false (to not log queue events).</td>
</tr>
<tr>
<td>wt.queue.log.file</td>
<td>Sets the name of the queue log file.</td>
</tr>
<tr>
<td></td>
<td>Default is Queue.log in the current log directory. The default log directory, specified in the Windchill wt.properties file, is $(wt.logs.dir)\Queue.log, where wt.logs.dir is set to $(wt.home)\logs.</td>
</tr>
<tr>
<td>wt.queue.log.append</td>
<td>Determines whether queue log entries append to or overwrite the existing log file.</td>
</tr>
<tr>
<td></td>
<td>Default is true.</td>
</tr>
<tr>
<td>wt.queue.pollingQueueThread .verbose</td>
<td>Provides debug information specific to the actual polling threads.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.processingQueue .verbose</td>
<td>Provides general processing queue information that can be used to debug problems.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.processingQueue .execEntries.verbose</td>
<td>Provides debug information related to the execution of a set of process queue entries.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.queueWatcher .verbose</td>
<td>Provides information related to the control of a specific queue. Each queue has an associated queue watcher.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.queueWatcher .update.verbose</td>
<td>Provides queue state update debug.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.scheduleQueue .verbose</td>
<td>Provides general schedule queue information that can be used to debug problems.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
<tr>
<td>wt.queue.scheduleQueue .execEntries.verbose</td>
<td>Provides debug information related to the execution of a set of schedule queue entries.</td>
</tr>
<tr>
<td></td>
<td>Default is false.</td>
</tr>
</tbody>
</table>
There are other Windchill properties specific to queuing. The following table includes two examples:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.queue.scheduleQueue.execEntry.verbose</td>
<td>Provides debug regarding the execution of individual schedule queue entry. Default is false.</td>
</tr>
<tr>
<td>wt.queue.scheduleQueueEntry.verbose</td>
<td>Provides execute debug for schedule queue entries. Default is false.</td>
</tr>
<tr>
<td>wt.queue.scheduleQueueThread.verbose</td>
<td>Provides debug information related to the actual scheduling threads. Default is false.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt.index.defaultQueueInterval</td>
<td>Specifies the number of seconds in the time-out interval of the index queue polling thread. Default is 60.</td>
</tr>
<tr>
<td>wt.index.useQueue</td>
<td>Specifies whether indexing tasks are moved to the background queue. If this property is set to false, indexing tasks are processed immediately. Default is true.</td>
</tr>
</tbody>
</table>

See properties.html for descriptions of all available properties.
Regular Queue Maintenance

Regular Queue maintenance is important for system performance. Failed and severe entries can accumulate, resulting in large queue tables and failure conditions.

View failed and severe entries on a regular basis and either remove them or reset them to ready. At the point of production, it is a good idea to do so on a weekly basis. When you are more familiar with the patterns of your particular site, you can alter that schedule appropriately.

As the administrator, you must decide whether failed and severe entries can safely be deleted or must be reset to ready. This depends on the particular queue and on your site.

Note: By default, completed entries are removed from the queue; however, if your site is set to retain them, you also need to remove completed entries as part of regular maintenance.

Maintaining the Queue

The Queue Manager utility is part of the System Configurator. For information about opening the System Configurator, see Using the System Configurator.

Follow these steps to develop a regular routine for queue maintenance:

1. Access the Queue Manager utility by opening the System Configurator and then selecting the Queue Manager tab.

2. Locate the row corresponding to the queue for which you want to do the regular maintenance.

3. From the Actions drop-down list in the corresponding row, select View Entries.

   The Entries window opens.

4. From the View entries with status drop-down list, select Failed and then click Show to view failed entries.

5. Examine each failed entry and decide whether to delete the entry or reset it.

   – To remove failed entries, go to the Delete entries with status drop-down list and select Failed. Then click Delete to delete all failed entries.

   Alternately, select the entries to be deleted by selecting the check box in front of each entry in the table, and then clicking Delete at the top of the table to delete selected entries.

   – To reset failed entries, go to Change all entries from - to and select appropriate from and to status codes. Then click Update to make the change.
6. From the View entries with status drop-down list, select Severe and then click Show to view severe entries.

7. Examine each severe entry and decide whether to delete the entry or reset it.
   – To remove severe entries, go to the Delete entries with status drop-down list and select Severe. Then click Delete to delete all severe entries.

   Alternately, select the entries to be deleted by selecting the check box in front of each entry in the table, and then clicking Delete at the top of the table to delete selected entries.

   – To reset severe entries, go to Change all entries from - to and select appropriate from and to status codes. Then click Update to make the change.

8. Repeat the process until you have either deleted or reset all failed and severe entries in each queue.

9. When the status of an entry becomes severe, Windchill stops the queue from which the task for the entry was executed so that no other tasks will execute. If there were severe entries, manually restart the corresponding queue or restart Windchill.

   To restart a queue, go to the Queue Manager main page and select the check box in front of the corresponding queue row in the table and then click Start at the top of the table.

   Alternately, select Start from the Actions drop-down list in the corresponding queue row or from the Queue Properties page for the queue.
Administering RetrievalWare Libraries

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About RetrievalWare Libraries</td>
<td>9-2</td>
</tr>
<tr>
<td>Defining a RetrievalWare Library</td>
<td>9-2</td>
</tr>
<tr>
<td>About Indexing</td>
<td>9-3</td>
</tr>
<tr>
<td>Bulk Loading</td>
<td>9-6</td>
</tr>
<tr>
<td>Configuring Multiple RetrievalWare Libraries</td>
<td>9-8</td>
</tr>
<tr>
<td>Purging a RetrievalWare Library</td>
<td>9-12</td>
</tr>
<tr>
<td>Language Processing</td>
<td>9-12</td>
</tr>
<tr>
<td>Windchill Properties Used to Configure Indexing</td>
<td>9-13</td>
</tr>
</tbody>
</table>
About RetrievalWare Libraries

The Convera RetrievalWare term for index is library.

Indexing is the process of extracting text strings of attribute names and attribute values from Windchill objects and sending them to a search engine that builds RetrievalWare Libraries optimized for searching. This enables users to efficiently search for data stored in a Windchill database without having to know anything about the internal object model.

The Windchill library refers to the Windchill business object. RetrievalWare library refers to indexed libraries in general (not specific to Windchill or RetrievalWare).

Defining a RetrievalWare Library

In the wt.properties file, the Windchill libraries are specified with the property wt.index.collections. Each library has properties that define the following:

• Where the information in the library is stored.
• The language associated with the information.
• The character encoding used when transferring the information to the search engine.
• A user that the indexing service acts on behalf of, objects are indexed into a library.

The properties that follow define a Windchill library. Each is described in the properties table in the Windchill Properties Used to Configure Indexing section, at the end of this chapter.

wt.index.collections
wt.index.Windchill_Business_Collection.collectionName
wt.index.Windchill_Business_Collection.encoding
wt.index.Windchill_Business_Collection.locale
wt.index.Windchill_Business_Collection.user
wt.index.Windchill_Business_Collection.rwareDocHandlerAddress
wt.index.Windchill_Business_Collection.rwareLibName
wt.index.Windchill_Business_Collection.rwareWebServerURL

The default wt.properties file has a Windchill_Business_Collection library defined. The properties for this library are as follows:

wt.index.collections=Windchill_Business_Collection
wt.index.Windchill_Business_Collection.user=Administrator
Windchill_Business_Collection sends information to a RetrievalWare library named wb_lib. A Convera server process, which is located with the address cqdh@localhost: 5327, serves the library wb_lib.

**About Indexing**

Before the indexing process can begin, the Index Policy Manager subscribes to all events that may cause an object's entry in a search engine index to become stale. The list of events follows:

- PersistenceManagerEvent.POST_DELETE
- PersistenceManagerEvent.POST_STORE
- PersistenceManagerEvent.POST_MODIFY
- LifeCycleServiceEvent.STATE_CHANGE
- FolderServiceEvent.POST_CHANGE_FOLDER
- WorkInProgressServiceEvent.POST_CHECKIN
- SessionIterationEvent.POST_COMMIT_SESSION_ITERATION
- ContentServiceEvent.POST_UPLOAD
- IdentityServiceEvent.POST_CHANGE_IDENTITY
The following figure is an overview of how a domain’s indexing policy is enforced.

1. First, an event happens (1). If it is one of the events for which the Index Policy Manager is *listening*, the event and the object are dispatched to the Manager. The Index Policy Manager checks to determine whether the event triggers an indexing action. In many cases it does not (for example, if the object is not indexable or there is no list for the class/state combination). In this case, the event can be ignored (3). In some cases, there is an indexing list for the object/state/event combination. When a list exists, the manager queues the indexing request for deferred processing in a FIFO queue (3).

2. Later, the requests are asynchronously executed, at which time introspection is used to get strings representing all of the attribute names and values that the object holds. This information is then sent to the search engine to be indexed along with any content files that the object holds (4). For more information about background processing queues and their maintenance, see Configuring and Administering Background Queues.
3. When all the metadata is collected, a file similar to the following, is written to the file system:

```xml
<000121Windchill>
<fields>
field Description Sample requirements template
field docTitle ((null))
field ObjectIdentifier VR:wt.doc.WTDocument:5614
field Identity 2002.05.09.07:48:52.890 - Requirements Template, A
field SystemId Windchill
field Name Requirements Template
field TeamTemplate Default
field Number 2002.05.09.07:48:52.890
field LifeCycleState In Work
field Date 6/9/03 10:58 AM
field AppOID wt.content.ApplicationData:5647
field StandardIcon wt/clients/images/msword.gif
field PersistInfoOID VR:wt.doc.WTDocument:5614:951665092-1055173946531-19458623-177-10-253-132@hostname.mycompany.com
field BusinessType Document
</fields>
```

Branch Identifier 5614 Business Type WTDocument Cabinet System Conceptual Classname wt.doc.WTDocument Context Name DefaultOrg Created 6/9/03 10:58 AM Creator Full Name admin Creator Name Administrator Department Engineering Description Sample requirements template Type Template Type Template Folder Path /System/Requirements Template Format Name Microsoft Word Identity 2002.05.09.07:48:52.890 - Requirements Template, A Life Cycle Default No Routing State In Work Location /System Modifier Full Name admin Modifier Name Administrator Last Updated 6/9/03 10:58 AM Name Requirements Template Number 2002.05.09.07:48:52.890 Organization Name DefaultOrg Team Identity 2002.05.09.07:48:52.890 - Requirements Template A2 Team Name 2002.05.09.07:48:52.890 - Requirements Template A2 Team Template Identity Default Team Template Name Default Title ((null)) Type Document PRIMARY Microsoft Word Requirements_Template.doc

```xml
<file><title>Requirements_Template.doc</title><appOID>wt.content.ApplicationData:5647</appOID>C:\ptc\Windchill\temp\contentFile22.bin</file>
```

After this file is written, Windchill requests that Convera index the file. Convera then parses this file using commands from the custom document parser command file wb_lib.dp, which comes with the windchill_indexes working directory. When Convera finds the `<file>` tag, it indexes the file (identified by the full path name within this tag) as a child document of the metadata document. This makes it possible for all entries within the search engine index associated with a Windchill object to be deleted when the Windchill object is deleted. Windchill only needs to remember information about the metadata document’s key to delete it and all of its children from the index.
Bulk Loading

You should use the Bulk Index Tool to load RetrievalWare Libraries and their objects:

- To build indexes of existing data that belong in an index according to index policy. This is part of the process of upgrading from RetrievalWare 6.8 to RetrievalWare 7.0.
- To reinitialize a RetrievalWare library from a failed RetrievalWare system.
- To reinitialize a RetrievalWare library after changes have been made to the indexing policy.

If you are loading RetrievalWare Libraries from a legacy Windchill system, or reinitializing a RetrievalWare library from a failed system, see the section entitled Bulk Loading a RetrievalWare library.

Bulk Loading a RetrievalWare Library

You can use the Bulk Index Tool to load all the objects that belong in the RetrievalWare Libraries. This utility sends objects to a search engine to be indexed according to their domain’s indexing policy.

You can perform the following tasks with this utility:

- Start and stop the bulk indexing process. Because loading indexes can take a significant amount of time, it may be necessary to stop the operation for some length of time. State is maintained in the BulkIndexListEntry table, which is created by this tool, so the process can be stopped and restarted without having to reindex objects that have already been indexed.
- Schedule the process to start and stop at specified times.
- Check on the status of the overall bulk indexing process.
- Attempt to reindex objects that have failed the indexing process.
- Maintain a detailed log of the indexing process.

**Note:** The Bulk Index Tool can only be used to load RetrievalWare libraries.

This tool takes a snapshot of a indexable objects in Windchill’s database. The snapshot is taken at a particular time, and a list of all of the indexable objects residing in the database are stored in the BulkIndexListEntry table. The bulk indexing process consists of looping through this table and determining whether each object belongs in a collection(s) according to the index policy of the domain to which the object belongs. If so, the object is indexed into the appropriate collections.

You can start, stop, and schedule this bulk indexing process.
Note: You can open two command prompts, side by side, to simplify the process of running the tool. Use one command prompt to run the Bulk Index Tool and the other command prompt to tail the BulkIndexTool.log file. The tail utility is a standard UNIX utility. For more information, see the man page. This utility is also available for NT from GNU at http://www.gnu.org.

For real-time progress, you can run the tail utility on the BulkIndexTool.log file. However, this is not required.

The following prerequisites must be met before you can bulk load a RetrievalWare library:

- The index policy rules for your site must be in place.
- The RetrievalWare Libraries that appear in index policy rules must be properly configured.

When these prerequisites are met, you can run the Bulk Index Tool.

To run the Bulk Index Tool, type the following at a command prompt, and log in as a user from the Administrator user group:

```
windchill wt.index.BulkIndexTool
```

Below is a discussion of each of the ten Bulk Index Tool menu options:

1. Create the entire list of objects to index:

   Select option 1 when you first run the Bulk Index Tool so you can build the BulkIndexListEntry table containing all the indexable objects to be processed. Because the query for all indexable objects is very large, it may take a significant amount of time to build the table.

   The status of the overall bulk indexing process will be displayed whenever someone logs in to the Bulk Index Tool. The BulkIndexListEntry table is built after the first time the tool is run.

2. Create a list of objects to index based on current policy rules

   Select option 2 if you want to build the BulkIndexListEntry table based on the current Windchill PDM policy rules.

3. Create a list of objects to index that have not been indexed to date

   Select option 3 if you want to build an BulkIndexListEntry table that only contains items that have not yet been indexed.

4. Create a list of objects to index based on current policy rules that have not been indexed to date

   Select option 4 if you want to build an BulkIndexListEntry table based on the current Windchill PDM policy rules that only contains items which have not yet been indexed.
5. **Start the bulk indexing process:**
   If the tool has not been run, select option 2 to start the bulk indexing process. Select option 6 to check progress.

6. **Stop the bulk indexing process:**
   Select option 3 to stop the bulk index loading process.

7. **Schedule the bulk indexing process:**
   Select option 4 to set up a regular schedule to repeat the bulk indexing process. You may want to schedule this time during low user activity.
   - You must enter the following information:
   - Start time in the format `mm/dd/yyyy hh:mm am/pm`.
   - Stop time in the same format.
   - Total number of runs (how many times you want the scheduled task repeated).
   - Frequency (in days) that you want the bulk indexing task to run. (For example, enter 1 for daily; enter 7 for weekly.)

8. **Retry failed objects:**
   Select option 5 to reset the objects that failed during indexing, so they can be processed again.

9. **Check the bulk indexing progress:**
   Select option 6 to view indexing status.
   The following status example indicates that 2 out of 500 objects have been indexed and that no objects have failed.

   ```
   Current status of Bulk Indexing:
   Total objects: 500.0
   Objects processed: 2.0
   Objects failed: 0.0
   ```

   When all objects have been processed, the bulk indexing process is complete.

10. **Exit:**
    Select option 7 to close the Bulk Index Tool.

### Configuring Multiple RetrievalWare Libraries

RetrievalWare provides a wizard to lead you through the steps of creating an additional RetrievalWare library into which objects can be indexed. Instructions are provided on each screen. However, you should follow the Windchill-specific
instructions below, to access the RetrievalWare Administration (RWA) Wizard and set up a library.

1. If the RetrievalWare Administration servers are not running, start them by accessing the System Utilities menu and select the following options:
   - Administration Servers - Start/Stop/Restart/Status (option 2).
   - Start servers in background (option 3).

2. Enter the following URL in a browser to start the RWA Wizard:

   http://<host>/rware/home.html

3. Log into the RWA Wizard in the lower portion of the screen.
   a. Enter a user name and any password. (Security is not turned on, so the password does not matter.)
   b. Select Other for working directory, and enter the path to the windchill_indexes working directory that you installed with the wt.index.setup.RwareWorkDirInstaller utility.
   c. Click Login.

4. Click Library Setup.

5. Click Set up a new library to open the Library Setup Checklist, and complete the following information on the designated pages. (You can skip Configure Fields and Specify Files to Index.)

   - Name the Library
     • Enter a user-friendly name, a description, and an owner.

   - Set Library Parameter
     If the library’s default language is not English, select the check box labeled Allow 8 bit characters to be indexed (near the bottom of the page). Skip the other fields and click Save.

   - Set Default Language for Library
     Set the language to be used when indexing and querying for documents. Each language plug-in that you have installed is offered as a choice. For more information on language plug-ins, see the Convera RetrievalWare Language Plug-ins Toolkit Guide.

Write this Library Setup to the Configuration File

After you click this link, click Save Settings & Return to the Library Setup Menu to reopen the RetrievalWare library Setup Wizard.

6. Click I'm done, to open the RWA Wizards where you can configure search and index servers for the new library.
7. Click **Server Setup**.

8. Click **Set Up or Modify Your Server Processes**, to open the RetrievalWare Server Setup Wizard.

9. Click **Set Up Servers for Specific Libraries**.

10. Click **Change Settings**, and select the **Text Search Servers** and **Index Server** check boxes to give the new library the same settings as wb_lib, and click **Done**. An **Edit** hyperlink appears.

11. Click the **EDIT** hyperlink, and set the following index server parameters:
   - Set the maximum time between index reformats to 3 minutes.
   - Set the minimum time between index reformats to 1 minute.

   Click the **Save link at the bottom of the page**.

12. Click **Done** on the next page that opens.

13. Click **Generate the Configuration File**, and a red checkmark appears in the check box.

14. Click **Save Settings and return to Server Setup menu**.

15. Restart the search and indexing servers using the **System Utilities** menu.
16. Finally, you must move the wb_lib.dp file into the new library. There you will rename it, using the file name of the document parser command file. To accomplish this, follow the following procedure:

- Delete the new_lib.dp file.
- Open the wb_lib.dp file in the editor.
- Save the wb_lib.dp file as new_lib.dp.

For example, if a new library was created, new_lib, the \windchill_indexes\dp directory should contain a new file called new_lib.dp. This is the document parser command file that will be used when indexing documents into this library. To work with Windchill, the parser command file needs to contain the custom parser commands from the wb_lib.dp file, which is why the wb_lib.dp file is saved as new_lib.dp.

**Editing wt.properties Settings**

The wt.properties file can now be edited using the xconfmanager utility to send information to be indexed to the new library. For details, see Using the xconfmanager Utility.

The following properties are an example of what the property settings defining the original default Windchill_Business_Collection and a new Parts_Only_Collection would look like. The information belonging to the Windchill_Business_Collection is stored in a RetrievalWare library called wb_lib, which is serviced by a document handler at address cqdh@localhost:5327. The information belonging to the Parts_Only_Collection is stored in a RetrievalWare library called part_lib that is serviced by a document handler at address cqdh@localhost:5327

```
wt.index.Collections=Windchill_Business_Collection,Parts_Only_Collection
wt.index.Windchill_Business_Collection.user=Administrator
wt.index.Windchill_Business_Collection.collectionName=Windchill_Business_Collection
wt.index.Windchill_Business_Collection.encoding=8859_1
wt.index.Windchill_Business_Collection.rwareDocHandlerAddress=cqdh@localhost:5327
wt.index.Windchill_Business_Collection.rwareLibName=wb_lib
wt.index.Windchill_Business_Collection.saveFiles=true
wt.index.Parts_Only_Collection.user=Administrator
wt.index.Parts_Only_Collection.collectionName=Parts_Only_Collection
wt.index.Parts_Only_Collection.encoding=8859_1
```
To purge a library’s indexes follow this procedure.

1. Open the RetrievalWare System Utilities menu.
   - For NT: Open the start menu and select **Programs > RetrievalWare 6.7 > System Utilities Menu.**
   - For UNIX: Execute the `<RwareHome>/bin/menu` script.

2. Select Indexing and Index Utilities (option 6).

3. Select Delete and Create Empty indexes for a Library (option 11).

4. Enter the name of the library to purge.

**Language Processing**

Language must be specified in Windchill to determine which locale is used for getting display strings of enumerated types and which encoding is used for writing the metadata file.

In the following two properties, `<collectionName>` is one of the collections defined by the property `wt.index.collections`:

- The locale used when getting display strings of enumerated types is defined by the property `wt.index.<collectionName>.locale`.
- The encoding used when writing the metadata file to disk is defined by the property `wt.index.<collectionName>.encoding`.

The table that follows specifies how to set up locale and encoding properties for your collection, depending on the language with which you want the metadata file to be written. (Content files do not have translations available and are streamed from the Windchill database to the file system as is.)

<table>
<thead>
<tr>
<th>Language</th>
<th><code>wt.index.&lt;collectionName&gt;.locale</code></th>
<th><code>wt.index.collectionName.encoding</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese Traditional</td>
<td>zh_TW</td>
<td>Big5</td>
</tr>
<tr>
<td>Chinese Simplified</td>
<td>zh_CN</td>
<td>gb2312</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
<td>8859_15</td>
</tr>
</tbody>
</table>
For more information on administering RetrievalWare language plugins, including installation instructions for each of the language plug-ins, see the Language Plug-Ins Toolkit Guide on the RetrievalWare CD.

## Windchill Properties Used to Configure Indexing

Set the properties described in the following table using the xconfmanager utility to configure the indexing capability of your Windchill system. For xconfmanager details, see Using the xconfmanager Utility.

All the properties are defined in the wt.properties file.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| wt.index.collections   | **Default:** Windchill_Business_Collection  
**Synopsis:** Windchill collections.  
**Description:** This is a comma-separated list of collections that appear as choices when creating an indexing rule. A collection is information that is stored in a search engine's database, allowing for efficient searching of that information. The following properties are then used to define the collection, where `<collectionName>` is one of the elements from the comma-separated list of collections: |
|                        | `wt.index.<collectionName>.collectionName`  
`wt.index.<collectionName>.collectionPath`  
`wt.index.<collectionName>.encoding`  
`wt.index.<collectionName>.locale`  
`wt.index.<collectionName>.rwareDocHandlerAddress`  
`wt.index.<collectionName>.rwareLibName`  
`wt.index.<collectionName>.rwareWebServerURL`  
`wt.index.<collectionName>.saveFiles`  
`wt.index.<collectionName>.user`  
`wt.index.<collectionName>.webServerURL` |
| wt.index.enabled       | **Default:** true  
**Synopsis:** Enables (true) and disables (false) indexing.  
**Description:** Indexing should be disabled only for debugging. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| wt.index.defaultQueueInterval                | **Default**: 60 seconds  
**Synopsis**: Initial sleep or time-out interval for the index queue polling thread. |
| wt.index.tempFileDir                         | **Default**: c:\temp  
**Synopsis**: Temporary directory for intermediate files.  
**Description**: Files are extracted from the Windchill database and placed in this directory on the file system so that the RetrievalWare search engine can index them. |
| wt.index.Windchill_Business_Collection.collectionName | **Default**: Windchill_Business_Collection  
**Synopsis**: Collection name.  
**Description**: This is the name of the collection Windchill_Business_Collection. |
| wt.index.Windchill_Business_Collection.encoding | **Default**: No default value.  
**Synopsis**: The encoding used for transferring stream.  
**Description**: This encoding should be set appropriately depending on the locale of the collection:  
Chinese Simplified = UTF8  
Chinese Traditional = UTF8  
Japanese = SJIS  
Korean = EUC_KR |
| wt.index.Windchill_Business_Collection.locale | **Default**: No default value.  
**Synopsis**: The locale to use when indexing enumerated types.  
**Description**: There is no need to set this if you are indexing in English.  
de = German  
fr = French  
it = Italian  
ja = Japanese  
ko = Korean  
zh_CN = Chinese Simplified  
zh_TW = Chinese Traditional |
| wt.index.Windchill_Business_Collection.user   | **Default**: Administrator  
**Synopsis**: User whose rights determine access control of objects being indexed.  
**Description**: If this user does not have read access to an object for which an indexing rule applies, the object is not indexed. |
When RetrievalWare is installed on a remote server, a wt.properties file is used to configure logging and to specify a temporary directory to use for intermediate files. This wt.properties file is located in the codebase directory of the directory.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| wt.index.Windchill_Business_Collection.rwareDocHandlerAddress | **Default**: cqdh@localhost:5327  
**Synopsis**: The address of a RetrievalWare document handler.  
**Description**: When a rule specifying the collection Windchill_Business_Collection, applies to an object, an attempt is made to send an indexing request to the RetrievalWare document handler at this address. |
| wt.index.Windchill_Business_Collection.rwareLibName | **Default**: wb_lib  
**Synopsis**: The name of a RetrievalWare library that should hold information belonging to the Windchill_Business_Collection.  
**Description**: When a rule specifying the collection Windchill_Business_Collection applies to an object, an attempt is made to index the object into the RetrievalWare library specified by this property. |
| wt.index.Windchill_Business_Collection.rwareWebServerURL | **Default**: No default value.  
**Synopsis**: Should only be set if indexing to a RetrievalWare server installed on a remote machine.  
**Description**: Should be of the form http://$(java.rmi.server.hostname)/WindchillIndex, where java.rmi.server.hostname is replaced by the hostname of the machine on which RetrievalWare is installed and /WindchillIndex is a servlet alias to the wt.indexloader.IndexLoaderServlet. |
| wt.index.verboseExecution | **Default**: No default value.  
**Synopsis**: Verbosity flag.  
**Description**: Prints debug tracing for index policy operations, such as the creation of rules and lists. |
| wt.index.verbosePolicy | **Default**: No default value.  
**Synopsis**: Verbosity flag.  
**Description**: Prints debug tracing for index policy execution operations, for example, event capturing, dispatching, and processing. |

**Remote RetrievalWare Server wt.properties file**

When RetrievalWare is installed on a remote server, a wt.properties file is used to configure logging and to specify a temporary directory to use for intermediate files. This wt.properties file is located in the codebase directory of the directory.
into which the index search component was installed. The following properties appear in this file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| wt.index.log.append       | **Default:** true  
**Synopsis:** Append flag.  
**Description:** Determines whether the indexloader log is appended to or overwritten during each indexing request. |
| wt.index.log.enabled      | **Default:** true  
**Synopsis:** Master switch for enabling logging in applications that support it.  
**Description:** Setting to true enables logging; setting to false disables logging. |
| wt.index.log.file         | **Default:** c:\WTSearch\IndexEntryToRware.log  
**Synopsis:** Log file name.  
**Description:** The full pathname to a file where information is to be logged. |
| wt.index.tempFileDir      | **Default:** c:\temp  
**Synopsis:** Temporary directory for intermediate files.  
**Description:** Files are extracted from the Windchill database and placed in this directory on the file system so the RetrievalWare search engine can index them. |
| wt.index.verboseExecution | **Default:** No default value.  
**Synopsis:** Verbosity flag.  
**Description:** Prints debug tracing for index policy operations, such as the creation of rules and lists. |
This chapter presents customization and administration information and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill PDM, Windchill PDMLink, and Windchill ProjectLink. The primary audience is Pro/ENGINEER and Windchill system administrators.

The topics presented include Pro/ENGINEER configuration information (environment variables and config.pro options) that applies to the interaction with Windchill, and Windchill server-side settings, including a general discussion of Windchill INI files, as well as specific information on parameter mapping, parameter customization, customizing object naming, automated part creation, and supporting custom parts. In addition, recommendations for system configuration and performance tuning are offered.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customizing Pro/ENGINEER Wildfire</td>
<td>10-2</td>
</tr>
<tr>
<td>System Configuration Recommendations</td>
<td>10-23</td>
</tr>
<tr>
<td>Performance Tuning</td>
<td>10-24</td>
</tr>
<tr>
<td>Other Recommendations</td>
<td>10-25</td>
</tr>
<tr>
<td>Default Directory INI Files</td>
<td>10-27</td>
</tr>
<tr>
<td>Site Directory INI Files</td>
<td>10-32</td>
</tr>
</tbody>
</table>
Customizing Pro/ENGINEER Wildfire

Environment Variables and Config.pro Options for Pro/ENGINEER Wildfire

Environment Variables

Pro/ENGINEER Wildfire 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 Pro/ENGINEER and the server. The cache (which is managed by Pro/ENGINEER 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 Pro/ENGINEER.

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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 Pro/ENGINEER to use the new location as a location for the cache.  

**Note:** Existing cache data will not be copied to the new location automatically.

| 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.  

**Note:** This new folder only applies to newly created workspaces. Existing workspaces will continue to reside in $PTC_WF_ROOT/.cache |
Config.pro Options

The following table lists Pro/ENGINEER config.pro options that are especially relevant to the Pro/ENGINEER Wildfire interaction with Windchill:

<table>
<thead>
<tr>
<th>Config.pro Option</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm_cache_size</td>
<td>Integer [default = 400]</td>
<td>Sets the size (in MB) of the cache allocated to each workspace on the client hard disk.</td>
</tr>
<tr>
<td>dm_remember_server</td>
<td>YES [default] NO</td>
<td>If this option is set to yes, the last primary server/workspace of a Pro/ENGINEER session will be set automatically for the next Pro/ENGINEER session.</td>
</tr>
<tr>
<td>web_browser_homepage</td>
<td>string value</td>
<td>Sets the location of Pro/ENGINEER browser homepage.</td>
</tr>
<tr>
<td>save_model_display</td>
<td>wireframe, shading_low, shading_high, shading_lod</td>
<td>Sets the quality of graphics that are shown on the Windchill properties page. Setting this option to shading_lod creates the best images, but requires larger Pro/ENGINEER file sizes to store the additional graphical information.</td>
</tr>
<tr>
<td>dm_http_compression_level</td>
<td>Integer, from 0 (no compression) to 9 (maximum compression) [default = 0]</td>
<td>Sets the level of compression for data upload and download. 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Type and Default</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| dm_network_retries | integer >0 [default = 10 ] | Sets the number of attempts to connect to a Windchill server before the connection is considered broken.  
(This is a hidden config option that was introduced in Pro/ENGINEER Wildfire date code 2003210.)  
Recommended setting: default  
Note: 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. |
| dm_network_threads | integer >0 [default = 3] | Sets the number of concurrent threads Pro/ENGINEER uses for downloading data from a Windchill server.  
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.  
(This is a hidden config option that was introduced in Pro/ENGINEER Wildfire date code 2003210.)  
Recommended setting: default |
| dm_upload_objects | explicit [default] automatic | Defines the behavior of the **Save** command in Pro/ENGINEER.  
If this option is set to explicit, the Pro/ENGINEER **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**, **File > Upload**, or **File > Checkin**). If this option is set to automatic, **File > Save** in Pro/ENGINEER will also upload the Pro/ENGINEER files to the server. |
| dm_secondary_upload | automatic [default] explicit | Defines the behavior of saving to a secondary server (See also dm_upload_objects). |
Note: In Pro/ENGINEER Wildfire 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 Pro/ENGINEER search path.

Setting File Retrieval Options
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 PDMLink library cabinet using the following syntax:

```
start_model_dir wtpub://<server_alias>/Libraries/<library_name>
```

Similarly, the value of pro_group_dir is set to point to a Windchill PDMLink 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 Pro/ENGINEER 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
```

**INI Files**

**About INI Files**

INI files are used to set certain preferences on the Windchill server. Typical uses are to let the system create a customized WPart, or to choose the date format to be displayed in the user interface.

Generally, an administrator modifies INI files and a restart of the method server is required before the changes take effect.

**Location of INI Files**

Many INI files are used to configure different aspects of the Windchill server. The INI files used to configure the Pro/ENGINEER Wildfire services can be found in:

```
$WT_HOME/codebase/com/ptc/windchill/cadx/cfg/
```

There are six INI files that are installed by default under the following subdirectories:

- `$WT_HOME/codebase/com/ptc/windchill/cadx/cfg/site`/autopart.ini, autoassociate.ini, newdocument.ini

  **Note:** Files under the site directory are used for setting preferences applicable to the site and cannot be overridden except by the administrator.

- `$WT_HOME/codebase/com/ptc/windchill/cadx/cfg/default`/cadxhtmlui.ini, console.ini, newdocument.ini

  **Note:** Files under the default directory are used for setting default preferences.

**Structure of an INI File**

INI files have the following characteristics:

- The *section* is defined in square brackets, for example: `[general]`.

- The *key-value* pairs are listed as `<key>=<value>`, for example: `DefaultDateFormat=ShortDateFormat1`

- The semicolon character (`;`) is used for comments. (For many key-value pairs, comments are available in the INI file indicating their use and possible values.)
Example -- A sample newdocument.ini file:

[general]
; allow download of an already checkout object
okToDownloadAlreadyCheckedOut=true
; for a newly created CAD Doc the destination folder
; where the document may be moved upon checkin
DefaultDocFolder=/<<Name of the folder>>
; set to true by default for Pro/E
isModelNameUnique=true

[proe]
; Valid Pro/E model file name extensions.
proe.files.component.ext=.prt
proe.files.assembly.ext=.asm
proe.files.drawing.ext=.drw
proe.files.diagram.ext=.dgm
proe.files.format.ext=.frm
proe.files.layout.ext=.lay
proe.files.manufacturing.ext=.mfg
proe.files.markup.ext=.mrk
proe.files.report.ext=.rep
proe.files.sketch.ext=.sec

;no specific extension for "other"

CADComponentDocNumber=00008_comp_1.prt
CADAsemblyDocNumber=00008_asm_1.asm
CADDrawingDocNumber=00008_drw.drw
DiagramDocNumber=00008_dgm.dgm
FormatDocNumber=0008_frm.frm
LayoutDocNumber=00008_lay.lay
ManufacturingDocNumber=0008_mfg.mfg
MarkupDocNumber=0008_mrk.mrk
ReportDocNumber=00008_rep.rep
SketchDocNumber=0008_sec.sec
OtherDocNumber=00008_other.oth

Note: The parameter isModelNameUnique=true should not be changed.

Note: New document templates are specified by replacing existing *DocNumber
settings with the number of your chosen template document.

Overriding INI File Settings

To override the default preferences of an INI file, create a user-modified version
of the file in the user’s personal cabinet:

Note: The following procedure applies to Windchill PDM only (user cabinets are
not directly accessible in Windchill PDMLink or Windchill ProjectLink).

1. Upload the selected INI file to the user’s personal cabinet, giving it a name
and number as follows:

The <name>_<number> format of a user preference file is
"<username>_<nameOfIniFile>". For example, if user "sachin" wants to set a preference in "cadxhtmlui.ini," the file he uploads to his personal cabinet is to be named and numbered "sachin_cadxhtmlui."

2. Make modifications to the respective key-value pairs.

   **Note:** If a key is defined more than once, the last entry of the key-value pair is used.

3. Restart the method server to apply the changes made to the preference system.

**About Templates in INI Files**

Pro/ENGINEER Wildfire uses templates (if available) when new documents are created from a workspace. To create a new document using **Create CAD Document**, edit the template CAD document properties under section [proc] in the newdocument.ini file found at WT_HOME\codebase\com\ptc\windchill\cadx\cfg\default.

   **Note:** Template editing using INI files applies to Windchill PDM only. In Windchill PDMLink, CAD document templates are created without configuring the newdocument.ini file by using the **Create CAD document template** action (which is available on the **Administration** tab). For more information, see Windchill PDMLink online help.

For example:

```
; Template document number for creating new component documents
CADComponentDocNumber= template.prt

; Template document number for creating new assembly documents
CADAssemblyDocNumber= template.asm

; Template document number for creating new drawing documents
CADDrawingDocNumber= template.drw
```

   **Note:** These template CAD documents must exist in the Windchill database. If the templates do not exist, Pro/ENGINEER Wildfire creates empty CAD documents known as missing dependents or ghost objects.

**INI File Recommendations**

**Setting a Context in Windchill PDMLink**

When a workspace is created in a Windchill PDMLink server through the Pro/ENGINEER user interface (**Tools > Server Registry**), the context (part or document location) is assigned by default to the first writable library (or product if there are no libraries). Use the workspace property page to change the default settings.
Setting the Workspace Date Format

Pro/ENGINEER Wildfire recommends that the cadxhtmlui.ini setting ShortDateFormat1=4 be used to set an appropriate workspace column width and avoid possible display issues. The date format used can be customized by modifying entries in section [general] of the cadxhtmlui.ini file as follows:

[general]

DefaultDateFormat = ShortDateFormat1

Setting the above entry sets the date format as in 15 Mar 02 13:25. Please note that the value in the above key-value pair comes from the section [dateformat] in the same file, which means that if the date format expected is 3/15/02 1:25 PM, then the value of DefaultDateFormat should be set as follows:

DefaultDateFormat = ShortDateFormat2.

Users may not modify the key-value pairs in the [dateformat] section unless they customize wt.util.utilResource_<locale>.rbInfo to add more date formats than the out-of-the-box date formats. In such a case, the new entries to be added to the [dateformat] section should follow the pattern outlined below:

1. Create a key of your choice, for example, customDateFormat.

2. Set the value of this key to the resource key from the utilResource.rbInfo file (for example, if you created a date format called “myDateFormat” in wt.util.utilResource_<locale>.rbInfo). (For more information on this kind of customization, see the Windchill Customizer’s Guide). For example:

100.constant= myDateFormat
100.value=dd-MM-yy

The entry in section [dateformat] of cadxhtmlui.ini will be as follows:

;Date format as "dd-MM-yy".
;Check "myDateFormat" key in wt.util.utilResource_<locale>.rbInfo for exact locale specific format
;For example, 03/12/02
customDateFormat=100

The entry in section [general] of cadxhtmlui.ini will be as follows:

DefaultDateFormat = customDateFormat

Mapping Pro/ENGINEER Parameters to Windchill Instance-Based Attributes

Pro/ENGINEER Wildfire lets you map Pro/ENGINEER designated parameters onto Windchill instance-based attributes (IBAs). Attribute mapping transfers parametric information from the CAD models created in Pro/ENGINEER to the Windchill system. The attribute mapping can be done as follows:

• Explicit parameter-to-attribute mapping
• Implicit parameter-to-attribute mapping
Explicit Parameter-to-Attribute Mapping

In order to explicitly map Pro/ENGINEER designated parameters to Windchill IBAs, entries must be added to the iba.properties file on the server. The iba.properties file can be found at the following location:

\$WT_HOME\codebase\com\ptc\prowt\proesvcs\util

Using the xconfmanager, add a line to the iba.properties file, in the following format:

<Pro/ENGINEER parameter name>=<Windchill IBA name>

Examples:

• To map a Pro/ENGINEER designated parameter MCOST to a leaf node IBA (with no children) named MfgCost in Windchill, the entry is as follows:

  MCOST=MfgCost

• To map a Pro/ENGINEER designated parameter named VENDORCOST onto a child node IBA named VendorCost (with a parent IBA named TotalCost) in Windchill, the entry is as follows:

  VENDORCOST=TotalCost|VendorCost

Parameter names in Pro/ENGINEER are not case-sensitive; however, the mapping in the iba.properties file must contain the name of the parameter in uppercase. Windchill IBAs are case-sensitive and should always be mapped accurately for the mapping to work correctly.

Implicit Parameter-to-Attribute Mapping

Implicit parameter-to-attribute mapping occurs when there is an IBA in Windchill with a name (all uppercase) identical to the name of a designated parameter in a Pro/ENGINEER model file and there is no entry for the Pro/ENGINEER designated parameter in the iba.properties file. When the Pro/ENGINEER model file is uploaded into Windchill as content of a CAD document, the values of the Pro/ENGINEER parameter are transferred to the Windchill IBA.

Upload Behavior for Attribute Mapping

On upload, the service first looks for a mapping in the iba.properties file. The Pro/ENGINEER designated parameter name in the properties file must always be in upper case. If a mapping does not exist, or if the properties file itself does not exist, the Pro/ENGINEER designated parameter is converted to upper case and the service looks for an IBA definition by this name. If an IBA definition is not found, the upload service reports a conflict. All conflicts are reported in the Event Console in terms of the Pro/ENGINEER designated parameter name.
Download Behavior for Attribute Mapping

On download, the reverse is done; that is, the IBA name is mapped to the Pro/ENGINEER designated parameter name. It is possible that two or more Pro/ENGINEER designated parameters map to a single IBA. In this case, the Pro/ENGINEER designated parameter that is last among those that map to a common IBA is chosen.

Note: Once added to the iba.properties file on the server side, IBA-to-designated parameter mappings should not be removed or modified. Doing so might affect the models already stored in the Windchill database (including historical data and released designs), particularly if such models have IBA values modified through the Windchill user interface (as opposed to modifications done in a Pro/ENGINEER session).

For the same reason, new mappings should be added only for Pro/ENGINEER designated parameters that do not exist yet in the models already stored in the Windchill database. Therefore, it is strongly recommended that:

- The Windchill server administrator adds any new mappings as part of the process of adding new IBA definitions in Windchill.
- Workgroup Manager for Pro/ENGINEER and Pro/ENGINEER Wildfire PDM users do not designate Pro/ENGINEER parameters which are not mapped (implicitly or explicitly) onto a Windchill IBA; the upload service provides warnings in the Event Console in such cases, and users are encouraged to undesignate the parameters in Pro/ENGINEER and upload the models again.

Customizing the Parameters in the Download Service

Windchill provides a server-side delegate that can be used to insert parameters into a Pro/ENGINEER model upon download. This mechanism can be used to pass information from the server down to Pro/ENGINEER, where it can be used like any other Pro/ENGINEER 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: The customized parameters are provided to the client upon download and, unlike system parameters such as PTC_WM_ITERATION, are not updated in the Pro/ENGINEER 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 Pro/ENGINEER 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:

```java
package com.ptc.prowt.proesvcs.util;
public interface ModeledAttributesDelegate {
    /*
     * Implement this API to determine if the parameters that are
     * added by the customization (as in getModeledAttributes() below), must be stored in Windchill on upload. Typically,
     * you would not want the customized parameters to be stored in
     * Windchill. This API is invoked by the upload service. The
     * input parameter passed is the parameter name. The
     * implementation must return true if the parameter must not be
     * persisted in Windchill on upload/checkin. Else, return false
     * @param definition The name of the parameter
     * @return true if the parameter must not be persisted in
     * Windchill on upload, else false
     */
    public boolean isModeledAttribute(String definition);
    /*
     * This is the API, invoked by the download service on download,
     * to be implemented for the customization. Create and return a
     * HashMap of parameter name- value pairs that must be
     * propagated to Pro/E par represented by the EPMDocument (the
     * input parameter to this API). Use the getCADName() API on the
     * EPMDocument to identify the Pro/E part.
     * @param object The EPMDocument representing the Pro/E part to
     * which parameters and values are added.
     * @return a map of parameter name/value pairs. The key is the
     * parameter name and the value is the parameter value
     */
    public HashMap getModeledAttributes(Object object) throws WTException;
}
```

2. In the service.properties file (found in $WT_HOME/codebase), use the xconfmanager to update the following line:

```
```

Use the full path to your class (that is, replace `com.ptc.prowt.proesvcs.util.DefaultModeledAttributesDelegate` with the path to your class):

```
wt.services/svc/default/com.ptc.prowt.proesvcs.util.ModeledAttributesDelegate/null/java.lang.Object/0=<the full path to your class>/singleton
```

3. Restart the method server to apply the changes to service.properties.
Customizing the Naming Service

The naming service uses the Windchill service delegate mechanism to allow the
user to specify the following for the new CAD document (EPMDocument) to be
created:

- A number for the EPMDocument
- A name for the EPMDocument
- Parameter values for the EPMDocument

The following steps explain how to customize the naming service:

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

   ```java
   package com.ptc.prowt.proesvcs.namesvc;
   import java.util.ArrayList;
   public interface EPMDocumentNamingDelegate
   {
   /* This method is not yet supported. Provide a dummy
   implementation */
   public void validateDocumentIdentifiers(ArrayList docIdentifier);
   /* Implement the customized business logic for
   naming/numbering EPMDocument(s) in this method */
   public void validateDocumentIdentifier(DocIdentifier docIdentifier);
   }
   ```

   The definition of the class DocIdentifier is as below:

   ```java
   package com.ptc.prowt.proesvcs.namesvc;
   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;
}
/** set the Pro/E designated parameters for the model. These are currently ignored **/
public void setParameters(HashMap params)
{
    m_parameters = params;
}

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

public void validateDocumentIdentifier(DocIdentifier docIdentifier)

– The DocumentIdentifier object has the EPMDocument name, number, and Pro/ENGINEER designated parameters 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 Pro/ENGINEER designated parameters may be used to provide a hint for EPMDocument numbering/naming.

Use DocIdentifier.getParameters() to get the associated parameters.

Use the “set” methods on the DocIdentifier to set the new name, number, and parameter values. The upload service uses these suggestions if they are feasible.
3. In the Service.properties file (found in $WT_HOME/codebase), use the xconfmanager to update the following line:

```java
wt.services/svc/default/com.ptc.prowt.proesvcs.namesvc.EPMDocumentNamingDelegate/null/epm.EPMDocument/0=com.ptc.prowt.proesvcs.namesvc.EPMDocumentNamingDelegate/singleton
```

using the full path to your class (that is, replace `com.ptc.prowt.proesvcs.namesvc.EPMDocumentNamingDelegate` with the path to your class):

```java
wt.services/svc/default/com.ptc.prowt.proesvcs.namesvc.EPMDocumentNamingDelegate/null/epm.EPMDocument/0=<the full path to your class>/singleton
```

4. Restart the method server to apply the changes to Service.properties.

Example:

**Note:** This example implements a customized naming service delegate that sets the name of a CAD Document to the value of the Pro/ENGINEER designated parameter DESCRIPTION. If the DESCRIPTION parameter is not present, the name of the CAD Document is set to the name of the source Pro/ENGINEER model.

```java
package ext.namesvc;
import com.ptc.prowt.proesvcs.util.PrintHelper;
import com.ptc.prowt.proesvcs.namesvc.EPMDocumentNamingDelegate;
import com.ptc.prowt.proesvcs.namesvc.DocIdentifier;
import wt.util.WTProperties;
import java.util.ArrayList;
import java.util.HashMap;
public class EPMDemoDocumentNamingDelegate implements EPMDocumentNamingDelegate {
    private static int _verboseLevel=0;
    private static boolean _verbose=false;
    private static String _docNameParamName = null;
    static {
        try {
            WTProperties wtproperties = WTProperties.getLocalProperties();
            _docNameParamName = wtproperties.getProperty("ext.epmnaming.proeParameterName", "DESCRIPTION");
            _verbose = wtproperties.getProperty("ext.epmnaming.verbose", false);
            _verboseLevel = wtproperties.getProperty("ext.epmnaming.verbose.level", 1);
        }
        catch(Exception exception) {
            exception.printStackTrace();
        }
    }
    public String getName(String docID) {
        if (_docNameParamName != null) {
            String name = wtproperties.getProperty(_docNameParamName,"");
            if (_verbose) {
                PrintHelper.println("EPMDemoDocumentNamingDelegate: Using parameter: ", _docNameParamName);
            }
            return name;
        } else {
            return docID;
        }
    }
}
```
/* This method is not yet supported. Provide a dummy implementation */
public void validateDocumentIdentifiers(ArrayList docIdentifier) {
    printMsg(9,"DemoEPMDocumentNamingDelegat.validateDocumentIdentifiers: This method is not yet supported.");
}

/* Customized business logic for naming/numbering EPMDocument(s)
 * The EPMDocument name will be set to the Pro/E parameter DESCRIPTION
 * If DESCRIPTION does not exist in Pro/E model name will be default (Pro/E object name)
 */
public void validateDocumentIdentifier(DocIdentifier docId) {
    printMsg(1,"****** validateDocumentIdentifier() START");
    HashMap params = docId.getParameters();
    printMsg(2, "NEW EPMDocument ------------------------
----------");
    printMsg(2, "DocName:     "+docId.getDocName());
    printMsg(2, "DocNumber: "+docId.getDocNumber());
    printMsg(2, "ModelName: "+docId.getModelName());
    printMsg(2, "Parameters: "+params );
    printMsg(2, "Looking for parameter: "+_docNameParamName);
    String value = (String)params.get(_docNameParamName);
    if (value != null && value.length() > 0 && validateName(value)) {
        printMsg(1,"Setting EPMDoc name for CAD model "+
        docId.getModelName()+ " to "+value);
        docId.setDocName(value);
    } else {
        printMsg(1,"WARNING Pro/E object "+
        docId.getModelName()+ " does not contain parameter "+_docNameParamName);
    }
    printMsg(1,"****** validateDocumentIdentifier() END");
}

private boolean validateName(String name) {
    String notAllowedChars = "$^*()_}{[\]";  
    for (int i=0; i<notAllowedChars.length();i++) {
        if (name.indexOf(notAllowedChars.charAt(i)) >= 0)
            System.out.println("***** Invalid char "+notAllowedChars.charAt(i)+") found in EPMDocument name.");
            return false;
    }
    return true;
}
private void printMsg(int level, String s) {
    if (_verbose && level <= _verboseLevel)
        System.out.println(s);
}

Enabling Support for Custom Parts

In the Pro/ENGINEER Wildfire 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 Pro/ENGINEER Wildfire HTML client. (For information on extending the Windchill object model, see the Windchill Application Developer’s Guide and the Windchill Customizer's Guide.)

The Pro/ENGINEER Wildfire 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, Create Part and Auto Associate Part, are specific to WTPart. Additionally, when you view the properties of a custom part, any IBAs you may have added to the custom part can be seen; however, newly modeled information is not displayed.

Whenever Part is available in the object type list on the Pro/ENGINEER Wildfire HTML client object selection page, if Part or All is selected both WTPart objects and custom part objects will be 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 AutoAssociate, later in this chapter.

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 “Customizing the HTML Client” in the Windchill Customizer's 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 Pro/ENGINEER Wildfire HTML client object selection page’s default implementation, you must add support for the custom part in the configured wt.query.SearchAttributeListDelegate. (For more details see the section, Customizing the HTML Search, in the “Customizing GUIs” chapter of the Windchill Customizer’s Guide.)

In addition you must modify the Pro/ENGINEER Wildfire HTML files that use the object selection page, and use the xconfmanager modify or override the type list id entries in com\ptc\windchill\cad\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.
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.

Customizing AutoAssociate

AutoAssociate functionality can be customized in the following ways:

• Modifying the implementation of the AutoAssociatePartFinderCreator interface
• Modifying the CAD document IBA value to be used to search for a WTPart with a matching number.
• Modifying the implementation to search for Customized parts or custom parts

Each of these three ways is described in the following sections.

Modifying the Implementation of the AutoAssociatePartFinderCreator Interface

A default implementation of the AutoAssociatePartFinderCreator interface is provided to search for matching parts and create new parts. You can modify the implementation of this interface to allow customized search and create part criteria.

To modify the implementation, do the following:

1. Create "public class CustomizedAutoAssociatePartFinderCreator implements AutoAssociatePartFinderCreator"
2. In $WT_HOME\codebase\com\ptc\windchill\cadx\cfg\site\autoassociate.ini, modify the entry under [General] to read as follows:

[General]

PartFinder=com.ptc.windchill.cadx.autoassociate.
CustomizedAutoAssociatePartFinderCreator

3. Restart the method server

Modifying the CAD Document IBA Value

In $WT_HOME\codebase\com\ptc\windchill\cadx\cfg\site\autoassociate.ini, the default implementation specifies the following:

    SearchForPartAttribute=PARTNUMBER

where PARTNUMBER is an attribute of the CAD document that is read programmatically to get the number of the part to be searched for. You can modify the attribute by doing the following:

1. Create a new attribute (for example, MYATTRIBUTE) in Windchill using the Attribute Manager.

   **Note:** The attribute must be in upper case.

2. In $WT_HOME\codebase\com\ptc\windchill\cadx\cfg\site\autoassociate.ini, modify the entry under [General] to read as follows:

    [General]

    SearchForPartAttribute=MYATTRIBUTE

3. 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 (for example, wt.part.MyCustomPartMaster), do the following:

1. In $WT_HOME\codebase\com\ptc\windchill\cadx\cfg\site\autoassociate.ini, modify the entry under [General] to read as follows:

    [General]

    SearchPartMasterClass= wt.part.MyCustomPartMaster
2. Restart the method server.

Customizing the Latest Configuration Specification for Checkout

During a checkout operation, when the option selected for configuration specification is **Latest**, by default table of selected objects displays only WTParts with no view specified (view == null).

You can customize this behavior by modifying the following entry in $WT_HOME/codebase/com\ptc\windchill\cadx\cfg\default\cadxhtmlui.ini as follows:

```
[config spec policy]
LatestWithViewFromWorkspacePref=true (default is false)
```

Setting this value adds the view set to the workspace general options for Latest configuration specification processing.

Example

For example, say you have the following associations:

<table>
<thead>
<tr>
<th>EPMDocNull</th>
<th>associated with</th>
<th>WTPartNull (view == null)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPMDocDesign</td>
<td>associated with</td>
<td>WTPartDesign (view == Design)</td>
</tr>
<tr>
<td>EPMDocDesign</td>
<td>associated with</td>
<td>WTPartMfg (view == Manufacturing)</td>
</tr>
</tbody>
</table>

Default Behavior

Default behavior, with the INI setting LatestWithViewFromWorkspacePref=false and the checkout page rule for **Include Parts/Docs** set to **All**, is as follows:

- Selecting EPMDocNull for checkout displays EPMDocNull and WTPartNull.
- Selecting EPMDocDesign for checkout displays only EPMDocDesign.
- Selecting EPMDocMfg for checkout displays only EPMDocMfg.

Customized Behavior

Customized behavior, with the INI setting LatestWithViewFromWorkspacePref=true and the checkout page rule **Include Parts/Docs** set to **All**, is as follows:

- With the **View for Parts** on the workspace preferences page **General** tab set to <null>:
  - Behavior is the same as default behavior.
- With **View for Parts** set to **Design**:
– Selecting EPMDocNull for checkout displays EPMDocNull and WTPartNull.
– Selecting EPMDocDesign for checkout displays only EPMDocDesign and WTPartDesign.
– Selecting EPMDocMfg for checkout only displays EPMDocMfg.

**With View for Parts set to Manufacturing:**
– Selecting EPMDocNull for checkout displays EPMDocNull and WTPartNull.
– Selecting EPMDocDesign for checkout displays only EPMDocDesign and WTPartDesign.
– Selecting EPMDocMfg for checkout displays only EPMDocMfg and WTPartMfg.

### Customizing the HTML Client Object Selection Page

The HTML client object selection page is used in the Pro/ENGINEER Wildfire 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 the section, Customizing the HTML Search, in the “Customizing GUIs” chapter of the *Windchill Customizer’s Guide*.)

If this PickerSearchAttributeListDelegate implementation is not sufficient, then you can create and configure your own PickerSearchAttributeList to be used by the object selection page.

### 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:


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 Pro/ENGINEER Wildfire 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-separated 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.

Defining the Rename Report Mail Server
The mail server host name should be defined in the wt.properties file under $WT_HOME/codebase on Unix and %WT_HOME%/codebase directory on Windows platform as follows:
wt.mail.mailhost=localhost

The value of "localhost" should be changed, using the xconfmanager, as per the mail server and domain name, in order to send e-mail through the Rename Report page.

**Generation of Viewables**

Server-side generation of viewables is enabled by setting up the Windchill Visualization Service.

For information about setting up Windchill Visualization Service, see the *Windchill Installation and Configuration Guide - Visualization Services*

**System Configuration Recommendations**

**Running Multiple Servers**

It is recommended that Windchill and Windchill PDMLink 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.

For more information on additional servers, see the Additional Method Servers and Background Method Servers section in *Performance Best Practices for Windchill Pro/ENGINEER Data Management*.

**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.

For more information on external vaulting and vaulting rules see the External Vaulting and Vaulting Rule for EPM Documents sections in *Performance Best Practices for Windchill Pro/ENGINEER Data Management*.

**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 Content Replication For WAN Clients and Local Upload For WAN Clients sections in Performance Best Practices for Windchill Pro/ENGINEER Data Management.

Performance Tuning

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 Pro/ENGINEER data sets that are common to the products developed by Pro/ENGINEER users.

For more information on setting the max heap size, see the Method Server Max Heap Size section in Performance Best Practices for Windchill Pro/ENGINEER Data Management.

Setting the SQL Statement Cache Size

It is recommended that the size and reuse limit of the method server’s SQL statement cache be increased to (field suggestion) 200 - 300 to avoid recomputation of SQL queries.

For more information on setting the size of the SQL statement cache, see the SQL Statement Cache Size section in Performance Best Practices for Windchill Pro/ENGINEER Data Management.

Data Compression

The meta data compression option is intended to improve the upload and download performance of the Pro/ENGINEER Wildfire 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.

In Pro/ENGINEER Wildfire, compression is controlled by a Pro/ENGINEER config.pro setting (dm_http_compression_level) as follows:

\[\text{dm_http_compression_level} \quad < \text{an integer between 0 and 9 -- 0 for no compression, 9 for max compression}>\]

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.

For more information on data compression, see the Upload/Download Metadata Compression Option In R6.2.6 DSU 4 section in Performance Best Practices for Windchill Pro/ENGINEER Data Management.
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 high bandwidth WANs.

Customizing the Event Console

Pro/ENGINEER Wildfire logs messages to an event console. You can use the *Edit Console Preferences* page to specify which types of messages you want to be logged by the logging system. The *Edit Console Preferences* page is accessed by clicking *Edit Console Preferences* on the Event Console page.

The following message types are supported by the console functionality:

- Debug
- Status
- Info
- Warning
- Error

Use the check boxes to select which message types to log under *Show Message Types in Console*, or additionally to *Show Message Types in Status Line*. (If you select a message to be displayed in the status line, logging for this type of message is automatically enabled.)

Other Recommendations

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 Online Java Performance Guide section in *Performance Best Practices for Windchill Pro/ENGINEER Data Management*. 
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

Pro/ENGINEER Wildfire 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 Pro/ENGINEER Wildfire interaction with the server. If the Windchill server is using secure HTTP (HTTPS), then Pro/ENGINEER Wildfire also uses HTTPS.

Note: General usage of Pro/ENGINEER Wildfire (for example, managing CAD data through check-in or check-out) does not involve any applet, and therefore RMI is not used. However, if Pro/ENGINEER Wildfire is used as a Web browser to access pages containing applets, then RMI should be taken into consideration when configuring the firewall.
Default Directory INI Files

The following tables list key-value pairs for files in the Default directory:

**Cadxhtmlui.ini File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[general] section:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefaultApplication=</td>
<td>PROE</td>
<td></td>
</tr>
<tr>
<td>DefaultDateFormat=</td>
<td>ShortDateFormat1 (recommended), or LongDateFormat</td>
<td>Value of DefaultDateFormat should be one of the formats defined in the [dateformat] section</td>
</tr>
<tr>
<td></td>
<td>ShortDateFormat2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DateOnlyFormat1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DateOnlyFormat2</td>
<td></td>
</tr>
<tr>
<td><strong>[newworkspace] section:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefaultTeam=</td>
<td>&lt;TeamsName&gt;</td>
<td>Team name to be used for objects created in the workspace</td>
</tr>
<tr>
<td>DefaultDocLifecycle=</td>
<td>&lt;docLifeCyclesName&gt;</td>
<td>Default life cycle to be used for CAD documents created in the workspace</td>
</tr>
<tr>
<td>DefaultPartLifecycle=</td>
<td>&lt;partLifeCyclesName&gt;</td>
<td>Default life cycle to be used for WTParts</td>
</tr>
<tr>
<td>DefaultPartView=</td>
<td>&lt;viewsName&gt;</td>
<td>Default view for WTParts created in the workspace</td>
</tr>
<tr>
<td>DefaultDocFolder=</td>
<td>/Administrator</td>
<td>Default folder where documents will be moved to upon checkin</td>
</tr>
<tr>
<td>DefaultPartFolder=</td>
<td>/Administrator</td>
<td>Default folder where parts are moved to upon checkin</td>
</tr>
<tr>
<td>AdditionalValidCharacters_en_US=</td>
<td></td>
<td>Defines valid characters in workspace name. (All alphanumeric characters are valid, so there is no need to define them.)</td>
</tr>
<tr>
<td><strong>[editworkspaceoptions] section:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>effectivityConfigSpecForDocsActive=</td>
<td>False (default) true</td>
<td>Indicates whether effectivity configuration specification is applicable to EPMDocument objects for a specified workspace.</td>
</tr>
<tr>
<td>dateformat</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>LongDateFormat= 3</td>
<td>Date format as yyyy-MM-dd HH:mm:ss z, For example, 2002-05-15 13:25:58 EST.</td>
<td></td>
</tr>
<tr>
<td>ShortDateFormat1= 4</td>
<td>Date format as dd MMM yy HH:mm, For example, 15 Mar 02 13:25.</td>
<td></td>
</tr>
<tr>
<td>ShortDateFormat2= 19</td>
<td>Date format as M/dd/yy h:mm a, For example, 3/15/02 1:25 PM.</td>
<td></td>
</tr>
<tr>
<td>DateOnlyFormat1= 20</td>
<td>Date format as M/dd/yy, For example, 3/15/02.</td>
<td></td>
</tr>
<tr>
<td>DateOnlyFormat2= 22</td>
<td>Date format as M/dd/yyyy, For example, 3/15/2002.</td>
<td></td>
</tr>
</tbody>
</table>
**Console.ini File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[general] section:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>level3icon=</td>
<td>&lt;path to icon&gt; e.g., com/ptc/windchill/cadx /images/redlight.gif</td>
<td>Specifies path to icon displayed for alert level 3.</td>
</tr>
<tr>
<td>level2icon=</td>
<td>&lt;path to icon&gt;</td>
<td>Specifies path to icon displayed for alert level 2.</td>
</tr>
<tr>
<td>level1icon=</td>
<td>&lt;path to icon&gt;</td>
<td>Specifies path to icon displayed for alert level 1.</td>
</tr>
<tr>
<td>level0icon=</td>
<td>&lt;path to icon&gt;</td>
<td>Specifies path to icon displayed for alert level 0.</td>
</tr>
<tr>
<td>[debug] section:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>level=</td>
<td>0</td>
<td>Indicates icon for this type of message.</td>
</tr>
<tr>
<td>console=</td>
<td>yes</td>
<td>Yes displays the information in the console.</td>
</tr>
<tr>
<td>statusbar=</td>
<td>yes</td>
<td>Yes displays the information in the status bar.</td>
</tr>
<tr>
<td>[info] section:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>level=</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>console=</td>
<td>yes</td>
<td>Yes displays the information in the console.</td>
</tr>
<tr>
<td>statusbar=</td>
<td>yes</td>
<td>Yes displays the information in the status bar.</td>
</tr>
<tr>
<td>[status] section:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>level=</td>
<td>1</td>
<td>Indicates icon for this type of message.</td>
</tr>
<tr>
<td>console=</td>
<td>yes</td>
<td>Yes displays the information in the console.</td>
</tr>
<tr>
<td>statusbar=</td>
<td>yes</td>
<td>Yes displays the information in the status bar and automatically enables logging.</td>
</tr>
</tbody>
</table>
### [warning] section:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level=</td>
<td>2</td>
<td>Indicates icon for this type of message.</td>
</tr>
<tr>
<td>console=</td>
<td>yes no</td>
<td>Yes displays the information in the console.</td>
</tr>
<tr>
<td>statusbar=</td>
<td>yes no</td>
<td>Yes displays the information in the status bar and automatically enables logging.</td>
</tr>
</tbody>
</table>

### [error] section:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level=</td>
<td>3</td>
<td>Indicates icon for this type of message.</td>
</tr>
<tr>
<td>console=</td>
<td>yes no</td>
<td>Yes displays the information in the console.</td>
</tr>
<tr>
<td>statusbar=</td>
<td>yes no</td>
<td>Yes displays the information in the status bar and automatically enables logging.</td>
</tr>
</tbody>
</table>
## Newdocument.ini File:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>[general] section:</strong></td>
</tr>
<tr>
<td>okToDownloadAlreadyCheckedOut=</td>
<td>true false</td>
<td>Allows download of an already checked out object.</td>
</tr>
<tr>
<td>DefaultDocFolder=</td>
<td>/&lt;&lt;Name of the folder&gt;&gt;</td>
<td>The destination folder where a newly created CAD document may be moved upon checkin.</td>
</tr>
<tr>
<td>isModelNameUnique=</td>
<td>true</td>
<td>Uniqueness constraint automatically set to true by Pro/ENGINEER.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This value should not be changed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>[proe] section:</strong></td>
</tr>
<tr>
<td>proe.files.component.ext=</td>
<td>.prt</td>
<td>Valid Pro/E model file name extension (no specific extension for &quot;other&quot;)</td>
</tr>
<tr>
<td>proe.files.assembly.ext=</td>
<td>.asm</td>
<td></td>
</tr>
<tr>
<td>proe.files.drawing.ext=</td>
<td>.drw</td>
<td></td>
</tr>
<tr>
<td>proe.files.diagram.ext=</td>
<td>.dgm</td>
<td></td>
</tr>
<tr>
<td>proe.files.format.ext=</td>
<td>.frm</td>
<td></td>
</tr>
<tr>
<td>proe.files.layout.ext=</td>
<td>.lay</td>
<td></td>
</tr>
<tr>
<td>proe.files.manufacturing.ext=</td>
<td>.mfg</td>
<td></td>
</tr>
<tr>
<td>proe.files.markup.ext=</td>
<td>.mrk</td>
<td></td>
</tr>
<tr>
<td>proe.files.report.ext=</td>
<td>.rep</td>
<td></td>
</tr>
<tr>
<td>proe.files.sketch.ext=</td>
<td>.sec</td>
<td></td>
</tr>
<tr>
<td>CADComponentDocNumber=</td>
<td>00008_comp_1.prt</td>
<td>Defines numbering convention for document. [Replace existing values with examples of your chosen convention.]</td>
</tr>
<tr>
<td>CADAssemblyDocNumber=</td>
<td>00008_asm_1.asm</td>
<td></td>
</tr>
<tr>
<td>CADDrawingDocNumber=</td>
<td>00008_drw.drw</td>
<td></td>
</tr>
<tr>
<td>DiagramDocNumber=</td>
<td>00008_dgm.dgm</td>
<td></td>
</tr>
</tbody>
</table>
### Site Directory INI Files

The following tables list key-value pairs for files in the Site directory:

**Autopart.ini File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[general] section:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>autoPartCreator</td>
<td>com.ptc.windchill.cadx.autopart.DefaultAutoPartCreator</td>
<td></td>
</tr>
</tbody>
</table>
### Autoassociate.ini File

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[general] section:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| SearchForPartAttribute=                  | PARTNUMBER          | The value of this preference setting will be an IBA of the CAD Document. The value of the IBA of the CAD Document will give the number of the part to be searched.  
The value of IBA PARTNUMBER will be read programmatically to get the number of part to be searched in database. This preference will be handled by the search delegate class. |
| SearchPartMasterClass=                   | wt.part.WTPartMaster| Enables search for customized parts. This preference will specify the fully qualified class name of the master of the customized part, so that the search is restricted to customized part class and the whole WTPart class will not be searched.  
The value wt.part.WTPartMaster indicates that the search for part will be done on WTPart class. |

### Newdocument.ini File

<table>
<thead>
<tr>
<th>Key</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[general] section:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| isModelNameUnique=                       | true     | Uniqueness constraint automatically set to true by Pro/ENGINEER.  
**Note:** This value should not be changed. |
# Windchill Runtime Environment

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Infrastructure</td>
<td>A-2</td>
</tr>
<tr>
<td>Java Platform Support</td>
<td>A-2</td>
</tr>
<tr>
<td>Three-Tier Architecture</td>
<td>A-3</td>
</tr>
<tr>
<td>Client Software Components</td>
<td>A-4</td>
</tr>
<tr>
<td>Server Software Components</td>
<td>A-6</td>
</tr>
<tr>
<td>Database Components</td>
<td>A-17</td>
</tr>
<tr>
<td>Full Text Retrieval Indexing Components</td>
<td>A-20</td>
</tr>
</tbody>
</table>
Web Infrastructure

Windchill's computing architecture is Web-based. This means that TCP/IP-based intranets and extranets are used to deploy applications built with standard Internet protocols and tools, including HTTP servers and HTML browsers.

Applications designed exclusively for this Web environment can be built and maintained more easily than those supplying Web connectivity on top of older client/server architectures. Web-based applications can leverage the strengths of existing tools and administrator experience to reduce their complexity.

Java Platform Support

Windchill is built using Java. In addition to being a robust programming language, Java provides a complete programming environment and many platform services not normally found in a programming language. Java is a complete programming environment because it provides basic services that allow you to get what you need from Java runtime rather than the operating system. Normally, programs that need to access graphics, network services, the disk, even RAM, use a function call provided as part of the base-level operating system. But, in Java, the built-in runtime, called a Virtual Machine (VM), provides all of these basic services.

Java's support for network programming comes in the form of classes that deal directly with sockets so that connections to servers can be opened. There are also classes to parse network data and to send full Java objects over the wire. In addition, there is Remote Method Invocation (RMI), Java's middleware that allows one object to invoke methods directly on remote objects without any difference in syntax. RMI allows developers to focus on the application, using the objects most appropriate for the task at hand, and separately find the machine architecture tier most appropriate for that object. RMI handles the underlying communication, determines how parameters will be accessed, and provides the serialization of data necessary for the method call so that it can be transported from client to server and back again.

Java also provides GUI building frameworks that contain widgets (i.e. windows, menus, buttons, and so on) for building effective user interfaces. These GUI building frameworks give Java applications a uniform look and feel across platforms, while trying to use the underlying operating-system mechanisms directly.

A series of independent Application Programming Interfaces (APIs), collectively called Java Enterprise, support the building of enterprise applications in Java. Java Enterprise includes facilities to support distributed applications, interfacing to non-Java code, directory services, databases, and more.

One Java Enterprise API is Java Interface Definition Language (IDL). Using Java IDL, Java clients and servers can interact with CORBA-compliant services. With Java IDL, it does not matter what language the CORBA service is written in or is designed to support.
JDBC (frequently referred to as *Java database connectivity*) enables Java clients to interact with databases. You use JDBC to open and close connections, query metadata, issue SQL queries, get result sets, and more. JDBC can use native drives to access any type of data store, but the most common type is relational.

The Java VM implements a security system, called the sandbox model, for running code. As specified by this model, Java code can generally access data only within this secure sandbox. Desktop Integration and other functionality that interact with the user's local file go outside the sandbox, but require user permission.

*Internationalization* is the process of designing and developing an application that can be adapted to the culture and language of a locale other than the one for which it was originally developed. Java facilitates this task by providing classes that convert dates and numbers to formats conforming to local conventions, and providing facilities to load localized resource bundles that contain text visible to users.

### Three-Tier Architecture

The Windchill runtime architecture, illustrated below, is a three-tier application designed and optimized for the deployment of business information applications. The client tier is the presentation layer of the architecture. This tier uses commercial Web browsers executing a combination of HTML, JavaScript, and Java applets to accomplish discrete user tasks.

The next tier, the application server tier, provides the business logic that supports business transactions processing. Commercial HTTP servers, such as Apache or SunONE, and the Windchill method servers provide these functions.

The third tier provides a persistence function. The persistence tier uses an Object Relational Database Management System (ORDBMS) to store structured and unstructured data.
Client Software Components

This section describes the client tier components of the Windchill runtime architecture.

Web Browser

Windchill's primary client component is a Web browser. The widespread availability of low-cost, powerful Web browsers, makes it possible to deploy a large, distributed information system with little or no maintenance of individual client hosts.

The ability to display HTML pages, although adequate for simple applications, does not provide enough functionality for all aspects of complex information authoring applications. Therefore, Windchill requires a browser capable of hosting Java applets based on the Java runtime and base classes. Two popular examples are the Netscape Communicator and the Microsoft Internet Explorer.

Using a Web browser as a front-end, allows leveraging of HTTP server capabilities on the back end. For example, HTTP request authentication, designed for controlling access to other Web server resources, is used to authenticate access to the Windchill system with the need to license and embed security software into Windchill clients and servers. Instead, rapidly evolving authentication schemes can be used in a manner transparent to the Windchill system, giving you more freedom to manage your Web security infrastructure as you see fit.

A Web browser front end also allows you to leverage built-in file download and upload capabilities and the launching of helper applications and plug-ins.

HTML Pages

The initial point of contact between a client and a Windchill server is an HTTP GET or POST request. It is typically a GET request, activated by a link embedded in an HTML page, that initiates connection with the Windchill system.

The Windchill system responds with an HTML page. This page may contain JavaScript or JScript to coordinate window or frame usage within the browser.
Many simple accesses to the system may use only HTML presentation, with HTML form data serving as input. However, the typical client session requires that applet tags (used to carry out complex user interactions involving complex data), be embedded in these HTML responses.

**Java Applets**

Java applets are downloaded from Windchill servers and executed within the address space of the client browser. They provide sophisticated graphical user interface functionality, allowing for complex interactions with the user.

Once running, the applets communicate directly with Windchill servers via Java RMI. This avoids the additional overhead of communicating indirectly through the HTTP server and allows for very complex data to be passed easily between client and server.

If it is necessary to get through firewalls by using an HTTP proxy, Java RMI communication is automatically layered on HTTP. However, this results in greater performance degradation than a direct connection to a Windchill server.

Applet classes loaded from the same Windchill system communicate with one another to use the browser windows and frames, presenting a seamless system image.

Applet classes loaded from federated Windchill systems (coming from separate HTTP servers) cannot communicate directly with one another for security reasons. Intersystem links are therefore accomplished using HTTP URLs, given to the browser for loading into HTML windows. The resulting HTML pages contain JavaScript/JScript and applet tags that use windows and frames to present a seamless system image.

Interactive applets can present feedback on behalf of long-running server transactions. This feedback can take the form of progress indicators and, in some cases, provides the ability to cancel the operation.
Server Software Components

HTTP Server

The HTTP Server is a commercial HTTP server such as Apache or SunONE. The HTTP server is purchased separately, but is expected to be present on each Windchill server host. The Web server will provide HTML pages and Java classes, as well as give access to a Windchill HTTP gateway (described later in this section) as an in-process Java servlet.

User Authentication

The user authentication capabilities of the Web server are leveraged by Windchill to take advantage of the improving authentication standards being built into Web browsers and servers. These include HTTP 1.0 Basic authentication, HTTP 1.1 Message Digest authentication, Digital Certificates, Windows/NT Challenge-Response authentication, and more. Since Windchill is Web-centric, it is important to leverage the server's user authentication rather than become a hole in that security by using an obsolete authentication scheme that is not integrated with the customer's environment. For example, a site using Web servers that support LDAP-based, centralized user and access management (such as SunONE), will be automatically integrated with Windchill for user authentication, rather than maintain a second set of user preferences.

Integration is achieved by configuring a protected instance of the Windchill HTTP gateway. Java applets send a session login request to this URL. The web server does not allow access until the user satisfies the server's user authentication requirements. Normally this involves the server returning an unauthorized response to the client browser that identifies the authentication scheme required. The browser then reacts by resending the request with the appropriate authentication headers, possibly after prompting the user for a password.

Essentially, Windchill is not involved until the Web browser and Web server have securely established the user's identity. Only then does it receive the session login request along with the authenticated user identity.

See the Windchill Application Developer's Guide for more information about authentication and to customize authentication methods.

HTTP Gateway

HTTP gateway is a Java application executed as a servlet. It serves as the initial point of contact between a client browser and Windchill services. The HTTP gateway acts as a conduit to carry the requests and responses between the HTTP server (Web server) and Windchill method servers.

The HTTP gateway connects to a Windchill method server and invokes a special method to handle the HTTP request. The request headers (or CGI properties), set by the Web server, are passed to the Windchill method server along with any submitted data. The invoked method determines what is being requested based on
the submitted data. It delegates to appropriate submethods to generate a HTTP response, usually in the form of an HTML page with appropriate applets embedded within it.

Most requests to the HTTP gateway originate from an HTML browser window, either as a result of an embedded link within a static HTML page that is already being shown, or from a Java applet using the AppletContext.showDocument method to bootstrap a page into the HTML browser window.

This is a fundamental mechanism for linking federated Windchill systems, the Java classes from two systems cannot communicate directly. Showing pages from several Windchill systems in standard Web browser HTML windows allows the client browser to be the center of a star configuration, linking the systems without requiring violation of the strict security restrictions placed on untrusted applets. Requests are forwarded between the systems by encoding appropriate GET requests against their HTTP gateways and delegating to frames within the Web browsers HTML windows to submit these requests.

**HTTP Requests**

The HTTP gateway is accessed through HTTP GET or POST requests. A Windchill URL generally takes the following form:

```
http://<host>:<port>/<gateway path>/<class name>/<method name>?<arguments>
```

The `<class name>` and `<method name>` are used by the method server to dispatch the request to a specific method for processing, and `<arguments>` is a URL-encoded query string. The query string is used to supply additional data that is specific to the method being invoked, such as an object ID. When using a POST request, additional data may also be supplied within the body of the POST request.

This data can range from simple URL-encoded HTML form data to multi-part MIME messages containing the entire contents of one or more files. In either case, the target class is responsible for forming the URL, and, the target method will understand what to expect.

Many target methods will accept both GET and POST requests, and expect the GET request's query string or the POST request's body to contain URL-encoded form data. This is the standard encoding that would result from submitting a simple HTML form to the Web server. It allows using HTML forms as test drivers for these methods, even if the requests are generated in Windchill Java applets rather than from HTML forms.

Basically, URL-encoded form data sends arbitrary name=value pairs separated by a question mark (`?`). All spaces are replaced by plus characters (`+`), and all special characters are hex-escaped into `%dd` format, where `dd` is the hexadecimal ASCII value that represents the original character.
Session Credentials

The HTTP gateway is used when establishing authenticated user credentials. This is done by configuring two identical HTTP gateways: one public and the other protected by Web server user access controls. When a Java client needs to establish valid credentials (to perform secure RMI calls to a Windchill method server), it submits a login request via the protected HTTP gateway. The Web server supplies the authenticated user name and authentication type to the HTTP gateway, and that information is passed on to the Windchill method server.

HTML Page Generation

The HTTP gateway acts as a conduit for delivering requests to Windchill method servers and returning responses through the HTTP server. The content of the responses are controlled by methods implemented within Windchill method servers. These methods may make sophisticated use of JavaScript or JScript in their responses in order to manage HTML browser windows and standalone Java windows from one or more Windchill systems, thereby giving the appearance of a seamlessly integrated environment.

File Upload Using RMI

Files are transferred from the client to the server using a chunked RMI upload. The file is split into manageable pieces and then sent to the server where it is reconstructed and inserted into persistent storage. This capability is only accessible to applet clients and is available as a standard bean within Windchill core. This bean has direct access to the client's file system. It can upload files with the RMI transfer, and it can remove and replace files from the Windchill system.

Be aware that this upload architecture addresses limitations in some browser's Java HTTP classes. The HTTP upload procedure is still available, but PTC does not recommend using it for content upload from an applet. Downloads from the Windchill server via HTTP do not exhibit the same limitations as uploads, and downloads can still use the HTTP architecture described below.

File Transfer Using HTTP

To leverage the Web browser's ability to view, save, and operate on a diverse set of content types, it must be possible to stream file content from the Windchill system to the browser through the HTTP gateway. As shown in the following figure, requests for file transfer are encoded into appropriate
HTTP requests against the server’s HTTP gateway. Requests are then delegated to frames within the Web browser’s HTML windows, where they are submitted and responses are received.

In the Windchill method server, HTTP responses are generated using a streaming interface, allowing the responses to be arbitrarily large. As shown below, this is accomplished by invoking the method to generate the response from within the RMI reply marshaling so the response can be written directly to the RMI result marshaling stream. This allows entire files to be streamed directly from the database without the need to stage them on disk or in memory.
In the following figure, upload streaming is performed in a similar manner, using HTTP POST requests. In this case, the method to read the post is invoked from within the RMI argument marshaling so it can read directly from the RMI argument marshaling stream.

It is possible to develop customized trusted applets that access the client file system directly. They can use similar techniques to stream data to and from Windchill servers. However, the Windchill architecture tries to minimize dependence on techniques like code signing because of the client-side administration required. Therefore, this type of file transfer client applet is generally built as a customization when a site has a client infrastructure that can support code signing.

**Server Manager**

The server manager is a Java application running on each server host. Its primary role is to manage a set of method servers, but it also maintains user session credentials, and manages background processing and other system management functions.

There is a single instance of a server manager on each Windchill server host. It runs in its own Java Virtual Machine (VM) and must be running for the Windchill
system to be considered available. This process could be viewed as a Windchill daemon since it must be running at all times.

Running more than one server VM is not a requirement of the Java architecture. Windchill implements this architecture for reasons of reliability and scalability. Allowing for multiple method servers reduces the risk of a single VM being unable to fully use high-performance multiprocessor hardware when contention for shared resources within a single VM becomes a limiting factor. By allowing multiple processes, the system itself can scale beyond the capacity of the individual VMs to handle high transaction rates.

For example, if a given type II (native method) JDBC driver implementation began to show synchronization bottlenecks at some number of concurrent DB transaction threads, a second method server could double the system's capacity for concurrent transactions.

This architectural feature also addresses reliability because the method servers, unlike the server manager, will execute customized Java code developed by non-Windchill programmers. Although the Java VM provides a very reliable, thread-safe environment, which makes it difficult for errant code to affect other threads, instability can be introduced in the form of memory consumption or resource deadlocks. Further, method servers may use native (non-Java) libraries for database interfaces or other application-specific interfaces. These native libraries can contain bugs that introduce instability into an entire VM. By keeping the Windchill system daemon (server manager) and instances of method servers in separate VMs, individual method servers can terminate without making the Windchill system unavailable or losing user validation information.

Performance concerns are addressed by minimizing the interprocess communication required between the method servers and the server manager, and between clients and the server manager. After clients use the server manager to bind to a method server once, they call that method server directly. If that method server later becomes unavailable (terminates), automatic exception handling transparently rebinds the client to a new one.

**RMI Bootstrap Registry**

Windchill Java clients use Java RMI to communicate with Windchill servers. To use RMI, a client must first obtain a reference to a remote object on which it can invoke methods. The Java RMI runtime initiates this operation by using the concept of a bootstrap registry object, which clients have a built-in ability to construct. This allows them to invoke lookup operations on the registry and receive other, references to remote objects.

To reduce the complexity of the system as well as reduce the number of network connections between clients and servers, Windchill runs its own registry object in the server manager, using a configurable port number. The only object registered in this registry is the local server manager implementation. Other Java RMI applications do not share this registry, and Windchill does not depend on any registry that other Java RMI applications may be using.
Unlike the default RMI registry implementation, the one used internally by the server manager allows client connections to be timed out (discussed later), which improves the scalability of the system in environments with many users. This flexibility is one of the justifications for controlling the bootstrap registry as an internal part of the Windchill system.

RMI-Based Server Locator

The primary purpose of the Windchill server manager is to introduce clients to method servers as needed. The Windchill architecture separates the server manager VM from the method server VM for purposes of reliability and scalability. Clients call the server manager to obtain a reference to a method server and then communicate directly with that server as long as they can. When more than one method server is available, the server manager returns references so as to distribute the load among the available servers.

The protocol for obtaining method server references in the client is encapsulated within the classes that invoke remote methods. It includes fault tolerance for network failures and server manager restarts, and generally will never be accessed directly by Windchill customizers.

Server Management

The server manager is responsible for maintaining the method servers.

Server Launching

The server manager executes method servers as child processes on an as-needed basis. Under high load, it expands the pool of available servers and contracts as usage declines, within some range of management thresholds.

In general, all Windchill method servers are created equal. They are all instances of the same implementation, which dynamically loads Java classes as necessary to carry out requests received from clients. However, to allow for specialty servers that may have unique management requirements, such as limitations due to application-specific native libraries, the server management protocol allows the assignment of unique service names that control the management thresholds and the method server's startup arguments.

Although most generated interfaces invoke the default method service, you can build custom interfaces that request specific service names.

Background Processing

It is often necessary to have a system carry out operations without being directly connected to an end user. This is the case for periodic (time-based) activities, as well as operations that are triggered by a user operation, but for which the user does not wait. For example, an action is performed that promotes an object to a new life cycle state. The change to this life cycle state may trigger additional processing that is not directly related to the user's action. These follow-on activities should be carried out in the background.
The Windchill server manager is responsible for guaranteeing that background processing takes place. The implementation of processing queues and triggering mechanisms actually resides in the Windchill method servers. The server manager is simply responsible for keeping an instance of the method server running so that background processing can take place.

As described in the chapter entitled Administering Runtime Services, your environment can be configured such that there are multiple method servers, one of which is dedicated to running background processing queues.

For more information about queue configuration and maintenance, see Configuring and Administering Background Queues.

Session Credentials and Properties

Windchill leverages the user authentication capability of the HTTP server. However, the vast majority of client requests do not come through the HTTP server, but instead come directly from the client through Java RMI. This requires a place to cache the HTTP authenticated user names so they can be securely associated with subsequent RMI calls. Because the server manager represents a daemon process that outlives individual method server processes, that place is within the server manager VM.

As discussed previously, when clients need valid credentials, Windchill is uninvolved until after the HTTP server allows access to a protected Windchill HTTP gateway. The gateway then passes the authenticated HTTP request to a method server for processing. The method server processes the request for credentials by storing the authenticated user name and associated session properties (passed on the request) with a session manager that runs in the server manager VM.

Live connections are not used to maintain the session database within the server manager. To reduce resource consumption, credentials are validated by the method server, even though the client is disconnected from the server manager. Rather than live connections, a limited size, most-recently-used caching algorithm is used. In the event a client is still alive after its session credentials have been aged out, automatic exception handling transparently reestablishes the credentials.

Client Time-Out and Connection Limits

Scalability demands that individual clients do not consume significant server resources indefinitely. A large number of infrequent users should not require that the system is hosted on super-server hardware. Server host sizing should be a function of transaction throughput, not of user count.

The Java I/O model, in particular the Java RMI implementation, dedicates at least one thread to each network connection. To make this scalable to large number of users, Windchill implements two mechanisms to free network connections and threads. The first is to time out connections that remain idle for a specified period of time. The second is to limit the total number of client sockets the RMI runtime is allowed to consume. This limit is enforced by closing the least-recently-used
connect. Thus, new client connections are not refused, and connection timeout is faster when under a heavy client load. Clients recover from the disconnection automatically.

System Management

Being the daemon process of the Windchill architecture, the server manager becomes the key process for performing Windchill system management functions, such as starting and stopping method servers.

The System Configurator provides an interface for these functions, although some actions (like shutting down the servers) are restricted to authorized user names.

Method Server

This component is a Java application that executes all methods representing business object transactions. Architecturally, it starts out simply as a skeleton process that dynamically loads specific Java classes as they are needed to service client requests. The following figure shows the anatomy of a method server.

RMI-Based Method Invoking Interface

When a method server process is started, it creates an instance of a method server object, which is exported as a remote object to the server manager. Clients bind to a method server by retrieving this object reference from the server manager, and interacting with the method server directly.

The binding and method-invoking machinery is hidden from application developers by utility classes and generated helper classes. Its architecturally significance is that it helps explain how the Windchill runtime operates.

A significant advantage of using Java RMI to invoke server methods is the built-in support for transferring arbitrarily complex object graphs between client and server. This allows transactions to use sophisticated arguments and results without complex programming of the client-to-server interface.

Access to server-side methods is exposed to clients by using helper classes corresponding to each business class. These classes wrap the externally available
server-side methods of their business class with implementations that forward the calls to a method server where the real method is invoked. The modeling of the interfaces and the generation of helper classes is discussed in detail in the *Windchill Application Developer's Guide*.

**Database Access**

The method server is the only Windchill process that communicates directly with the database. In this sense, Windchill runtime is a classic three-tier architecture. Using a shared database login, the method server maintains multiple database connections assigned to worker threads as needed to carry out individual transactions.

The interface to the database is implemented by a Persistent Object Manager (POM) layer within the server that acts to abstract the actual database interface from the business logic. Persistence is described in detail in the *Windchill Application Developer's Guide*.

**Client Time-Out and Connection Limits**

As with the server manager, scalability demands that individual clients do not consume significant method server resources indefinitely. Therefore, Windchill method servers implement the same mechanisms as the server manager to time out idle connections and limit the number of client sockets the RMI runtime is allowed to consume.

**Client Feedback**

Although some of today's distributed object technologies, including Java RMI, allow servers to call back to client objects with feedback, there are problems with this obvious approach to client feedback.

First, it forces a logical decoupling of the feedback from the operation, because the client must create objects to receive feedback calls. These objects must maintain state about the operation, or pass enough information on calls to reassociate feedback to the operations at a later time. In either case, this additional overhead is wasted if the server does not produce any feedback. An analogy may be the unwieldy exception processing that would result if the exception were decoupled from the operation throwing it. It can be argued that there is a logical similarity between operation feedback and exception handling.

Second, passing remote object references incurs overhead that is wasted if the server does not perform a callback. If one tries to eliminate this by caching the references up front (that is, send once, reuse later), robustness suffers because the communication transport on which the original object was exported may be disconnected by the time it is used. Java applets cannot accept incoming connections, so a stale client reference cannot be reconnected. Attempting to call back on a timed-out connection simply throws an exception in the server.

Finally, because applets cannot accept incoming connections, Java RMI tunneled through a HTTP proxy will not allow the server to call back because
communication transport used for the call (HTTP) is not sufficient to handle a call in the reverse direction.

The Windchill architecture addresses these concerns by implementing a lightweight feedback mechanism into the remote method-invoking protocol. This is done by allowing feedback objects to be sent from the server to the client as part of the RMI reply marshaling stream. They are received and processed within the thread performing the call, and they share the same communication connection as the call, thus remaining logically coupled to the call itself.

When processing a method invocation from a client, the server-side method is invoked from within the RMI reply marshaling code, allowing the server-side method to flush feedback objects onto the reply stream at will. The client reply unmarshaling code recognizes these objects as feedback and calls their init methods, then continues to wait for the real reply. When starting a long operation, the server methods can send a GUI component such as a progress bar and cancel button. The server can periodically flush additional feedback objects that update this component. The cancel button is programmed to invoke an operation canceling method in a second thread capable of interrupting the first thread in the method server.

User Authorization

To authorize access to a given object or operation, the method server must be able to reliably identify the user performing the action. Various aspects of user authentication (securely establishing session credentials) have already been discussed. These things come together in the method server to allow a method to inquire about the user associated with the current execution thread. This capability allows applications to implement access control policies, which are described in detail in the Administering Access Control chapter.

Java RMI does not provide an inherent means of reliably identifying the calling user. However, the Windchill runtime architecture satisfies this need within the method server's remote method-invoking interface. Client credentials are implicitly included with RMI method arguments, and digital signatures are used to securely associate the RMI thread with an authenticated user name. This association is established before the target method is called, so method signatures do not need to contain an extra context or user argument. The information is retrieved if and when it is needed.

Additionally, the association can be dynamically modified in the course of executing an operation. For example, it may be necessary to carry out certain steps of a transaction as a principal other than the user initiating the transaction. To implement arbitrary authorization delegation schemes, methods are allowed to push and pop the principal currently associated with the execution thread.

Background Processing

Windchill provides for background processing through the use of background method queues stored in the database. The queues are tables of method invocation
specifications that are executed by a background processing manager. The specifications are essentially method names and serialized arguments (stored as BLOBs) that are stored in the database for reliability.

A transaction that triggers background processing includes updating a background method queue as part of the overall transaction. Once committed, the background manager is notified, and it proceeds to execute the methods asynchronously. Removal of queue entries is performed within the transaction that carries out the method, thus guaranteeing that entries are processed to completion only once, while still ensuring that incomplete transactions are restarted after system failures. Upon failure, entries are marked as requiring administrator intervention and ignored.

Examples of the background processing mechanism include life cycle processing, workflow automation, and full text retrieval (FTR) index maintenance.

For information about background queue configuration and maintenance, see Configuring and Administering Background Queues.

Log Files

Log files are used to capture exception/error tracebacks and debug tracing messages. In the first case, log entries are generally infrequent, marking exceptional events. However, you can enable more verbose logging levels for troubleshooting purposes. (Full tracebacks may not be available when you run some JIT compiler implementations.)

Many packages support the printing of messages during execution to assist you in debugging. This option is typically controlled by property settings in the wt.properties file. Using these properties, you can enable or disable writing to log files. Additionally, log files can be appended or overwritten at each execution. Output can be sent to both the console and log file, or just the log file.

Logging does have a performance impact, so the verbose mode should be turned off if you are not debugging.

Each server application (server manager, method server, and HTTP gateway) has a separate log. For the HTTP gateway, CGI and Servlet share the same log file. In addition, code generation tools also have log files.

Database Components

Object Relational Database Management System (ORDBMS)

The Windchill system uses an ORDBMS to store structured and unstructured business data. The database manager is typically run on the same host as the Windchill servers, but at larger sites it may run on a dedicated host and be accessed remotely from one or more Windchill server hosts. Oracle, with support for very large objects and object tables, is the reference feature set for which Windchill is designed.
The use of an ORDBMS is leading-edge, but Windchill does not push the technology past reasonable bounds of usability and safety. Windchill leverages support for very large objects and object references (bigger BLOBs and object-ID navigation capability). It does not rely on the more futuristic capabilities of complex data types where, through extensions (object types, cartridges, and so on), the DBMS tries to understand the structure and meaning of Java objects.

**Caution:** Windchill uses the object relational features of the Oracle database server to store data objects. In order to maintain the integrity of the associations among stored objects, users and administrators should avoid using tools such as SQL*PLUS to directly manipulate database data. Directly changing data in the database could compromise data integrity. This does not preclude the use of such tools for standard database administration, which does not alter or change the values stored in the tables.

**Single Logical Database**

A single Windchill system uses one logical database. The database administrator may use vendor tools to physically partition the database but, for simplicity, Windchill will not try to coordinate transactions between multiple databases in real-time. It is assumed that the reasons which justify having separate databases also justify deploying Windchill as two or more federated systems, or using DBMS store/forward replication technology.

**Storing File Content as Large Objects**

Information managed by Windchill exists either as pure structured business information (attributes of objects and relationships) or as unstructured information created by applications in the form of external files utilizing either standard or
proprietary data formats. The following figure illustrates ORDBMS management of structured and unstructured attributes.

Structured data is stored using normal relational techniques (tables), and unstructured data is stored as objects. Storing file content in the database obviously results in large databases and would cause performance problems with traditional RDBMS technology. However, new ORDBMS technologies (Oracle, for example) are designed to enable this approach.

The runtime architecture for persistence is based on the CORBA model, as shown in the following figure. Every object that implements the persistable interface is assigned a persistence identifier. The PersistenceManager interface identifies the set of methods that applications use to manage the persistent state of their business.
objects. While all of the methods declared by this interface execute on the server, they are accessible to client applications through a helper class.

The Persistent Object Manager brokers persistence requests and forwards them to a PersistentDataService to handle the actual persistence operation. The protocol used to pass objects back and forth to the Oracle Persistent Data Service is a combination of introspection and JDBC calls to stored procedures. Introspection is used to bind the attributes to stored procedure variables.

**Full Text Retrieval Indexing Components**

This section provides a conceptual overview of Windchill's indexing capabilities. For information about the maintenance of collections with Convera RetrievalWare, see [Administering RetrievalWare Libraries](#). For detailed information about creating an indexing policy, see the *Windchill Business Administrator’s Guide*.

**File Content Indexing**

Each file that is managed by Windchill has metadata attributes that are useful in identifying the file, including its purpose, MIME type, description, time of creation, and so forth. (Attributes can be added as desired). Files may also be related to domain-specific objects, further identifying their purpose and use, and providing a means to locate the appropriate file and version.

Still, in some cases, metadata attributes and relationships may prove insufficient to find a desired file, particularly in situations where it is not known if a file exists. In such a case, the ability to search for key words and phrases that are contained within the body of the document (file) represent the best mechanism for locating documents. This capability is used by Web search engines such as InfoSeek, AltaVista, and Lycos.
Windchill uses Convera RetrievalWare technology to index file content for selected MIME file types at the time the file is added to the system. Windchill method servers stream the file content and certain metadata fields to the RetrievalWare interfaces in the background after the information is initially placed into or updated in the database.

**Publishing**

The initial focus of the Windchill indexing architecture is to publish information to full text retrieval (FTR) indexes, creating index entries that correspond to managed business objects. Later, Windchill systems may make use of FTR indexes to perform internal searches or processing.

The Windchill strategy is to allow multiple Windchill systems to push information to shared RetrievalWare indexes and leverage RetrievalWare tools to build Web search interfaces to the indexes. The Web search interface becomes a powerful integration point between separate Windchill systems, allowing users to locate objects independently from their own systems and navigate back into those systems to access the objects.

**Indexable Objects**

Windchill provides a general-purpose architecture that allows for any business object to be indexed in one or more RetrievalWare indexes. *Indexable objects* are those objects for which index entries can be constructed. The decision to make a class indexable is done at modeling time and implies that a meaningful URL can be constructed to link the search results back to some behavior the object provides.

The decision about which classes of objects should be indexed and what information should be included in an index entry is best made in conjunction with the designed use of each index and its associated Web search interface. Therefore, the Windchill indexing architecture separates the indexing behavior from the business object classes. These decisions are delegated to indexing policies and indexer objects.

**Indexing Policies**

*Indexing policies* determine what objects require indexing. They provide administrative control over indexing by associating indexable objects to indexer objects. Indexing policies are similar in concept to access control policies.

Essentially, indexing policies are an association between indexable objects and a set of indexer objects based on the indexable object's class, administrative domain, and life cycle state. Changes to an object's state may make it eligible for indexing according to an existing indexing policy. This may cause it to be indexed for the first time or cause the set of indexes containing its entry to change.
When an object is subject to indexing, index entries are maintained in the background whenever the object or one of the objects contributing to its index entry is changed.

To view, update, or create indexing policies, use the Policy Administrator. For details on how to use the Policy Administrator, see the *Windchill Business Administrator’s Guide*.

**Indexer Objects**

For each Windchill system publishing to a RetrievalWare index, there is an associated instance of an indexer object. The indexer object acts as an adapter between the indexable objects within the Windchill system and the given RetrievalWare index. The indexer classes implement the way in which objects are indexed. This behavior can include translating metadata to common attribute names and values, and collecting attributes from related or contained objects to be included in an indexable object's index entry.

Indexer classes are implemented as standard Windchill business classes to make them easily customizable and extendable. A reference implementation is provided that knows how to map simple attributes to a general-purpose index format. The reference implementation can be augmented or subclassed to tailor this behavior for the needs of particular kinds of RetrievalWare indexes. Simple customizations will typically include navigating associations between several objects in order to build more meaningful and complete index entries. For example, the index entry for a container object may include information from the objects contained within it in addition to the attributes of the container itself.

Indexer objects perform their work in the background and execute as a predefined user. The user identity is configured when an indexer object is created and is used to enforce access control policies for the objects being indexed.

**Index Loader**

The index loader is responsible for feeding information into RetrievalWare for indexing. It is invoked by submitting index data to an HTTP Servlet or CGI that runs on the host where the RetrievalWare is located. The index loader invokes a RetrievalWare API to initiate the indexing of metadata and content file information.
## Windchill Considerations for Security Infrastructures

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>B-2</td>
</tr>
<tr>
<td>Protocols</td>
<td>B-2</td>
</tr>
<tr>
<td>Authentication</td>
<td>B-3</td>
</tr>
<tr>
<td>URL Generation</td>
<td>B-4</td>
</tr>
<tr>
<td>RMI</td>
<td>B-8</td>
</tr>
<tr>
<td>Configuration Considerations</td>
<td>B-11</td>
</tr>
</tbody>
</table>
Overview

As a Web-based application, Windchill must be compatible with security infrastructures of intranets, extranets, and the Internet.

This appendix provides some basic information for dealing with firewalls, proxy servers, reverse proxy servers, Network Address Translation (NAT), and so on.

Note: This information is provided only to assist you with security infrastructure management. PTC does not provide support for any third-party products mentioned here, nor is PTC responsible for your security infrastructures.

Protocols

To understand how network security infrastructures affect Windchill, you need to understand the communication protocols within a Windchill system. To understand the affect of network security products on this connectivity, you should understand how clients connect to servers. See the following table:

<table>
<thead>
<tr>
<th>Client</th>
<th>Example</th>
<th>Communicate Protocols</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser with pure HTML user interface</td>
<td>Local search page; properties page</td>
<td>HTTP or HTTPS</td>
<td></td>
</tr>
<tr>
<td>Java clients, in the form of Java applets in HTML pages</td>
<td>Windchill Explorer; Product Information Explorer</td>
<td>Java RMI and HTTP or HTTPS</td>
<td>Java applet or application HTTP requests are performed via the java.net.URLConnection class.</td>
</tr>
<tr>
<td>Stand-alone Java applications</td>
<td>Workgroup Manager for Pro/E; Workgroup Manager for CADDS</td>
<td>Java RMI and HTTP or HTTPS</td>
<td>Java RMI attempts to establish direct socket connections from client to server (never the reverse) on well-known server port numbers (configurable). But it may also fail to be tunneled over HTTP or HTTPS.</td>
</tr>
</tbody>
</table>
Windchill servers use other protocols between various server components within a single system. These systems are local to the server host(s) or behind the firewall(s), where they do not cause additional configuration concerns. These are some examples:

- Servers connecting to directory services using LDAP.
- Windchill servers connecting to Oracle database servers using Net8.
- Info*Engine servers connecting to application adapters.
- Web server plug-ins connecting to Java servlet engines or Web application servers.
- Windchill servers in a cluster connecting to one another using Java RMI.

**Note:** HTTP is used when federated systems communicate (for example, in a federated search, proxy refresh, or content replication). Windchill uses Java RMI only for internal communication between Java classes belonging to a single system (that is, classes from the same codebase).

### Authentication

Windchill relies on a site's existing HTTP authentication infrastructure to provide user authentication. Typically, this is a Web server, which authenticates HTTP requests using an LDAP-accessible directory service as its user database. Access to Windchill-served resources is then restricted to authenticated users. This authentication often uses HTTP basic authentication. However, because it is a function of the Web server and browser, additional authentication schemes and third-party security products can be used transparently in Windchill. Windchill does not rely on HTTP session state (such as cookies) for authentication. It does not preclude the use of Web application servers that use cookies in their proprietary authentication schemes, but its use would be transparent to Windchill. In Windchill, each HTTP request is authenticated by the HTTP server before reaching Windchill code. Windchill requires that the hosting Web server and servlet engine provide the authenticated user name with each HTTP request. It does not matter how the user name determined.

Windchill keeps track of the resources that are used for authentication in the following file:

```
<Windchill>/apacheConf/config/authResAdditions.xml
```

where `<Windchill>` is the directory where your solution is installed. Any resource that requires user identification to generate a unique dynamic response for the given user are included in this file.
Although each authenticated HTTP request is individually authenticated by the Web server or Java servlet/JSP server, Java RMI communication uses direct connections between Java clients and Windchill RMI servers. This direct communication leverages HTTP authentication in the following manner:

- It establishes session state on behalf of the RMI client within the Windchill servers.
- It uses an authenticated HTTP request to identify the session's user.

Subsequent RMI calls from the client to the server contain information that maps the call to an existing authenticated session. This RMI session authentication happens automatically on an as-needed basis. When an attempt is made to invoke services that require user identification, this is handled transparently to the calling code, unless the calling client is a multi-user server application itself. In that case, the calling code should explicitly manage thread-based context when calling Windchill APIs. (For more information, see JavaDoc for wt.util.WTContext and wt.httpgw.WTContextBean.)

## URL Generation

HTTP URLs can be references to static resources or dynamically generated responses.

*Static resources* are files contained in the Windchill codebase, which are usually served directly by the Web server from a virtual directory alias.

*Dynamically generated resources* are responses generated by Windchill server code and are usually served by a servlet engine executing a Windchill servlet.

The dynamic content is further divided by the servlet responsible for generating the response.

Multiple servlets exist primarily so different access restrictions can be placed on them by the Web server. For example, there are different gateway URLs for anonymous access, authenticated access, and system administrator access. This makes it possible for the Web server to be configured differently for each of these servlets.

To accommodate different access restriction capabilities of Web servers and servlet engines, each servlet URL may require separate access restriction. This means they do not all need to appear underneath a single Web application root URL. Each servlet is configured by a different Windchill property, as shown below:

- wt.httpgw.url.anonymous property
- wt.httpgw.url.authenticated property
- wt.sysadm.url property
Server Codebase Property

The server codebase property, wt.server.codebase, specifies the URL to the Windchill codebase virtual directory used by Windchill servers when producing URLs to static files. Most often, the server codebase property is used in a <BASE> tag within dynamic HTML pages. This allows relative HREFs to be used within the page for static resources, such as style sheets and images. It is also used by client-side Java code to access files from the server's codebase, such as wt.properties or JAR files.

All files in the Windchill codebase virtual directory can be available anonymously, except JSPs. This is because the dynamic nature of the JSPs typically requires that most pages are unique to a particular user.

When standalone Java applications are run outside of a browser, some files in the server codebase must be available anonymously because the HTTP protocol handler in the standard Java Runtime Environment, does not support authentication challenges. These files include wt.properties and JAR files.

Using HTTPS Protocol

HTTPS is the HTTP protocol layered over the Secure Socket Layer (SSL) protocol to allow secure data transfer using encrypted data streams. This section describes the Windchill configuration necessary. It assumes the web server has been set up for HTTPS. See your web server documentation for details on this procedure. RMI is not encrypted, but may be tunneled over HTTPS. See RMI-over-HTTP for more information.

The Java 2 platform does not include default support for HTTPS until version 1.4. The Java Secure Socket Extension (JSSE) enables secure communication, including HTTPS. To install this extension into a pre-1.4 Java runtime:

1. Install the JSSE JAR files. Copy jsse.jar, jnet.jar, and jcert.jar from <windchill>/lib to <java>/lib/ext
2. Add the JSSE provider to your list of approved security providers. Edit <java>/lib/security/java.security to contain:

```
security.provider.n=com.sun.net.ssl.internal.ssl.Provider where n is the next provider preference available.
```

Add an HTTPS protocol handler to the runtime. Windchill servers have this set by default, so no action is needed there. Any java application, including the servlet engine process, that creates URL objects with an HTTPS protocol will need to set a system property on the command line.

Windchill includes a HTTPS protocol handler via the HTTPClient package. To use, specify ":-Djava.protocol.handler.pkgs=HTTPClient".

Alternatively, the JSSE includes a handler. To use it, specify:

":-Djava.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol"
When using the &lt;Windchill&gt;/bin/windchill java class execution wrapper, you can add the java arguments that it will use in the wt.java.args property in wt.properties. To enable HTTPS using this method, set the following in wt.properties using the xconfmanager utility:

```
wt.java.args=-Djava.protocol.handler.pkgs=HTTPClient
```

For further information, see [http://java.sun.com/products/JSSE](http://java.sun.com/products/JSSE) and the INSTALL.txt file located in the JSSE download.

Prior to Java 1.4, the Java plug-in provides support for HTTPS through the browser. One shortcoming of this approach is that the JAR files cached for applet execution require a version number to indicate when updates are available. Windchill applets include this JAR cache version when wt.properties contains the following property:

```
wttaglib.util.plugin.useCacheVersion=true
```

The JAR versions are stored in a jar.properties file residing in the same directory as the requested JAR file. To ensure that the JAR versions are compatible with Windchill, the following variable in the wt.properties file must be set:

```
wt.tools.boot.updateVersion=1.3
```

To rebuild client JAR files and increment the JAR versions, complete the following steps:

1. Ensure that the wt.taglib.util.plugin.useCacheVersion and wt.tools.boot.updateVersion properties are set as described earlier in this section. You can use the xconfmanager utility to display current values and set values for these properties.

2. Enter the following command from a windchill shell:

```
  ant -f MakeJar.xml
```

**Tip:** You can also use this ant command to force a version update so that clients download all JAR files.

You must restart the servlet engine and method server for the applet tag generation utilities to pick up the updated version information.

### Relative and Absolute URLs

The notion of relative hyperlinks (HREFs) exists only within the context of HTML pages. In Windchill, relative HREFs are used within static HTML pages and the static portions of HTML template files. Absolute HREFs are used for all dynamically generated HREFs.

A typical dynamically generated Windchill HTML page includes the following:

- A &lt;BASE&gt; tag, specifying an absolute URL to the static Windchill codebase as configured by the wt.server.codebase property.
• Relative HREFs to static resources.
• Absolute HREFs to other dynamically generated pages.

Most Windchill HTML pages are generated from HTML template files. Templates are allowed to contain HREFs to other static resources (such as images, backgrounds, and style sheets), without requiring the links to be generated by script calls if the document base is specified as the root of the Windchill virtual directory. To make sure the template’s contents are not tightly coupled with the request URL, the <BASE> tag is dynamically generated using a script code. This allows a response template to be shared by many requests that may have a variable number of PATH_INFO elements. Links to other dynamically generated pages (via servlets) are also generated by script calls and product-absolute HREFs.

Most dynamically generated HREFs share some URL components (for example, protocol, host, port, and path prefix) with the base URI of the pages containing them. It should be possible for Windchill to generate relative HREFs into the pages. However, most Windchill code currently uses java.net.URL objects internally when generating HREFs, and there is no such thing as a relative java.net.URL object. Thus, it is currently not possible to configure Windchill to generate all HREFs as relative links. If it were possible, it would still not be advisable to access a Windchill system using more than one base URL, such as using one URL for internal users and another for external users accessing through a reuse proxy. Although this might not result in changes to the internal system's configuration, host names and URLs are not used only in HTTP requests and responses. Host names also appear in RMI stubs, and URLs also appear in HTML e-mail.

Enterprise deployments, reverse proxy configuration, in particular, should use single, application-specific host name aliases to enable controlling network connectivity through name resolution, as described in the next section.

Choosing Host Names

A Windchill system is an enterprise resource, much like a mail server, directory, or corporate intranet Web server. As such, it is good practice to give the system its own host name alias. This allows the system to move to different hosts or even to different networks, without affecting user bookmarks or e-mail links that already exist. For example, suppose the ACME company has used Windchill to implement an engineering change management system, with the code name ECMS. A DNS alias of ecms.acme.com should be set up, rather than using a specific host name, such as server12.east.acme.com.

By configuring all Windchill host name and URL properties to use the desired alias, the IT department can control how this name is resolved to an IP address, both internally and externally.

• From within the corporate intranet, the name can resolve directly to an internal server on a private internal network.
• From outside the corporate intranet, such as from a partner extranet or the Internet, this name can resolve to a reverse proxy on an external service network.

By using a DNS alias, access to the system remains location independent. The physical location of the user does not affect bookmarks, e-mail, or saved HTML pages. This is important for mobile users.

**RMI**

Many existing Windchill applets and applications use Java RMI to invoke server transactions. There is a continuing shift of focus from this form of communication towards HTTP and XML. But for now, the Windchill development environment continues to support code generation of classes that use RMI to invoke remote service methods.

RMI is a Java-centric remote procedure call (RPC) mechanism implemented on sockets. RMI stub objects perform a remote method invocation between an RMI client and an RMI server. These stub objects contain a host name and port number to which a TCP/IP connection is opened by the client. Windchill exposes only two RMI objects to clients: a server manager object and a method server object. Other RMI objects are used server-to-server to coordinate cached information, but these are not important for client connectivity.

**Server Hostname Property**

Each RMI stub contains a server host name. The value serialized into stub objects is controlled by the java.rmi.server.hostname property of the RMI server. Although this is a Java system property, it can be set in the Windchill wt.properties file, because values in that file are used as Java system properties by the Windchill servers.

Use the xconfmanager utility to set the java.rmi.server.hostname property to a symbolic name that all clients are able to resolve to a server address. Because Java applets can connect only to their codebase hosts, it should be the same symbolic name used in the wt.server.codebase property, which is used as the document base for Windchill HTML pages.

If a Windchill server host name alias is used, and it does not resolve to the local server (such as an alias for an IP load balancing server cluster), the name must be forced to resolve locally to the loopback address, 127.0.0.1. This is because the RMI stubs can contain only one host name, which will be used by all clients, both local and remote. However, to remain local, some local communication between the server manager and method servers must be guaranteed. If you give the system its own host name alias, as recommended above (rather than using actual host names), then you can safely override the local name resolution (in the /etc/hosts file) for this alias.
Configuration Properties

By default, the RMI system chooses random available port numbers for RMI servers. However, this makes it impossible to configure firewalls to allow direct RMI connectivity. Port numbers accepting incoming connections are controlled by configuration properties.

Windchill clients first connect to a server manager, which acts as a broker for service implementations. A Windchill system has only one server manager per server host, and its port number is controlled by the wt.manager.port property in wt.properties. Each server host may have multiple method servers running, so their port numbers are configured as a range controlled by the wt.method.minPort and wt.method.maxPort properties. The following are the default ports:

- wt.manager.port=5001
- wt.method.minPort=5002
- wt.method.maxPort=5010

To change these defaults, use the xconfmanager utility to set the properties to different values.

RMI-over-HTTP

If a direct TCP/IP socket connection cannot be established between the client and an RMI server's host and port, RMI calls can be transported over the HTTP protocol. Although the Java RMI specification is clear about this tunneling, the default Java implementation depends on some Java system property settings. Therefore, RMI does not automatically fail over.

RMI Servers within Windchill overcome this limitation by allowing the socket factory, which is used for RMI communication between client and server, to be configurable. Socket factories supplied by the Windchill bootstrap package (boot.jar), that support RMI-over-HTTP(S), may be used. The following properties control the socket factories exported by Windchill RMI Servers (the default values are null, which result in using the default Java socket factories):

- wt.rmi.clientSocketFactory=wt.boot.WTRMIMasterSocketFactory
- wt.rmi.serverSocketFactory=wt.util.WrappedRMISocketFactory

Note: RMI-over-HTTP tunneling is enabled only when the client has installed the bootstrap package (boot.jar). Otherwise, only direct RMI socket connections to the RMI server host and port are supported.

Windchill includes socket factory WTRMIMasterSocketFactory, which improves on the J2SE default connection failover logic, which:

- Supports tunneling of RMI calls over HTTP and HTTPS regardless of system properties.
• Supports configurable URL paths for Java RMI CGI compatible proxy script
• Uses asynchronous connection attempts for all socket factories to reduce total connection time on the initial connection.

The WTRMIMasterSocketFactory uses a series of secondary socket factories to connect to the RMI server. The first connection that is successful, is used:

1. If WTRMIMasterSocketFactory does not have sufficient privilege to perform any of its required operations, the installed default socket factory is used instead.

2. The wt.boot.socketFactory system property is read on the client, and if the fully qualified class contained within the setting can create an instance of a RMISocketFactory, that socket factory is used for client to server communication.

3. If the client configurable socket factory fails, the socket factory starts a direct socket connection (wt.boot.WrappedRMIDirectSocketFactory).

4. If the direct connection fails, or does not complete within the failover time-out (as defined by wt.boot.failoverTimeout), the socket factory starts an HTTP connection to the RMI target port in case the client is behind an HTTP proxy server (wt.boot.WrappedRMIHttpToPortSocketFactory).

5. If the HTTP to port connection fails or does not complete within the failover time-out, it starts an HTTP connection to the Java RMI CGI gateway found on the server which supplied the client codebase (wt.boot.WTRMIHttpToCodebaseSocketFactory).

6. If the HTTP to codebase connection fails or does not complete within the failover time-out, it simultaneously starts HTTP and (optionally) HTTPS connections to the default Java RMI CGI proxy (wt.boot.WrappedRMIHttpToCGISocketFactory and wt.boot.WTRMIHttpsToCGISocketFactory).

The first connection type that completes successfully is used, and the resulting socket factory is reused for all subsequent connections to that host.

Port 80

When tunneling RMI over HTTP, the Java RMI specification supports only port 80 (default) for HTTP and a fixed URL path of /cgi-bin/java-rmi.cgi. This is because in Java 1, a single RMI socket factory is shared by all RMI client stubs. With a shared socket factory, there is no support for each RMI server to specify its own unique connection requirements.

Because this limits the desired configuration options, support has been added allowing the server host, protocol, and port number, to be derived from the codebase URL where the calling Java code is downloaded.
If a firewall does not reject connections, then this failover behavior is defeated. In that case, the client must be configured to use the specific secondary socket factory that is required. This can be done by setting the client’s system property, wt.boot.socketFactory. The secondary socket factories, used within the master socket factory, are listed after their explanation in the failover logic list above.

**Java RMI Servlet**

When directly connecting to RMI servers is not allowed, the Web server must respond to requests for /cgi-bin/java-rmi.cgi, to make the forwarding of HTTP requests possible. An actual CGI file is provided in the Java JDK. The Java RMI specification expects this file to be added to the Web server's cgi-bin directory. By setting wt.rmi.javarmicgi to another URI, the CGI file can exist anywhere, for example /servlet/JavaRMIServlet.

To improve performance, security, and flexibility, Windchill delivers a servlet that can be mapped to the same URL. The servlet class is wt.tools.javarmi.JavaRMIServlet. This class adds security as it can be configured through servlet initialization parameters that forward connections to a predefined range of destination port numbers. However, the java-rmi.cgi file provided in the JDK allows the HTTP request to identify any port number on the local host, opening other services to potential attack.

To improve performance, wt.tools.javarmi.JavaRMIServlet does not start a new process for each RMI call. To add flexibility, it allows itself to be configured to forward requests to a nonlocal RMI server host, thereby acting as an RMI proxy server. The servlet parameters are:

- serverHost
- minPort
- maxPort

**Note:** By default, the Java RMI servlet is disabled. However, its configuration elements are included as comments in the web.xml file for the Windchill Web application.

**Configuration Considerations**

**Firewalls**

You must be connected to the Windchill web server to allow the HTTP or HTTPS port number through. Defaults are 80 and 443 respectively. If RMI clients are used outside the firewall, then direct connectivity can be supported by allowing the following two ports through the firewall:

- wt.manager.port
- wt.method.minPort through wt.method.maxPort.
Direct connections to the application port numbers are as secure as forcing RMI communication to be tunneled over HTTP requests. However, you can disallow direct connections to the RMI server ports to force all RMI communication to be tunneled over HTTPS for data privacy or to leverage an HTTP reverse proxy.

The host name used in URLs and RMI stubs, which is controlled by the java.rmi.server.hostname property, must resolve to an IP address for clients inside and outside the firewall. If the firewall is performing network address translation, or is configured to proxy Windchill connections, the host names presented by Windchill to its clients must be valid for them to connect to the servers. The host names presented by Windchill are controlled by the various hostname and URL properties previously described.

**Client-Side Proxy Servers**

All Windchill HTTP traffic is compatible with indirect access through an HTTP proxy. However, tunneling RMI requests over HTTP through the HTTP proxy requires the use of Windchill's bootstrap package to enable the necessary RMI socket factory logic.

Windchill applets using RMI from within a browser automatically take advantage of the browser's HTTP proxy settings when opening URL connections. However, stand-alone applications require that http.proxyHost and http.proxyPort Java system properties be set. This may be done by altering the command line of the application to include `-Dhttp.proxyHost=proxy.acme.com -Dhttp.proxyPort=8080`.

**Server-Side Reverse Proxy Servers**

Typical use of a reverse proxy server requires all incoming traffic to be HTTPS. This requires that the wt.rmi.clientSocket factory is configured to tell the client to send RMI requests through the alternate socket factories and that the Windchill server is configured with the java-rmi.cgi file or wt.tools.javarmi.JavaRMIServlet.

The Windchill host name should resolve to the reverse proxy server for clients that are required to access through the reverse proxy. Name resolution tricks can be used to allow internal users to bypass the reverse proxy.

To generate outgoing Windchill URLs referencing the reverse proxy server, you must set the wt.httpgw.mapCodebase property in the wt.properties file using the xconfmanager utility. This property maps the Windchill server codebase to the reverse proxy codebase.

For example, if the reverse proxy base URL is https://rp.rphost.com/Windchill and the Windchill URL is http://wc.wchost.com/Windchill, then the following properties must be set:

```properties
wt.server.codebase=https://rp.rphost.com/Windchill
wt.httpgw.mapCodebase=http://wc.wchost.com/Windchill
```
If external users are required to use HTTPS while internal users are allowed to use HTTP, then dual Windchill servers should be used, one configured with HTTPS URLs and the other with HTTP URLs. Windchill background processing can happen on either configuration, as long as all users are able to access these URLs when e-mail links are followed by the users. The incorrect Web server may perform a server redirect to tell the user’s browser to access the appropriate server.

**Note:** These servers must be configured to communicate with one another as if they were in a load balancing cluster.
This appendix describes policies and mapping rules, and then describes conflict messages.

In addition to system defaults and actions available in the user interface, mapping rules and policy files can be used to control Windchill Import and Export processes. Mapping rules specify modifications to be made to the XML import or export files, while XSL-based policy files specify actions to be performed upon the attribute data of database objects during import or export. Mapping rules can be used in conjunction with either the import or export actions offered in the user interface or with policy files, but not both, during any given transaction.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSL-based Policy Files</td>
<td>C-2</td>
</tr>
<tr>
<td>Mapping Rules</td>
<td>C-4</td>
</tr>
<tr>
<td>Mapping Through Special Rules</td>
<td>C-4</td>
</tr>
<tr>
<td>Mapping Through XSL Transformation</td>
<td>C-11</td>
</tr>
<tr>
<td>Java Mapping with the METHOD Element</td>
<td>C-12</td>
</tr>
<tr>
<td>Hierarchical Instance Based Attribute Definitions, Exporting, and Importing</td>
<td>C-12</td>
</tr>
<tr>
<td>Conflict Messages</td>
<td>C-16</td>
</tr>
<tr>
<td>Reforming Custom Modeled Attributes</td>
<td>C-24</td>
</tr>
</tbody>
</table>
XSL-based Policy Files

Policy files can be written to apply to a specific export or import process or a set of such processes. Conditions set forth in the policy files can selectively apply actions available in the user interface. The following actions are available in import: Ignore, Create New Object, Substitute Object, and Unlock and Iterate. Policy files can apply the Lock action in export. You set properties in mapping rules files by editing the files, and you cannot use the xconfmanager utility for this purpose. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

Policy File Example

The following example shows the syntax of an XSL-based, import policy file. Comments explaining the use of policy files are embedded within the example:

```xml
<?xml version="1.0"?>
<!-- Use this file as an example and as documentation for Import policy. -->

<!-- The syntax of Import Policy is standard XSL syntax. The output of XSLT using the XSL policy file must have at most one element of the form:

  <actionInfo>
    <action>...</action>
    <actionParams>
      ...
    </actionParams>
  </actionInfo>

The element 'actionParams' holds additional information, which is necessary for the implementation of certain actions. Since each action may have its own list of parameters, no validation is made for the child elements of 'actionParams'. Thus, it is the USER’S RESPONSIBILITY to provide the correct parameters.

Currently there are 2 actions which require parameters: 'CreateNewObject' and 'SubstituteObject'. For both these actions, the list of parameter tags is as follows:

  <newNumber>
  <newName>
  <newVersion>
  <newIteration>

The meaning of these parameters is that they provide a new object identity, instead of the original one.

-->

<!-- For the detailed tutorial on XSL syntax and XSLT, see online: http://www.w3.org/TR/xslt.
```
For the list of action names, see 'wt.actor.actions.IxbActionsHelper'.
For the list of Windchill XML tags (to be used in 'test' statements), see 'src\wt\ixb\registry\dtds\standard62.dtd\coreobjects.dtd'.

```xml
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<!-- Import all Parts as New Iteration -->
<xsl:template match='WTPart'>
    <actionInfo>
        <action>NewIteration</action>
    </actionInfo>
</xsl:template>

<!-- Import the document with number='12345' as New Version;
instead of importing the document with number='99999', create
new document with name='My name', number='My number', etc.;
import remaining documents as New Iteration. -->
<xsl:template match='WTDocument'>
    <actionInfo>
        <xsl:choose>
            <xsl:when test="number='12345'">
                <action>NewVersion</action>
            </xsl:when>
            <xsl:when test="number='99999'">
                <action>CreateNewObject</action>
                <actionParams>
                    <newName>My name</newName>
                    <newNumber>My number</newNumber>
                    <newVersion>A</newVersion>
                    <newIteration>1</newIteration>
                </actionParams>
            </xsl:when>
            <xsl:otherwise>
                <action>NewIteration</action>
            </xsl:otherwise>
        </xsl:choose>
    </actionInfo>
</xsl:template>

<!-- Import CAD Document with name='MyTestDocument' as New Version; no action is defined for other CAD Docs -->
<xsl:template match='EPMDocument'>
    <actionInfo>
        <xsl:if test="name='MyTestDocument'">
            <action>NewVersion</action>
        </xsl:if>
    </actionInfo>
</xsl:template>
</xsl:stylesheet>
```
Mapping Rules

Windchill Import and Export allow mapping that either excludes attribute information, or maps it to other attributes during exporting and importing operations. Mapping attributes can adapt data to new environments that cannot accept the data in its original format. PTC supports three methods of mapping:

- mapping through special rules:
  Mapping through rules is the simplest method, but is not as powerful as mapping through XSL transformation.

- mapping through XSL transformation:
  Mapping through XSL transformation requires knowledge of XML and XSL. The XSL transformation functions are called by a form of special rule.

- mapping through rules that call Java functions:
  A software engineer with Java expertise is required to map data through rules that call Java functions.

Mapping rules can resolve situations announced by conflict messages during Windchill Import and Export.

This appendix describes mapping rules and then describes conflict messages.

Mapping Through Special Rules

Mapping rules can be written to apply to a specific export or import process or a set of such processes. The rules reside in either or both of two types of ASCII XML files that can also include properties that control import and export operations. You set properties in mapping rules files by editing the files, and you cannot use the xconfmanager utility for this purpose. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

- Client-based Files -- These files are selected by browsing in the graphical interface. These mapping rules file can have any name and can be located anywhere that the software can access and read them. These files govern if they conflict with generalized files.

- Generalized Files -- These files provide rules for either import operations or export operations. Their names must end in .xml. They are in either of two specific locations whose names define their functions:

  \Windchill\codebase\registry\ixb\export_settings

  \Windchill\codebase\registry\ixb\import_settings.

This appendix shows examples of the two type of files in the following two sections. After the examples you will see a section about properties, and several sections explaining rules (with examples).
Mapping Priorities

The four possible sources that control conflict resolution, have the following priority:

- **Import window** The *resolve overridable conflicts* checkbox in the Import window controls the property located in the wt.properties file named wt.ixb.import.overrideConflicts. This property enables the automatic resolution of folder and other conflicts. Import conflicts requiring the creation of a cabinet are not overridden. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

- Client-based files of mapping rules.
- Generalized files of mapping rules.
- Entries in the wt.properties file.

Client-based Mapping Rules Files

The rules and property values that appear in a client-based mapping file control Windchill Export and Import operations, overruling conflicting rules and values in the wt.properties file or a generalized mapping rules file. The `<debugProperties>` element is the location for properties, and it is not required. This element can include the import.parser.validate property that enables you to debug import operations by generating messages when the XML parser detects inconsistencies. The property that enables the automatic resolution of folder and other conflicts is named import.overrideConflicts when it appears in mapping files.

In a client-based mapping file the mapping rules occur in the `<mappingRules>` element.

Note that all the following examples can have the tag-value pair:

```
<path>...</path>
```

This tag-value pair allows the narrowing down of the elements applicable for the mapping rule. For example, the following mapping rule will change the value for tag `<number>` with value 1 to 4 for all XML files such as WTPart and WTDocument instances.

```
<COPY_AS>
  <tag>number</tag>
  <value>1</value>
  <newValue>4</newValue>
</COPY_AS>
```
If you wanted the preceding example to apply only to WTPart, the following example would achieve that by specifying the tag <path> and its value in the mapping rule:

```
<COPY_AS>
  <tag>number</tag>
  <path>WTPart</path>
  <value>1</value>
  <newValue>4</newValue>
</COPY_AS>
```

In this case, even though the number of a WTDocument instance is 1, its value will be still 1 instead of 4 for both import and export.

**Client-based Mapping Rules File Example**

The following example shows the syntax of a client-based mapping rules file.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<userSettings>
  <debugProperties>
    import.keepAllFilesInMemory=true
    client.log.level=10
    import.parser.validate=true
    import.default.lifecycleInfo.lifecycleState=RELEASED
    import.default.lifecycleInfo.lifecycleTemplateName=Released
    logLevel=5
    debug.enable=true
    mappingRules.log.enable=false
    mappingRules.debug.dir=C:\\TUNER_RESU
  </debugProperties>
  <mappingRules>
    <COPY_AS>
      <tag>number</tag>
      <value>1</value>
      <newValue>4</newValue>
    </COPY_AS>
    <COPY_AS>
      <tag>number</tag>
      <value>2</value>
      <newValue>5</newValue>
    </COPY_AS>
    <COPY_AS>
      <tag>number</tag>
      <value>*</value>
    </COPY_AS>
  </mappingRules>
</userSettings>
```
Generalized Mapping Rules File Example

The following example shows the syntax of a generalized mapping rules file. In such a file there is no `<mappingRules>` or `<debugProperties>` element. The properties that appear early in the file are not required and repeat properties that appear in the `wt.properties` file. The rules and property values that appear in a generalized mapping file, control Windchill Export and Import operations in the event that they conflict with entries in the `wt.properties` file. The rules and property values that appear in a generalized mapping file are overruled by conflicting values in a client-based mapping file.

```
import.keepAllFilesInMemory=true
client.log.level=10
import.parser.validate=true
import.default.lifecycleInfo.lifecycleState=RELEASED
import.default.lifecycleInfo.lifecycleTemplateName=Released
import.reposGuidPrefix=77746
logLevel=5
debug.enable=true
mappingRules.log.enable=false
mappingRules.debug.dir=C:\\TUNER_RESU

<newValue>N-05-*</newValue>
</COPY_AS>

<COPY_AS>
<tag>teamIdentity</tag>
<value>WWWWW*</value>
<newValue>System.Default</newValue>
</COPY_AS>

<COPY_AS>
<tag>folderPath</tag>
<value>*</value>
<newValue>/Administrator/NEW-FOLDER-22</newValue>
</COPY_AS>

<IGNORE_PARENT>
<tag>filename</tag>
<path>content</path>
<value>EngineReq</value>
</IGNORE_PARENT>
```

Import and Export Policies, Mapping Rules, and Conflict Messages
Properties in Mapping Rules Files

The preceding examples showed how to place properties in mapping rules files. The chapter in this document about Windchill Export and Import describes the properties that you can use to control Windchill Import. You set properties in mapping rules files by editing the files, and you cannot use the xconfmanager utility for this purpose. If you are not setting properties through a graphical user interface or in a mapping file, you add or edit properties with the xconfmanager utility, which is discussed elsewhere in this guide.

Do Not Map Number Attributes for MCAD Documents

The number attribute of an MCAD document in the Windchill database is a string identical to the document's Pro/ENGINEER file name. If you change an MCAD document's number attribute by a mapping rule or by altering the object when it is on the local disk, you create data that is incompatible with the assembly files that refer to it. Attempting to repair a number change by reverting to the original information does not succeed because the software perceives an attempt to check in the renamed object as an attempt to duplicate an existing object.

About Mapping Rules

Each mapping rule is an XML element within the mapping rule file. Each mapping rule element, except for one that specifies copying, has at least two sub-elements: <tag> and <value>. These two sub-elements determine whether the rule applies for any given element in an imported or exported XML file. If multiple rules in a file could apply to an element in an imported or exported file, only the first rule applies.

The following examples show the types of rules and how to apply them to a variety of attributes. To work with attributes that do not appear in the following examples, you need to understand XML and read the XML file that you are mapping.

COPY Element

By default, all elements in a source XML file are copied into the resulting XML file, and consequently it is not necessary to specify a rule that copies elements without alteration. If any rule specifies an action other than copying for an element, copying does not occur and the other rule controls the result for the
element. The only element in a rule that specifies copying is COPY, and it has no sub-element.

**COPY_AS Element**

Rules that use the COPY_AS element alter an element from a source XML file and place the altered element in a resulting XML file. A <newValue> sub-element is required in addition to <tag> and <value> sub-elements. The following examples show possible syntaxes:

Mapping an Object's View "Source_View" to View "Local_View"

```xml
<COPY_AS>
  <tag>view</tag>
  <value>Source_View</value>
  <newValue>Local_View</newValue>
</COPY_AS>
```

Mapping any Object's View to View "LOCAL_VIEW"

```xml
<COPY_AS>
  <tag>view</tag>
  <value>*</value>
  <newValue>Local_View</newValue>
</COPY_AS>
```

Mapping an Object's Number Attribute "2222" to Number "LOCAL_2222"

```xml
<COPY_AS>
  <tag>number</tag>
  <value>2222</value>
  <newValue>Local_2222</newValue>
</COPY_AS>
```

Mapping any Object's Number Attribute to a Number Constructed from the Prefix "FROM_SITE_AAA_ " and the Same Number

This example shows the number "2222" mapped to "From_Site_AAA_2222" in the resulting file.

```xml
<COPY_AS>
  <tag>number</tag>
  <value>*</value>
  <newValue>From_Site_AAA_*</newValue>
</COPY_AS>
```

Mapping an Object's Version "A" to Version "B"

```xml
<COPY_AS>
  <tag>versionInfo/versionId</tag>
  <value>A</value>
  <newValue>B</newValue>
</COPY_AS>
```
Mapping any Object's Version to Version "A" and any Iteration to Iteration "1"

```
<COPY_AS>
  <tag>versionInfo/iterationId</tag>
  <value>*</value>
  <newValue>1</newValue>
</COPY_AS>
```

Mapping any Object's Team That Begins with "MyTeam" to the Default Team

```
<COPY_AS>
  <tag>teamIdentity</tag>
  <value>MyTeam*</value>
  <newValue>System.Default</newValue>
</COPY_AS>
```

Mapping any Object's Folder to "Administrator/NewFolder"

```
<COPY_AS>
  <tag>folderPath</tag>
  <value>*</value>
  <newValue>/Administrator/NewFolder</newValue>
</COPY_AS>
```

Mapping Objects in Subfolders under the "Marketing" Folder to the Same Subfolders Under the "Publications" Folder Plus Some Folder Mapping Advice

```
<COPY_AS>
  <tag>folderPath</tag>
  <value>/Marketing/*</value>
  <newValue>/Publications/*</newValue>
</COPY_AS>
```

An asterisk (*) placed in the new and old value strings in folder mapping rules results in the creation of new folders in the position of the asterisk that duplicate the folders that existed in the old path in the position of the asterisk. The following is the most generalized syntax for such mapping rules:

```
<COPY_AS>
  <tag>folderPath</tag>
  <value>/PrefixOld*SuffixOld</value>
  <newValue>/PrefixNew*SuffixNew</newValue>
</COPY_AS>
```

Any string from PrefixOld, SuffixOld, PrefixNew, or SuffixNew could be an empty string.

**IGNORE Element**

Rules that use the IGNORE element exclude an element in a source XML file from a resulting XML file. The `<tag>` and `<value>` sub-elements are required. The following example shows a possible syntax.
Excluding Lifecycle State Information from a Resulting XML File

<IGNORE>
  <tag>lifecycleState</tag>
  <value>*</value>
</IGNORE>

**IGNORE_PARENT Element**

Rules that use the IGNORE_PARENT element exclude the parent of an element in a source XML file and all the child elements of that parent element from a resulting XML file. The <tag> and <value> sub-elements are required. As usual, the <path> element is optional. The following example shows a possible syntax.

Excluding an IBA value Named "Price" from an IBA Holder Such as WTPart

<IGNORE_PARENT>
  <tag>ibaPath</tag>
  <path>WTPart</path>
  <value>Price</value>
</IGNORE_PARENT>

In the preceding example, if the following line were deleted, all parent elements in all XML files with <ibaPath>Price</ibaPath> would be excluded.

<path>WTPart</path>

Excluding Parent Root Element

In the special case when the parent element to be excluded is the root element, the whole XML file will be excluded. This is equivalent to ignoring the corresponding object to exclude it from export. The following example creates a case in which the WTPart instances with number MyNumber will be excluded:

<IGNORE_PARENT>
  <tag>number</tag>
  <value> MyNumber </value>
</IGNORE_PARENT>

Mapping Through XSL Transformation

You can apply an XSL script to source file by specifying the script in the XML file that contains the user's settings. Set the property xsl.filename which is in that file's <properties> element to the full path name of the XSL script file name. The following example of a mapping file shows how to refer to an XSL script with the location C:\script1.xsl:

<?xml version="1.0" encoding="UTF-8"?>
Java Mapping with the METHOD Element

Mapping rules that reside in the <mappingRules> element of a user's settings file can invoke Java programs. The METHOD element has the two sub-elements <tag> and <value> and an additional required sub-element, <class>. The <tag> and <value> sub-elements identify the element in the XML source file for which Java programs will perform mapping. The sub-element <class> defines a name of Java class, that must have the method with the following specification:

```java
static public String mapElement (String path, String tag, String oldValue, IxbElement oldXmlFile) throws WTException;
```

This method will be called to get the new value for the specified element of the source XML file. It returns the element's new value as a return value, or it returns either of two special values:

- `wt.ixb.tuner.Tuner. S_IGNORE;`
- `wt.ixb.tuner.Tuner. S_IGNORE_PARENT;`

The S_IGNORE return value means (like the IGNORE element) that this element will be excluded from resulting XML file. The S_IGNORE_PARENT return value means (like the IGNORE_PARENT element) that the parent of this element will be excluded from result XML file.

The following example shows the syntax for applying Java programs to map the value of a number attribute. The example assumes the package wt.ixb and the class MapByJava:

```xml
<METHOD>
  <tag> number</tag>
  <value>*</value>
  <class>wt.ixb.MapByJava</class>
</METHOD>
```

Hierarchical Instance Based Attribute Definitions, Exporting, and Importing

Importing hierarchical Instance Based Attribute (IBA) definitions may require some preparation.

When to Use Mapping Files for Hierarchical IBAs

In Release 7.0 it is suggested that you do not create hierarchical IBA definitions unless the following line is present in the wt.properties file:

```text
wt.iba.definition.hierachicaldefinition.enabled=true
```
Setting the preceding property’s value true allows the import of hierarchical IBA definitions.

By default in Release 7.0, the default value of the property is false, and that value allows the creation of hierarchical IBA definitions. A false value for the property prevents the import of hierarchical IBA definitions, except when you use a properly written mapping file, called a mapping file. A mapping file maps hierarchical IBA definitions to non-hierarchical IBAs.

The creation of hierarchical IBA definitions without having the property set true is likely to have occurred in releases prior to 7.0 because no recommendation to set the property true existed for those releases. Nested Attribute Organizers are allowed in R7.0 as they were in prior releases, without regard to the property’s value.

This section describes the syntax of mapping files that provide rules to map hierarchical IBA definitions to non-hierarchical IBA definitions. Mapping files control both import and export and a given mapping file has the same effects on both import and export. Mapping files can be used at any time and for any XML files. Mapping files are more likely to be used with Windchill PDMLink than with Windchill, because Windchill PDMLink can use many containers while Windchill uses one layer.

A mapping file must map hierarchical IBA definitions to non-hierarchical IBA Definitions for both IBA definitions and for IBAHolder instances such as WTPart and TypeDefinition.

**How to Write a Mapping File for Hierarchical IBAs**

To understand how to write a mapping file, consider the case of a jar file to be imported named ibaDefinitions.xml, The file includes IBA definitions with the following structure:

```
Test Organizer
    Test String
        NestedTestString
```

The goal is to create a non-hierarchical StringDefinition of *NestedTestString* under the AttributeOrganizer *TestOrganizer* or any other AttributeOrganizer.

The following block achieves this goal:

```xml
<COPY_AS>
  <tag>path</tag>
  <value>TestOrganizer/TestString</value>
</COPY_AS>
```
For all XML files to be imported, the preceding mapping rule will change the values for tag `<path>` to `TestOrganizer`, if the original value for this tag is `TestOrganizer/TestString`. Changing the values for other XML files other than `ibaDefinitions.xml` may not be the result that is expected. For example, values would change for the file `ABC.xml` if it contains the following:

```xml
<path>TestOrganizer/TestString</path>
```

A general approach to overcome this problem is to supply the additional path for the `<path>` tag. For example, if we enhance the mapping rule to the following version, the change prevents the modification of the `<path>` value in `ABC.xml`:

```xml
<path>ibaDefinitions/StringDefinition</path>
<tag>path</tag>
<value>TestOrganizer/TestString</value>
<newValue>TestOrganizer</newValue>
```

The altered mapping rule would create the following `StringDefinition` in the database if it did not already exist:

```
Test Organizer
  NestedTestString
```

Another concern is mapping the IBA values to the appropriate IBA definitions for XML files corresponding to IBAHolder. In general terms, if a mapping rule is supplied for IBA definitions, a mapping rule should be supplied for the related IBA values. For example, consider a WTPart, `Tag_WTPart_0.xml`, which has the IBA values declared by the following block:

```xml
<iba>
  <ibaPath>TestString/NestedTestString</ibaPath>
  <ibaValue>My String value for NestedTestString</ibaValue>
  <ibaType>StringValue</ibaType>
</iba>
```
The tag `<ibaPath>` in the preceding code means the full path of the corresponding IBA definition is determined by disregarding the path of the AttributeOrganizer where it is nested.

To continue the example, the StringDefinition `TestOrganizer/TestString/NestedTestString` is mapped and created as `TestOrganizer/NestedTestString`, which allows mapping the definition of the preceding IBA value to `TestOrganizer/NestedTestString` as well. Therefore you could supply the following mapping rule:

```
<COPY_AS>
  <tag>ibaPath</tag>
  <value>TestString/NestedTestString</value>
  <newValue>NestedTestString</newValue>
</COPY_AS>
```

Similarly, if you only want to restrict your mapping for WTPart, you could achieve this by specifying the `<path>` value in the mapping rule, which is shown in the following example of the complete rule:

```
<?xml version="1.0" encoding="UTF-8"?>

<userSettings>
  <mappingRules>
    <COPY_AS>
      <!--The following line is optional-->
      <path>ibaDefinitions/StringDefinition</path>
      <tag>path</tag>
      <value>TestOrganizer/TestString</value>
      <newValue>TestOrganizer</newValue>
    </COPY_AS>
    <COPY_AS>
      <!--The following line is optional-->
      <path>WTPart/iba</path>
      <tag>ibaPath</tag>
      <value>TestString/NestedTestString</value>
      <newValue>NestedTestString</newValue>
    </COPY_AS>
  </mappingRules>
  <properties>
```

Conflict Messages

This section describes the conflicts that can arise from importing XML files into the Windchill database in the processes of Windchill Import and Export. Potential conflicts come from the fact that Windchill objects being imported exist already in the Windchill database, and the object properties of the imported and existing objects do not match. In this explanation, the term Windchill object refers to Parts, Documents, and EPMDocuments.

The following matrix lists the properties of Windchill objects that can cause the conflicts. Individual conflicts can be resolved through modification of the mapping rules. Because this type of resolution must be implemented manually for each object, it is a costly approach for importing large number of objects.

In general there are three types of conflicts:

- **Administrative conflicts** -- Mismatches between data infrastructure (for example, the existence of folders, life-cycle or IBA definitions) required by the imported object and the data definitions which exist in the target Windchill environment.

- **Dependency Conflicts** -- References in the imported object (for example, through part structure) to other business objects that do not exist in the target system.

- **Metadata conflicts** -- Mismatches between the metadata of objects (for example, name/number pair) in the target system with the metadata of imported object occurs.

Many conflicts are announced by a generic message that the software rewrites to fit each situation:

Object <type> already exists in database, but has different value for attribute <type>: existing value is <type>, new value is <type>.

**Importing RatioDefinition and RatioValue**

A particular change for the 7.0 release that can produce conflicts involves the RatioDefinition and RatioValue. These types of data, if included in an export from 6.2.6 or earlier, result in an overridable import conflict in the 7.0 release. If you override the conflict, the data is imported as FloatDefinition and FloatValue.
## Administrative Conflicts of Common Business Objects

<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBA name mismatch</td>
<td>User notification</td>
<td>&quot;Definition of Instance Based Attribute &lt;type&gt; cannot be found. See the Windchill Administrator's Guide for further information.&quot;</td>
</tr>
<tr>
<td>IBA datatype mismatch</td>
<td>User notification</td>
<td>&quot;Object &lt;type&gt; already exists in database, but has different type of Instance Based Attribute: existing type is &lt;type&gt;, new type is &lt;type&gt;.&quot;</td>
</tr>
<tr>
<td>IBA units of expression mismatch</td>
<td>User notification</td>
<td>&quot;Object &lt;type&gt; already exists in database, but has different value for Instance Based Attribute: existing value is &lt;type&gt;, new value is &lt;type&gt;.&quot;</td>
</tr>
</tbody>
</table>
| Denominator mismatch for Ratio values                   | User notification | Object &lt;type&gt; already exists in database, but has different type of Instance Based Attribute &lt;type&gt;: existing type is &lt;type&gt;, new type is &lt;type&gt;.
| Precision mismatch for Float, Ratio, and Unit values    | User notification | Object &lt;type&gt; already exists in database, but has different type of Instance Based Attribute &lt;type&gt;: existing type is &lt;type&gt;, new type is &lt;type&gt;.
| Existing IBAHolder has fewer IBAs                       | User notification | The existing Object &lt;type&gt; in database, but it does not have the value for Instance Based Attribute &lt;type&gt;. The value is &lt;type&gt;.
| Existing IBAHolder has more IBAs                        | User notification | The existing Object &lt;type&gt; in database, but it has an extra value for Instance Based Attribute &lt;type&gt;. The value is &lt;type&gt;.
| Type Identifier mismatch                                | User notification | Object existed with a different type. Existed type: &lt;type&gt;; expected type: &lt;type&gt;.
<p>| View Definition does not exist                          | User notification | &quot;Definition of View &lt;type&gt; cannot be found. See the Windchill Administrator's Guide for further information.&quot; |
| Life Cycle Definition does not exist                   | User notification | &quot;Definition of Life Cycle &lt;type&gt; cannot be found. See the Windchill Administrator's Guide for further information.&quot; |
| Life Cycle State does not exist in template             | User notification | &quot;Life Cycle State &lt;type&gt; cannot be found in the Life Cycle Template &lt;type&gt;. See the Windchill Administrator's Guide for further information.&quot; |</p>
<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template containing Life Cycle State cannot be found</td>
<td>User notification</td>
<td>&quot;Life Cycle State <code>&lt;type&gt;</code> cannot be found because the Life Cycle Template <code>&lt;type&gt;</code>, to which the State belongs, does not exist. See the Windchill Administrator's Guide for further information.&quot;</td>
</tr>
<tr>
<td>Domain containing team does not exist</td>
<td>User notification</td>
<td>&quot;Team <code>&lt;type&gt;</code> cannot be found because the Administrative Domain <code>&lt;type&gt;</code>, where team resides, does not exist.</td>
</tr>
<tr>
<td>Team does not exist</td>
<td>User notification</td>
<td>&quot;Team <code>&lt;type&gt;</code> cannot be found in the Administrative Domain <code>&lt;type&gt;</code>. See the Windchill Administrator's Guide for further information.&quot;</td>
</tr>
<tr>
<td>Full team name incorrect</td>
<td>User notification</td>
<td>&quot;Full team name must consist of <code>&lt;domain name&gt;?</code>?<code>team name?</code>&lt;br&gt;The current full team name is: `&lt;type&gt;!!!&quot;</td>
</tr>
<tr>
<td>Location (Cabinet) does not exist</td>
<td>User notification</td>
<td>&quot;Cabinet <code>&lt;type&gt;</code> cannot be found. See the Windchill Administrator's Guide for further information.&quot;</td>
</tr>
<tr>
<td>Location (Folder) does not exist</td>
<td>Create folder or user notification, (as checked)</td>
<td>&quot;Folder <code>&lt;type&gt;</code> cannot be found. See the Windchill Administrator's Guide for further information.&quot;</td>
</tr>
</tbody>
</table>
**Administrative Conflicts of Administrative Objects**

This section discusses conflicts for some common administrative objects such as IBA Definition, Attribute Organizer, Quantity of Measure, Measurement System, and Soft Type Definition.

<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Overridable</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>yes</td>
<td>User notification</td>
<td>Attributes are different for &quot;Description&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>yes</td>
<td>User notification</td>
<td>Attributes are different for &quot;Display Name&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Hierarchy Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>yes</td>
<td>User notification</td>
<td>Attributes are different for &quot;Hierarchy Display Name&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Attribute Organizer does not exist</td>
<td>yes</td>
<td>User notification</td>
<td>Attribute Organization &lt;type&gt; cannot be found. See the Windchill Business Administrator's Guide for further information.</td>
</tr>
<tr>
<td>Creating Hierarchical IBA Definitions not allowed</td>
<td>no</td>
<td>User notification</td>
<td>Hierarchical Definition &lt;type&gt; is not allowed.</td>
</tr>
<tr>
<td>IBA Definition mismatch</td>
<td>no</td>
<td>User notification</td>
<td>Attribute definition &lt;type&gt; is defined locally as &lt;type&gt;, but is imported as &lt;type&gt;. See the Windchill System Administrator's Guide for further information.</td>
</tr>
<tr>
<td>IBA Definition or Attribute Organizer does not exist</td>
<td>yes</td>
<td>User notification</td>
<td>Definition of Instance Based Attribute &lt;type&gt; cannot be found. See the Windchill System Administrator's Guide for further information.</td>
</tr>
<tr>
<td>Potential conflict</td>
<td>Overridable</td>
<td>Behavior</td>
<td>Resolution or Message</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Quantity of Measure does not exist</td>
<td>yes</td>
<td>User notification</td>
<td>Quantity of measure &lt;type&gt; does not exist.</td>
</tr>
<tr>
<td>Unit Definition exists with different base unit</td>
<td>no</td>
<td>User notification</td>
<td>The Quantity of measure &lt;type&gt; for Unit Definition &lt;type&gt; exists with different display units or overridden display units.</td>
</tr>
<tr>
<td>Measurement System mismatch for base symbol values</td>
<td>no</td>
<td>User notification</td>
<td>Measurement system &lt;type&gt; exists with different base symbol value.</td>
</tr>
<tr>
<td>Measurement System does not exist</td>
<td>yes</td>
<td>User notification</td>
<td>Measurement system &lt;type&gt; does not exist.</td>
</tr>
<tr>
<td>Type Definition does not exist</td>
<td>yes</td>
<td>User notification</td>
<td>Type Definition cannot be found: &lt;type&gt;</td>
</tr>
<tr>
<td>Attribute UserAttributable mismatch</td>
<td>no</td>
<td>User notification</td>
<td>Incompatible attribute &quot;userAttributeable&quot; for: &lt;type&gt;, expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Attribute Instantiable mismatch</td>
<td>no</td>
<td>User notification</td>
<td>Incompatible attribute &quot;instantiable&quot; for: &lt;type&gt;, expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Attribute Deleted mismatch</td>
<td>no</td>
<td>User notification</td>
<td>Incompatible attribute &quot;deleted&quot; for: &lt;type&gt;, expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Icon mismatch for Soft Type</td>
<td>yes</td>
<td>User notification</td>
<td>The icon &lt;type&gt; already exists. Overriding this conflict will rename the icon to a different name.</td>
</tr>
<tr>
<td>Existing Soft Type has fewer IBAs than in XML file</td>
<td>no</td>
<td>User notification</td>
<td>IBA value (attribute type: &lt;type&gt;, path: &lt;type&gt;, value: &lt;type&gt;) is expected with respect to import for: &lt;type&gt;</td>
</tr>
<tr>
<td>Existing Soft Type has extra IBA relative to XML file</td>
<td>no</td>
<td>User notification</td>
<td>Extra IBA value (attribute type: &lt;type&gt;, path: &lt;type&gt;, value: &lt;type&gt;) is found with respect to import for: &lt;type&gt;</td>
</tr>
<tr>
<td>Existing Soft Type has fewer Constraints than in XML file</td>
<td>no</td>
<td>User notification</td>
<td>Type constraint (enforcementRuleClassname: &lt;type&gt;, bindingRuleClassName: &lt;type&gt;, enforcementRuleData: &lt;type&gt;, IBA definition path: : &lt;type&gt;) is expected with respect to import for: &lt;type&gt;</td>
</tr>
</tbody>
</table>
### Dependency Conflicts

<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Overridable</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Soft Type has extra Constraints relative to XML file</td>
<td>no</td>
<td>User notification</td>
<td>Extra type constraint (enforcementRuleClassname: &lt;type&gt;, bindingRuleClassName: &lt;type&gt;, enforcementRuleData: &lt;type&gt;, IBA definition path: : &lt;type&gt;) is found with respect to import for: &lt;type&gt;</td>
</tr>
</tbody>
</table>

### Metadata Conflicts

#### Name and Number Conflict

<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imported Number matches while imported Name does not match</td>
<td>User notification. Conflict is overridable.</td>
<td>&quot;Warning: Name mismatch for part number &lt;part number&gt;&quot;</td>
</tr>
</tbody>
</table>
Default Values for Overridable Conflicts

Some import conflicts will cause import failure. This section explains the default values that are assigned when the listed conflicts are successfully overridden.

Life Cycle

There are 2 cases in which a life cycle conflict occurs:

- The life cycle of the object in the XML file doesn't exist in the database, in which case import fails -- a non-overridable conflict.
- The life cycle of the object in the XML file is different from the one in database, in which case the following import actions yield the following results:
  - Default -- The database object remains unchanged.
  - Import as latest iteration -- The life cycle of the newly created object is the life cycle of the previous iteration in the database.
  - Import as new version -- The life cycle of the newly created object is the life cycle from the XML file
  - Import as checked out -- The life cycle of the newly created object is the life cycle of the previous iteration in the database.
  - Modify non-versioned attributes -- The life cycle of the newly created object is the life cycle from the XML file.
  - Update checked out object in place-- The life cycle of the newly created object is the life cycle of the checked out object in the database.

Team

There are 2 cases in which a Team conflict occurs:

- The team of the object in the XML file doesn't exist in the database, in which case the team of the newly created object is the team of the previous iteration in the database.
- The team of the object in the XML file is different from the one in database, in which case the following import actions yield the following results:
  - Default -- The database object remains unchanged.
  - Import as latest iteration -- The team of the newly created object is the team from the XML file.
  - Import as new version -- The team of the newly created object is the team from the XML file.
  - Import as checked out -- The team of the newly created object is the team of the previous iteration in the database. This behavior is chosen because...
the team package doesn't provide an API method to reassign TeamTemplate for an object that is being checked out.

- Modify non-versioned attributes -- The team of the newly created object is the team from the XML file.
- Update checked out object in place-- The team of the newly created object is the team of the checked out object in the database. This behavior is chosen because the team package doesn't provide an API method to reassign TeamTemplate for an object that is being checked out.

**Domain**

There are 2 cases in which a domain conflict occurs:

- The domain of the object in the XML file doesn't exist in the database, in which case import fails -- a non-overridable conflict.
- The domain of the object in the XML file is different from the one in database, in which case the following import actions yield the following results:
  - Default -- The database object remains unchanged.
  - All other actions -- The domain of the newly created object is the domain of the existing object in the database.

**Folder**

There are 2 cases in which a folder conflict occurs:

- The domain of the object in the XML file doesn't exist in the database, in which case import fails -- a non-overridable conflict.
- The domain of the object in the XML file is different from the one in database, in which case the following import actions yield the following results:
  - Default -- The database object remains unchanged.
  - All other actions -- The folder of the newly created object is the folder of the existing object in the database.

**Context**

If there is no context mapping file, the object will be imported to the context from where the import process is launched.

If there is context mapping file, the object will be imported according to the mapping file. If the mapping file maps the object to a context that doesn't exist, the import process throws an exception.
IBA Value

Most conflicts for IBA Values are non-overridable in the following meaning. The following if violated make non-overridable conflicts:

- The IBA type should be the same if the IBA path are the same.
- The IBA values should be matching if the IBA path are the same.
- The XML file and the existing IBAHolder must have the same IBA values, including the number of IBA values.

Some conflicts are overridable, for example, the precision for float values, ratio values, and unit values.

Type Identifier

If an object is Typed, such as WTPart or WTDocument instance, then it will carry a value with tag <externalTypeId> which declares the associated soft type or hard type in the XML file. This value is always non-overridable unless they are matched exactly.

Organization ID

If the organization of Organization ID included in the export data is not found, the conflict is overridable. In such a case, if the software is configured to override conflicts, the default organization is used.

Reforming Custom Modeled Attributes

If an object with custom modeled attributes is exported from system A and then imported into system B where the object does not include the custom modeled attributes, the import fails. The custom modeled attributes should be exported as IBAs. This section explains how to write a mapping rule for use in export to insure that import will be successful in a such a case.

Using such a rule introduces achieves three goals:

- Using an export mapping rule like the one described in this section means that the custom modeled attributes will be ignored.
- The tags, especially the root tag, should be mappable so that the XML files can be handled by the import system.
- The DTD specified in the XML should be mappable so that the new DTD is recognized and the XML files can be validated by the import system.

As an example, assume there is a Class SubTypeOfWTPart, which extends wt.part.WTPart, with one additional attribute mySubTypeAttr1. At export side is the corresponding handler with a customer DTD Customer-DTD.dtd, which is not included in IXB framework. The export system has the handler to export SubTypeOfWTPart, but unfortunately the import system does not have this handler.
In order to make import successful, the export system can supply a mapping rule to achieve the preceding three goals. As for this example, the attribute mySubTypeAttr1 should be ignored and the tag SubTypeOfWTPart should be changed to WTPart, and the Customer-DTD.dtd should be changed to a DTD, which is understood by the import system, for example, standard70.dtd.

IXB framework supports two formats of mapping rule file on export in IXB: XML files and XSL files.

Example of Two Formats of Mapping Files

The following XML file exportMapping.xml and XSL file exportMapping.xsl are two examples.

XML Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<userSettings>
  <mappingRules>
    <IGNORE>
      <tag>mySubTypeAttr1</tag>
      <value>*</value>
    </IGNORE>
    <CHANGE_TAG>
      <tag>SubTypeOfWTPart</tag>
      <newTag>WTPart</newTag>
      <newDtd>standard70.dtd</newDtd>
    </CHANGE_TAG>
  </mappingRules>
</userSettings>
```

XSL Example

```xml
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="SubTypeOfWTPart">
    <xsl:choose>
      <xsl:when test="name='simplePart'">
        <mappingRules>
          <IGNORE>
        </IGNORE>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```
Ignoring an Attribute

To ignore an attribute, use the built-in command `<IGNORE>` in a syntax like the following:

```xml
<IGNORE>
  <tag>tagName</tag>
  <path>pathOfTheTag</path>
  <value>tagValue</value>
</IGNORE>
```

In the preceding syntax, the following line is optional:

```xml
<path>pathOfTheTag</path>
```

In the preceding syntax, you can use the wild card * in the following line:

```xml
<value>tagValue</value>
```

To continue the example (Class SubTypeOfWTPart, which extends wt.part.WTPart, with one additional attribute mySubTypeAttr1) tagName is mySubTypeAttr1 and tagValue is * (the wild card). This will ignore all mySubTypeAttr1 with any value.

If there is another object type with an attribute with the same name as mySubTypeAttr1, and this type is not to be ignored, including the type can be achieved by specifying the `<path>pathOfTheTag</path>`, for example, `<path>`
SubTypeOfWTPart</path>, which means the mySubTypeAttr1 will be ignored only if it is a tag under SubTypeOfWTPart.

Changing a Tag to a Different Name

To change a tag to a different name, use the built-in command <CHANGE_TAG>. The syntax for changing a tag to a different name is the following. You can write similar code to change the DTD by specifying the value of <newDtd>. Look at the two longer example files earlier in this topic to understand how to implement these changes.

<CHANGE_TAG>
   <tag>oldTagName</tag>
   <path>pathOfTheOldTag</path>
   <newTag>newTagName</newTag>
   <newDtd>newDTD</newDtd>
</CHANGE_TAG>

The following two lines in the preceding example are optional:
   <path>pathOfTheOldTag</path>
   <newDtd>newDTD</newDtd>

Administrative Conflicts of Common Administrative Objects

This section discusses conflicts for some common administrative objects such as IBA Definition, Attribute Organizer, Quantity of Measure, Measurement System, and Soft Type Definition.

<table>
<thead>
<tr>
<th>Potential conflict</th>
<th>Behavior</th>
<th>Resolution or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>User notification</td>
<td>Attributes are different for &quot;Description&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>User notification</td>
<td>Attributes are different for &quot;Display Name&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Hierarchy Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition</td>
<td>User notification</td>
<td>Attributes are different for &quot;Hierarchy Display Name&quot;. The value of attribute &lt;type&gt; is different from the value as found in database. Expected: &lt;type&gt;, found: &lt;type&gt;</td>
</tr>
<tr>
<td>Message</td>
<td>User notification</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attribute Organizer does not exist</td>
<td>User notification</td>
<td>Attribute Organization <code>&lt;type&gt;</code> cannot be found. See the Windchill Business Administrator's Guide for further information.</td>
</tr>
<tr>
<td>Creating Hierarchical IBA Definitions not allowed</td>
<td>User notification</td>
<td>Hierarchical Definition <code>&lt;type&gt;</code> is not allowed.</td>
</tr>
<tr>
<td>IBA Definition mismatch</td>
<td>User notification</td>
<td>Attribute definition <code>&lt;type&gt;</code> is defined locally as <code>&lt;type&gt;</code>, but is imported as <code>&lt;type&gt;</code>. See the Windchill System Administrator's Guide for further information.</td>
</tr>
<tr>
<td>IBA Definition or Attribute Organizer does not exist</td>
<td>User notification</td>
<td>Definition of Instance Based Attribute <code>&lt;type&gt;</code> cannot be found. See the Windchill System Administrator's Guide for further information.</td>
</tr>
<tr>
<td>Quantity of Measure does not exist</td>
<td>User notification</td>
<td>Quantity of measure <code>&lt;type&gt;</code> does not exist.</td>
</tr>
<tr>
<td>Unit Definition exists with different base unit</td>
<td>User notification</td>
<td>The Quantity of measure <code>&lt;type&gt;</code> for Unit Definition <code>&lt;type&gt;</code> exists with different display units or overridden display units.</td>
</tr>
<tr>
<td>Measurement System mismatch for base symbol values</td>
<td>User notification</td>
<td>Measurement system <code>&lt;type&gt;</code> exists with different base symbol value.</td>
</tr>
<tr>
<td>Measurement System does not exist</td>
<td>User notification</td>
<td>Measurement system <code>&lt;type&gt;</code> does not exist.</td>
</tr>
<tr>
<td>Type Definition does not exist</td>
<td>User notification</td>
<td>Type Definition cannot be found: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Attribute UserAttributable mismatch</td>
<td>User notification</td>
<td>Incompatible attribute &quot;userAttributable&quot; for: <code>&lt;type&gt;</code>, expected: <code>&lt;type&gt;</code>, found: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Attribute Instantiable mismatch</td>
<td>User notification</td>
<td>Incompatible attribute &quot;instantiable&quot; for: <code>&lt;type&gt;</code>, expected: <code>&lt;type&gt;</code>, found: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Attribute Deleted mismatch</td>
<td>User notification</td>
<td>Incompatible attribute &quot;deleted&quot; for: <code>&lt;type&gt;</code>, expected: <code>&lt;type&gt;</code>, found: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Icon mismatch for Soft Type</td>
<td>User notification</td>
<td>The icon <code>&lt;type&gt;</code> already exists. Override this conflict will rename the icon to a different name.</td>
</tr>
<tr>
<td>Existing Soft Type has fewer IBA than in XML file</td>
<td>User notification</td>
<td>IBA value (attribute type: <code>&lt;type&gt;</code>, path: <code>&lt;type&gt;</code>, value: <code>&lt;type&gt;</code>) is expected with respect to import for: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Existing Soft Type has extra IBA relative to XML file</td>
<td>User notification</td>
<td>Extra IBA value (attribute type: <code>&lt;type&gt;</code>, path: <code>&lt;type&gt;</code>, value: <code>&lt;type&gt;</code>) is found with respect to import for: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Existing Soft Type has fewer Constraints than in XML file</td>
<td>User notification</td>
<td>Type constraint (enforcementRuleClassname: <code>&lt;type&gt;</code>, bindingRuleClassName: <code>&lt;type&gt;</code>, enforcementRuleData: <code>&lt;type&gt;</code>, IBA definition path: : <code>&lt;type&gt;</code>) is expected with respect to import for: <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>Existing Soft Type has extra Constraints relative to XML file</td>
<td>User notification</td>
<td>Extra type constraint (enforcementRuleClassname: <code>&lt;type&gt;</code>, bindingRuleClassName: <code>&lt;type&gt;</code>, enforcementRuleData: <code>&lt;type&gt;</code>, IBA definition path: : <code>&lt;type&gt;</code>) is found with respect to import for: <code>&lt;type&gt;</code></td>
</tr>
</tbody>
</table>
This appendix explains how to customize Windchill online help. Customizers of online help should have advanced knowledge of HTML and JavaScript, and some familiarity with XML.

### Topic Page

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebHelp Overview</td>
<td>D-2</td>
</tr>
<tr>
<td>Customizing Topic Content</td>
<td>D-3</td>
</tr>
<tr>
<td>Customizing Navigation Pane Content</td>
<td>D-6</td>
</tr>
<tr>
<td>Customizing Topic Appearance</td>
<td>D-10</td>
</tr>
</tbody>
</table>
WebHelp Overview

Windchill solutions deliver online help in the WebHelp format provided by the eHelp Corporation. WebHelp is a cross-browser, HTML-based format that provides a three-pane window. The top pane contains buttons to select navigation features, the left pane contains the table of contents and search form for navigating and searching the help system; the online help content appears in the right pane.

In WebHelp, the term topic refers to a single HTML file that is displayed in the right pane of the WebHelp window. The term WebHelp system refers to a collection of topic files and the corresponding table of contents and full-text search files.

Several WebHelp systems are delivered with Windchill. They are stored in `<windchill>/wt/helpfiles/help_xx/online`, where `<windchill>` is the Windchill installation directory and `xx` is the two-character language suffix (for example, en for English).

You do not need a help compiler or other specialized software in order to customize WebHelp. This appendix assumes that you have access to the online help files and a text editor.

Customization Summary

WebHelp is a cross-browser format that uses dynamic HTML (DHTML) to display the table of contents and full-text search navigation tools. The following table summarizes the customizations you can make in WebHelp:

<table>
<thead>
<tr>
<th>Customization</th>
<th>DHTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit, add, and delete topic content.</td>
<td>Yes</td>
</tr>
<tr>
<td>Edit existing table of contents entries/links and search results/links.</td>
<td>Yes</td>
</tr>
<tr>
<td>Delete table of contents entries/links and search results/links.</td>
<td>Yes</td>
</tr>
<tr>
<td>Add table of contents entries and links.</td>
<td>Yes</td>
</tr>
<tr>
<td>Add search results and links.</td>
<td>No</td>
</tr>
<tr>
<td>Change background color of topics and make other basic topic style changes(^1).</td>
<td>Yes</td>
</tr>
<tr>
<td>Add navigation tools in the left frame.</td>
<td>No</td>
</tr>
<tr>
<td>Add graphics and text and modify color at the top of the navigation pane (above the tabs).</td>
<td>No</td>
</tr>
<tr>
<td>Change icons on Contents tab.</td>
<td>No</td>
</tr>
</tbody>
</table>
The rest of this appendix provides detailed instructions on how to make these customizations.

**Caution:** When you customize online help, always work with a copy of the online help files. After you have ensured that your customizations work properly, you can copy your changes to the correct directory.

### Customizing Topic Content

This section describes how to add, edit, and delete WebHelp topic content.

#### Adding a New Topic

To add a topic to a WebHelp system, simply create an HTML file using any standard HTML editor or text editor. Make sure you save the topic with an .html extension, rather than .htm or another extension.

To make sure the headings and other styles are correct, you may want to insert a temporary link to the WebHelp cascading style sheets (CSS), nm.css. This style sheet is stored in each online help directory. (You insert the permanent link to this style sheet later.)

In addition to providing style rules for standard elements, such as headings, nm.css defines several styles that you may want to use in your new topic. For additional information, see Customizing Topic Appearance.

When you have finished writing the content and applying standard styles, you must make the following modifications to the file:

- Add standard comments and links to the topic `<HEAD>` element.
- Add standard scripts and script references to the topic `<BODY>` element.

These modifications are described in detail in the following sections.

#### Modifying the Topic Head Element

In order to use PTC standard styles and to successfully integrate the new topics in the help, each topic must include the standard help system header. Insert this code in the `<HEAD>` element of your new topics below the TITLE element:

```html
<link rel='stylesheet' href='nm_ns.css'>
<script type="text/javascript" language="JavaScript" title="WebHelpSplitCss">
<!--
if (navigator.appName !="Netscape")
{
  document.write("<link rel='stylesheet' href='nm.css'>");
}
</script>
```
Modifying the Topic Body Element

Each online help topic must include the following lines in the <BODY> element, immediately after the opening <BODY> tag, preceding the <H1> heading and any other content:

```html
<script type="text/javascript" language="javascript1.2" src="whmsg.js"></script>
<script type="text/javascript" language="javascript" src="whver.js"></script>
<script type="text/javascript" language="javascript1.2" src="whproxy.js"></script>
<script type="text/javascript" language="javascript1.2" src="whutils.js"></script>
<script type="text/javascript" language="javascript1.2" src="whtopic.js"></script>
```

```html
</script>
```

```html
//-->
<style type="text/css">
</style>
```

D-4 Windchill System Administrator’s Guide

Windchill System Administrator’s Guide
setRelStartPage("help start page.html");

autoSync(1);
sendSyncInfo();
sendAveInfoOut();
}
else
    if (window.gbIE4)
        document.location.reload();
//-->
</script>

Note: In the preceding code, the addTocInfo() value must be changed to reflect the desired position in the table of contents navigation tool (TOC), and the setRelStartPage() value must be changed to the name of the WebHelp main file.

• To determine the name of the main file for the setRelStartPage() value, open the WebHelp in a browser. View the source of any topic and search for setRelStartPage. The value in the setRelStartPage() function is the WebHelp main file. Use relative path notation if the new topic will reside in a sub-directory.

• To determine the value for the addTocInfo() function, open the WebHelp in a browser, open the contents tab and browse to the desired heading for the new topic. Open any topic in this heading and search for addTocInfo(). Use the value in the addTocInfo() function call but replace the existing topic name with the name of the new topic. The topic will also need to be added to the table of contents resource file. To add the new topic or a new section of headings to the contents tab, see Customizing Navigation Pane Content.

Add the following lines immediately before the closing </BODY> tag:

<script type="text/javascript" language="javascript1.2">
<!--[if (window.writeIntopicBar)]]>
    writeIntopicBar(0);
//-->
</script>

Modifying an Existing Topic

To modify the content of an existing topic, use a standard HTML editor or text editor. Make sure you do not modify the following parts of the existing topic:

• Comments that refer to RoboHelp or eHelp
• Comments and other elements that contain the text "kadov"
• Script elements
• Style classes
All other aspects of existing topics can be customized. You can also add links to external files, as well as CSS references, other DHTML and JavaScript, forms, frames, and images, just as you would in any other HTML page.

Note: Although you can reference topics outside the WebHelp directory, you cannot add such external topics to a WebHelp system's table of contents or full-text search.

Deleting an Existing Topic

To delete an existing help topic, simply delete the HTML file from the appropriate directory. You should also delete references (within other topics) to that topic. To do so, use Windows Explorer or a search utility to search the online help HTML files for references to the deleted file. For example, if you deleted the file ObjectOview.html, you would search for and delete references similar to the following:

```html
<a HREF="ObjectOview.html">About Objects</a>
```

For information about deleting the TOC entry and search results that correspond to a deleted HTML file, see the next section, Customizing Navigation Pane Content.

Customizing Navigation Pane Content

This section describes how to add, edit, and delete text and links in a WebHelp table of contents (TOC). It also describes how to edit and delete full-text search results and links (currently, WebHelp does not support the addition of new search results and links). This section does not describe how to change a tab's name, color, or other aspects related to appearance.

DHTML Table of Contents

Overview

The DHTML navigation pane uses files located in the whxdata directory to define the table of contents. The whxdata/whtoc.xml file lists one or more XML resource files needed to build the table of contents. The whxdata/whtdata##.xml files contain the table of contents references, where ## in the filename is a number starting with 0.

The XML structure to build the table of contents in the whxdata/whtdata##.xml files is as follows:

```xml
<tocdata>
  <item name="NAME" url="URL" />
  <book name="NAME">
    <item name="NAME" url="URL" />
    <item name="NAME" url="URL" />
    <book name="NAME">
      <item name="NAME" url="URL" />
      <item name="NAME" url="URL" />
    </book>
  </book>
</tocdata>
```
Modifying Text and Links

The following sections describe how to edit book names and entries, delete books and entries, and add books and entries.

Editing a Book Name

To edit the name of a book in a DHTML TOC, open each whxdata/whtdata##.xml file in a text editor until you locate the desired <book> entry. Change the name value of the <book> entry to modify the book name displayed in the table of contents. For example, to edit the name of a TOC book called Home, you would modify the following element in whxdata/whtdata0.xml:

```xml
<book name="Home">
  <item name="Creating a Product" url="AdminProdCreate.html" />
  <item name="Updating a Product" url="AdminProdUpdate.html" />
  <item name="Current Product" url="AdminProdToTemplate.html" />
</book>
```

Editing an Entry

To edit the text or destination of an entry in a DHTML TOC, open each whxdata/whtdata##.xml file in a text editor until you locate the desired <item> entry. Make the necessary changes to the NAME and URL values of the <item> entry. For example, to change the topic name "About Windchill Administration", update the name value of the <item> entry, and to update the URL to point to a different hyperlink target, change url value of the <item> entry:

```xml
<item name="About Windchill Administration" url="WCAdmin.html" />
```

Deleting an Entry or Book

To delete a TOC entry or book, remove the <book> or <item> entry from the whxdata/whtdata##.xml file. Removal of a <book> tag requires removal of the closing </book> tag per XML rules.

Note: URL values are relative to the WebHelp system root directory (the directory containing the whxdata directory). Complete URL paths including protocol reference are also allowed, but they will be resolved in the client browser where the help is displayed.
Adding an Entry or Book

To add a TOC entry or book, insert new entry in an appropriate whxdata/whtdata##.xml file. For example:

```xml
<tocdata>
  <item name="About Windchill Administration" url="WCAdmin.html" />
  <book name="Home" >
    <item name="Creating a Product" url="AdminProdCreate.html" />
    <item name="Updating a Product" url="AdminProdUpdate.html" />
    <book name="New Book">
      <item name="New Item 1" url="NewItemURL1.html" />
      <item name="New Item 2" url="NewItemURL2.html" />
    </book>
    <book name="Current Product" url="AdminProdToTemplate.html" />
  </book>
</tocdata>
```

DHTML Search

Overview

The DHTML navigation pane uses files located in the whxdata directory to identify and display search results. WebHelp maintains a list of all help topics in the help, and a list of all words present in those topics. In general, the list of topics is counted via JavaScript, and each word present in the help is listed in an array accompanied by the topic numbers of the matching topics.

The search data in WebHelp is built starting with the whxdata/whfts.xml file. Please see the following example for reference:

```xml
<fts>
  <chunkinfo url="whfwdata0.xml" first="200" last="made"/>
  <chunkinfo url="whfwdata0.xml" first="make" last="zip"/>
  <tchunkinfo first="0" last="12" url="whftdata0.xml"/>
  <tchunkinfo first="13" last="24" url="whftdata1.xml"/>
</fts>
```

Find the values of the `<chunkinfo>` elements in the above example (there may be one or more of these elements). If there is only one `<chunkinfo>` element, all the words present in the help are located in one file. The url value in each `<chunkinfo>` element identifies the XML file with a list of words present in the help, the first value is the starting word in the current `<chunkinfo>` file, and the last value is the ending word in the current `<chunkinfo>` file.

Find the values of the `<tchunkinfo>` elements in the above example (there may be one or more of these elements). The first value of the current `<tchunkinfo>` element is the help topic number of the first topic in the current `<tchunkinfo>` topic list. The last value of the `<tchunkinfo>` element is the help topic number of the last topic in the current `<tchunkinfo>` topic list. The url value is the file name of the current tchunkinfo topic list.
For search, the list of help topics present in Webhelp is contained in files named whxdata/whftdata##.xml (from the <tchunkinfo> elements above). See the example below (this is the whftdata0.xml file referenced in the first <tchunkinfo> element above, xml header omitted):

```
<ftstdata>
  <topic name="About Windchill Administration" url="AdminAbout.html" />
  <topic name="Creating a Document Template" url="AdminDocTemplateCreate.html" />
  <topic name="Updating a Document Template" url="AdminDocTemplateUpdate.html" />
  <topic name="Creating a Library" url="AdminLibCreate.html" />
  <topic name="Using Library Context as a Template" url="AdminLibToTemplate.html" />
  <topic name="Updating a Library" url="AdminLibUpdate.html" />
  <topic name="Creating a Product" url="AdminProdCreate.html" />
  <topic name="Using Product Context as a Template" url="AdminProdToTemplate.html" />
  <topic name="Updating a Product" url="AdminProdUpdate.html" />
  <topic name="About Teams" url="AdminTeamAbout.html" />
  <topic name="Adding Users to a Context Team" url="AdminTeamCreate.html" />
  <topic name="About Administering Templates" url="AdminTemplates.html" />
  <topic name="About Administration Utilities" url="AdminUtilities.html" />
</ftstdata>
```

The number of topics in the list matches the range of numbers specified in the <tchunkinfo> element in the whxdata/whfts.xml file (zero through twelve, inclusive; total thirteen).

For search in WebHelp, the list of words present in the help topics is contained in files named whdata/whfwdata##.xml (from the <chunkinfo> elements above). See the example below (this is a portion of the whfwdata0.xml file referenced in the first <chunkinfo> element above):

```
<ftswdata>
  <key name="200">3,6</key>
  <key name="2000">1,2,14</key>
  <key name="25">6,16,17</key>
  <key name="32">1</key>
  <key name="40">3,6</key>
  <key name="50">6</key>
  <key name="60">1,15,19,21</key>
  <key name="64">21,19,1,2</key>
  <key name="abl">1</key>
  <key name="accept">1,6,2</key>
  <key name="acces">11,12,9,0,10,1,3,6,5,8,4,7</key>
  <key name="accessibl">3,6,5,8</key>
  <key name="locat">1,2</key>
  <key name="logo">18,1</key>
  <key name="low">0</key>
  <key name="made">0</key>
</ftswdata>
```

The <key> elements are listed alphabetically by name value, the list of numbers between the <key> opening and closing tags is terminated with a comma (,), and
that the list of numbers corresponds to the topic numbers referenced in the
<chunkinfo> elements above. The numbers between the <key> opening and
closing tags should be organized by relevance, with most relevant topic number
first.

Modifying Results and Links

To edit the text that appears in the search results list for a particular topic, open
each whxdata/whftdata##.xml file until the desired topic reference is located and
modify the name value in the <topic> element.

Similarly, to change the list of topics that correspond to a particular search result,
open each whxdata/whfwdata##.xml file until the desired word present in the
Webhelp is located in the name value of the <key> element, and modify the list of
numbers between the appropriate <key> tags.

Deleting Results

To prevent a topic from appearing in search results, remove the topic number
corresponding to the relevant topic from the number list between the <key>
opening and closing tags in each whxdata/whfwdata##.xml file. To identify the
topic number, open each whxdata/whfwdata##.xml file until the desired topic
reference is located. Note the number of the <topic> element by counting from the
top of the list starting with zero. Open the whxdata/whfts.xml and find the
<chunkinfo> element with the same url name as the whxdata/whfwdata##.xml file
where the <topic> reference is present. Note the start value. The topic number is
the start value of the correct <chunkinfo> element plus the count number you
noted above.

In the whfwdata0.xml file shown above, the "Creating a Library" <topic> element
is topic number three (topic number three counting from the top, with
<chunkinfo> first value zero).

Customizing Topic Appearance

WebHelp systems use two linked cascading style sheet files: one for Internet
Explorer and one for Netscape. The CSS files delivered with PTC online help are
named nm.css (for Internet Explorer) and nm_ns.css (for Netscape). A copy of
each of these files is stored with each WebHelp system. When you customize the
online help styles, you must modify each CSS file.

Because WebHelp scripts reference style names and require certain style rules,
you cannot simply replace the supplied style sheets with your own. You must edit
the nm.css and nm_ns.css files as they are shipped with Windchill.

Note: Embedded style sheets and inline styles do not affect WebHelp scripts.
You can add standard embedded styles and inline styles to topics without making
modifications to the CSS files or other WebHelp components.
The following table lists the styles that you are most likely to use in your topics.

<table>
<thead>
<tr>
<th>Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.Topic-Text-Bulleted</td>
<td>Used for bulleted lists</td>
</tr>
<tr>
<td>P.Topic-Text-Subbulleted</td>
<td>Used for second level bulleted lists</td>
</tr>
<tr>
<td>P.Topic-Text-Numbered</td>
<td>Used for numbered lists</td>
</tr>
<tr>
<td>P.Topic-Text-SubNumbered</td>
<td>Used for second level numbered lists</td>
</tr>
<tr>
<td>P.Table-Heading</td>
<td>Used for table heading rows</td>
</tr>
<tr>
<td>P.Table-Text</td>
<td>Used for text in the body of a table</td>
</tr>
<tr>
<td>P.Syntax</td>
<td>Used for monospace text (for example, code)</td>
</tr>
<tr>
<td>P.Syntax-indent</td>
<td>Used for indented monospace text</td>
</tr>
</tbody>
</table>

**Note:** Both CSS files include style selectors that contain the word "kadov." Do not modify these selectors; they are used by WebHelp scripts. (You do not need to define corresponding kadov selectors for new styles you create.)
### A

Adding
- content holder data formats, 7-4
- folders, 3-10
- hosts, 3-9

Architecture
- three-tier, A-3
- Web-based, A-2

attribute mapping, 10-9
- download behavior, 10-11
- explicit, 10-10
- implicit, 10-10
- upload behavior, 10-10

Authentication, 1-38–1-39
- troubleshooting, 1-39–1-40
- user, A-6, B-3

### B

Background queues. See Queues

Backup
- and replica sites, 4-26
- process, 1-34
- bandwidth, 10-25

Bootstrap client, 2-1–2-3
- configuration file, 2-3
- JAR file cache location, 2-3

Bootstrap package
- properties, 2-4–2-7
- versioning, 2-7

Bulk Index Tool, 9-6, 9-7

Bulk loading, 9-6

### C

Caches
- central cache vault, 3-4
- local, 4-23
- mirroring in, 4-25
- moving data from, to master site, 4-26
- vaults for, 4-25

Canceling revaulting schedule items, 3-20

Central Cache Vault, 3-4

Changing file location in external vaults, 3-23

Codebase
- administering, 2-8
- MakeJar script, 2-9
- server property, B-5
- codebase directory, 1-25

Collections, 9-1
- bulk loading, A-22
- defining, 9-2
- language processing, 9-12
- compression, 10-24
- config.pro options, 10-3

Configuration properties, 1-2, B-9

Configuring, 1-35
- bootstrap client, 2-3
- content handling, 7-3
- firewalls, B-11
- queues, 8-4
- Windchill environment, 1-35

Conflict messages
- administrative objects, C-19
- common administrative objects, C-27
- common business objects, C-17
- dependency, C-21
- metadata, C-21

Content Cache Server, 4-22

Content holders, 7-1
- data formats, 7-4
- overview, 7-2

Content replication, 4-1–4-26
- compression, 4-21
- importing certificates, 4-16
- keys, 4-4
- master Configuration, 4-14
- master sites, 4-5
- overview, 4-2
- performance, 4-22
- replica configuration, 4-14
- security, 4-16
- summary, 3-2
- two types of replica sites, 4-11
- Windchill Visualization Service, 4-19

content replication, 10-23

Context in import and export, 6-4
Context mapping files, C-12
Convera, 9-5, A-21
Converting multiple vaults to single vault, 3-21
Creating
   external file vaults with FvLoader, 5-1, 5-2
   file vault policies, 3-13–3-16
   master site, 4-5
   mounts, 3-11
   replica site, 4-7
   revaulting schedule items, 3-18
   security keys for content replication, 4-10
   user preferences, 1-33
   vault rules with FvLoader, 5-1, 5-2
   vaulting rules, 3-14
Custom modeled attributes, reforming, C-24
  customizing
     naming service, 10-13

D
  data compression, 10-24
  Database components, A-17
  date format, 10-9
  Default value, setting, 1-16
  Delegates
     creating, 1-32
     preference hierarchy, 1-30
     scope, 1-33
  Deleting
     folders, 3-11
     revaulting schedule items, 3-21
     vaulting rules, 3-16
  Dependency conflicts, C-21
  Desktop Integration, 1-25
  Downloading
     JAR files, 2-8

E
  Editing
     property files, 1-2
     user preferences, 1-33
  Enterprise search, 1-25
  Event Console, 10-25
  Export, 6-1
     Access control, 6-22
     EPMDocuments, 6-9
     EPMDocuments as Checked Out, 6-10
     graphical user interface, 6-11
     Hierarchical Instance Based Attribute Definitions, C-12
     mapping rules, C-4
  object from its properties page, 6-14
     summary, 3-2
     XSL-based policy files, C-2
  Exporting an object from its properties page, 6-14
  external file vaulting, 10-23
  External Storage Administrator window, 3-8
  External Storage Scheduling window, 3-17

F
  File Size Limit for Content Replication, 4-20
  File Vault Administrator, 3-4, 3-8–3-12
  File vaults, 10-23
     adding, 3-10
     creating policies for, 3-13–3-16
     creating with FvLoader, 5-1, 5-2
     deleting, 3-10
     deleting folder, 3-11
     external, 3-1
        overview, 3-2
     external in replication, 4-2
     maintaining, 3-12
     properties for, 3-6
     replica, 4-2
     rules for, 3-13–3-14
        creating, 3-15
        deleting, 3-16
        summary, 3-2
        updating, 3-10
     See also Revaulting
  Folders
     adding, 3-10
     deleting, 3-11
     updating, 3-10
  Forcing Content to Vault, 3-21
  Full-scale replica site, 4-11
  FvLoader, 5-1, 5-2

H
  heap size, 10-24
  Hierarchical Instance Based Attribute Definitions, C-12
  Hierarchy
     preference, 1-30
     scope, 1-31
  Hosts
     adding, 3-9
     deleting, 3-11
     updating, 3-9
  moving for server with external vaults, 3-24
  updating, 3-9
HTTP, 10-26
  gateway, A-6
  server, A-6
HTTPS, 10-26

I
ie.properties file, 1-37
Import, 6-1
  Access control, 6-22
  Conflict messages, C-16
  EPMDocuments, 6-9
  EPMDocuments as Checked Out, 6-10
  graphical user interface, 6-16
  Hierarchical Instance Based Attribute Definitions, C-12
  mapping rules, C-4
  overridable conflicts, C-22
  properties, 6-22
  summary, 3-2
  XSL-based policy files, C-2
Index Policy Manager, 9-3
Indexing, 9-3, A-20
  file content, A-20
  policies, A-21
  properties for, 9-13
  publishing, A-21
INI files
  about, 10-6
  date format, 10-9
  overriding, 10-7
  specifying templates, 10-8
  internal organization, 1-23
  Internationalization, A-3

J
JAR files
  downloading, 2-8
  importing and exporting, 6-2
  security, 2-9
Java applets, A-5
Java database connectivity (JDBC), A-3
Java Mapping, C-12
Java Performance Guide, 10-25
Java platform support, A-2
Java property files. See Property files
Java RMI servlet, B-11
Java Server Pages, 1-18
Java system property settings
  bootstrap client, 2-6
JSPs. See Java Server Pages

K
  Keys
    content replication, 4-4
      creating, 4-10

L
  latency, 10-25
  Lightweight replica site, 4-11
  Load balancing, 1-36–1-37
  Local Upload, 4-22
  log, 1-6, 10-25
  Log files, 1-9
  and replica sites, 4-26
  e-mail, 1-10
  maintaining, 1-34
  method server, 1-9
  queues, 8-5
  usage, A-17
  viewing, 1-6

M
  Mapping rules, C-1
    Context mapping files, C-12
    Java mapping, C-12
    METHOD Element, C-12
    overview, C-4
    special rules, C-4
    XSL transformation, C-11
  Master sites, 4-5
    configuration, 4-14
    local cache, 4-26
  Maximum File Size for Content Replication, 4-20
  Meeting Center, 1-19
  message log, 10-25
  metadata compression, 10-24
  Metadata conflicts, C-21
  METHOD Element, C-12
  Method server
    log file, 1-9
  Method Servers, 3-3
    component, A-14
    home page, 1-18
    multiple
      load balancing for, 1-36–1-37
  Microsoft Office, 1-25
  Mounts
    creating, 3-11
    updating, 3-11
  Moving server with external vaults to new host, 3-24
multiple servers, 10-23

N
Null value, setting, 1-16

O
Object Relational Database Management System (OR-DBMS), A-17
Oracle database
  backing up, 1-34
Oracle database in replication, 4-2
Oracle repository, 4-2
Organization ID, 1-24
Organization ID Type, 1-23
Organizations
  administering, 1-21
  containers, 1-21
  internal, 1-23
  restricted, 1-22
Overridable conflicts, C-22

P
parameter mapping, 10-9
  explicit, 10-10
  implicit, 10-10
Performance Tuning, 10-24
Policies for Import and Export, C-1
Preferences
  creating, 1-33
  defining scopes, 1-29
  delegates, 1-30, 1-33
  editing, 1-33
  hierarchy, 1-29–1-30
  hierarchy, defined, 1-29
  macros, 1-30
  managing, 1-32
  removing, 1-33
  scopes, 1-32
  structure, defined, 1-29
Property files
  See also wt.properties file
    editing, 1-2
    listing values, 1-17
    modifying, 1-6, 1-7
Proxy servers
  client-side, B-12
  Meeting Center, 1-20
  server-side, B-12

PTC
  documentation, xxii
  technical support, xxi

Q
Queue Manager, 8-2, 8-3
Queues
  configuring, 8-4
  entry states, 8-3
  groups, 8-2
  log files, 8-5
  maintaining, 8-8
  managing, 1-6

R
Reforming custom modeled attributes, C-24
Removing user preferences, 1-33
Replica sites, 4-22
  configuration, 4-14
  creating, 4-7
  installing, 4-11
Replication Administrator, 4-5
restricted organization, 1-22
RetrievalWare
  backup, 1-34
  index loader, A-22
  indexing, A-21
  libraries, 9-8
  purging, 9-12
Revaulting, 3-16–3-21
  schedule items, 3-17, 3-18
Revaulting History window, 3-18
RMI, 10-26, B-8
RMI-over-HTTP, B-9
Runtime environment, A-1
Runtime services chapter, 1-1

S
Scopes, preferences, 1-32
secret.text, 1-37
secret.text2, 1-37
Security
  Infrastructures, B-1
  JAR files, 2-9
Server, 10-23
Server codebase property, B-5
Server Hostname Property, B-8
Server Managers, 1-18
site.xconf, 1-3, 1-10
Special rules for mapping, C-4
SQL statement cache, 10-24
Starting Windchill, 1-7
Stopping Windchill, 1-7
Storage Administrator window, 3-8
System Administrator’s Guide
  about, xix
  audience, xix
  conventions, xxii
  organization, xx
  overview, xix
  related documentation, xix
System Configuration Recommendations, 10-23
System Configurator, 1-2, 1-6
System properties. See Property files

T
Technical support, xxi
Threshold detection, 1-37

U
Unrestricted Organizations group, 1-23
Updating
  content holder data formats, 7-5
  folders, 3-10
  hosts, 3-9
  mounts, 3-11
  revaulting schedule items, 3-20
URL generation, B-4
Usage Report utility, 1-26
User preferences. See Preferences
Users
  authenticating, 1-38–1-39, A-6
  preferences, 1-29

V
Vault Configuration window, 3-8
Vaulting Rule window, 3-15
Vaulting rules. See File vaults, rules for vaults, 10-23
Vaults. See File vaults
Viewing
  revaulting results, 3-18
  revaulting schedule items, 3-20
  sites in tree view, 3-4

W
Web browsers, A-4
WebEx, 1-19
Windchill adapter, 1-37
Windchill clients
  components of, A-4
Windchill environment
  configuring, 1-35
Windchill home page
  Meeting Center, 1-19
  overview, 1-18
Windchill Server page, 1-7
Windchill servers
  components of, 1-35, A-6
Windchill Service Pack, 1-41
Windchill Visualization Service and Content replication, 4-19
wt.fv.replicationFileSizeThreshold, 4-20
wt.org.InternalOrganization, 1-25
wt.org.OrganizationOwned.displayOrganization, 1-25
wt.prefs.delegates.DefaultSystemDelegate, 1-30
wt.prefs.delegates.DelegateOrder, 1-31
wt.prefs.delegates.UserDelegate, 1-30
wt.prefs.WTPreferences class, 1-30
wt.properties file
  content replication, 4-17
  file vaulting, 3-6
  import, 6-22
  indexing, 9-11, 9-13
  log files, 1-34
  modifying, 1-7
  preference delegates hierarchy, 1-31
  primary, 1-2
  queues, 8-4
  RetrievalWare, 9-15
wt.queue.queueGroup, 8-2, 8-5
WTOrganization, 1-21
wtSearch.xml file, 1-25

X
XCONF files, 1-3
xconf.dtd, 1-4
xconffmanager, 1-2, 1-10
XSL transformation, C-11