



# **Windchill® System Administrator's Guide**

**Windchill 8.0**

**April 2006**

**Copyright © 2005 Parametric Technology Corporation. All Rights Reserved.**

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

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

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

**Registered Trademarks of Parametric Technology Corporation or a Subsidiary**

Advanced Surface Design, Behavioral Modeling, CADDs, Computervision, CounterPart, Create Collaborate Control, EPD, EPD.Connect, Expert Machinist, Flexible Engineering, GRANITE, HARNESSDESIGN, Info\*Engine, InPart, MECHANICA, Optegra, Parametric Technology, Parametric Technology Corporation, PartSpeak, PHOTORENDER, Pro/DESKTOP, Pro/E, Pro/ENGINEER, Pro/HELP, Pro/INTRALINK, Pro/MECHANICA, Pro/TOOLKIT, Product First, Product Development Means Business, Product Makes the Company, PTC, the PTC logo, PT/Products, Shaping Innovation, The Way to Product First, and Windchill.

**Trademarks of Parametric Technology Corporation or a Subsidiary**

3DPAINT, Associative Topology Bus, AutobuildZ, CDRS, CV, CVact, CVaec, CVdesign, CV DORS, CVMAC, CVNC, CVToolmaker, EDAcompare, EDAconduit, DataDoctor, DesignSuite, DIMENSION III, Distributed Services Manager, DIVISION, e/ENGINEER, eNC Explorer, Expert Framework, Expert MoldBase, Expert Toolmaker, Harmony, InterComm, InterComm Expert, InterComm EDAcompare, InterComm EDAconduit, ISSM, KDiP, Knowledge Discipline in Practice, Knowledge System Driver, ModelCHECK, MoldShop, NC Builder, Pro/ANIMATE, Pro/ASSEMBLY, Pro/CABLING, Pro/CASTING, Pro/CDT, Pro/CMM, Pro/COLLABORATE, Pro/COMPOSITE, Pro/CONCEPT, Pro/CONVERT, Pro/DATA for PDGS, Pro/DESIGNER, Pro/DETAIL, Pro/DIAGRAM,

Pro/DIEFACE, Pro/DRAW, Pro/ECAD, Pro/ENGINE, Pro/FEATURE, Pro/FEM POST, Pro/FICIENCY, Pro/FLY THROUGH, Pro/HARNESS, Pro/INTERFACE, Pro/LANGUAGE, Pro/LEGACY, Pro/LIBRARYACCESS, Pro/MESH, Pro/Model.View, Pro/MOLDESIGN, Pro/NC ADVANCED, Pro/NC CHECK, Pro/NC MILL, Pro/NC POST, Pro/NC SHEETMETAL, Pro/NC TURN, Pro/NC WEDM, Pro/NC Wire EDM, Pro/NETWORK ANIMATOR, Pro/NOTEBOOK, Pro/PDM, Pro/PHOTORENDER, Pro/PIPING, Pro/PLASTIC ADVISOR, Pro/PLOT, Pro/POWER DESIGN, Pro/PROCESS, Pro/REPORT, Pro/REVIEW, Pro/SCAN TOOLS, Pro/SHEETMETAL, Pro/SURFACE, Pro/VERIFY, Pro/Web.Link, Pro/Web.Publish, Pro/WELDING, ProductView, PTC Precision, Routed Systems Designer, Shrinkwrap, Simple Powerful Connected, The Product Development Company, Wildfire, Windchill DynamicDesignLink, Windchill PartsLink, Windchill PDMLink, Windchill ProjectLink, and Windchill SupplyLink.

### **Patents of Parametric Technology Corporation or a Subsidiary**

Registration numbers and issue dates follow. Additionally, equivalent patents may be issued or pending outside of the United States. Contact PTC for further information.

GB2363208	25-August-2004	6,545,671 B1	08-April-2003	5,423,023	05-June-1990
GB2365567	10-March-2004	GB2354685B	18-June-2003	4,310,615	21-December-1998
6,665,569 B1	16-December-2003	GB2354683B	04-June-2003	4,310,614	30-April-1996
GB2353115	10-December-2003	6,608,623 B1	19-August-2003	4,310,614	22-April-1999
6,625,607 B1	23-September-2003	6,473,673 B1	29-October-2002	5,297,053	22-March-1994
6,580,428 B1	17-June-2003	GB2354683B	04-June-2003	5,513,316	30-April-1996
GB2354684B	02-July-2003	6,447,223 B1	10-Sept-2002	5,689,711	18-November-1997
GB2384125	15-October-2003	6,308,144	23-October-2001	5,506,950	09-April-1996
GB2354096	12-November-2003	5,680,523	21-October-1997	5,428,772	27-June-1995
GB2354924	24-September-2003	5,838,331	17-November-1998	5,850,535	15-December-1998
6,608,623 B1	19-August-2003	4,956,771	11-September-1990	5,557,176	09-November-1996
GB2353376	05-November-2003	5,058,000	15-October-1991	5,561,747	01-October-1996
GB2354686	15-October-2003	5,140,321	18-August-1992		

### **Third-Party Trademarks**

Adobe, Acrobat, Distiller, and the Acrobat logo are trademarks of Adobe Systems Incorporated. Advanced ClusterProven, ClusterProven, and the ClusterProven design are trademarks or registered trademarks of International Business Machines Corporation in the United States and other countries and are used under license. IBM Corporation does not warrant and is not responsible for the operation of this software product. AIX is a registered trademark of IBM Corporation. Allegro, Cadence, and Concept are registered trademarks of Cadence Design Systems, Inc. Apple, Mac, Mac OS, and Panther are trademarks or registered trademarks of Apple Computer, Inc. AutoCAD and Autodesk Inventor are registered trademarks of Autodesk, Inc. Baan is a registered trademark of Baan Company. CADAM and CATIA are registered trademarks of Dassault Systemes. COACH is a trademark of CADTRAIN, Inc. DOORS is a registered trademark of Telelogic AB. FLEXlm is a trademark of Macrovision Corporation. Geomagic is a registered trademark of Raindrop Geomagic, Inc. EVERSUNC, GROOVE, GROOVEFEST, GROOVE.NET, GROOVE NETWORKS, iGROOVE, PEERWARE, and the interlocking circles logo are trademarks of Groove Networks, Inc. Helix is a trademark of Microcadam, Inc. HOOPS is a trademark of Tech Soft America, Inc. HP-UX is a registered trademark of Hewlett-Packard Company. I-DEAS, Metaphase, Parasolid, SHERPA, Solid Edge, and Unigraphics are trademarks or registered trademarks of UGS Corp. InstallShield is a registered trademark and service mark of InstallShield Software Corporation in the United States and/or other countries. Intel is a registered trademark of Intel Corporation. IRIX is a registered trademark of Silicon Graphics, Inc. LINUX is a registered trademark of Linus Torvalds. MainWin and Mainsoft are trademarks of Mainsoft Corporation. MatrixOne is a trademark of MatrixOne, Inc. Mentor

Graphics and Board Station are registered trademarks and 3D Design, AMPLE, and Design Manager are trademarks of Mentor Graphics Corporation. MEDUSA and STHENO are trademarks of CAD Schroer GmbH. Microsoft, Microsoft Project, Windows, the Windows logo, Windows NT, Visual Basic, and the Visual Basic logo are registered trademarks of Microsoft Corporation in the United States and/or other countries. Netscape and the Netscape N and Ship's Wheel logos are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Oracle is a registered trademark of Oracle Corporation. OrbixWeb is a registered trademark of IONA Technologies PLC. PDGS is a registered trademark of Ford Motor Company. RAND is a trademark of RAND Worldwide. Rational Rose is a registered trademark of Rational Software Corporation. RetrievalWare is a registered trademark of Convera Corporation. RosettaNet is a trademark and Partner Interface Process and PIP are registered trademarks of RosettaNet, a nonprofit organization. SAP and R/3 are registered trademarks of SAP AG Germany. SolidWorks is a registered trademark of SolidWorks Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Sun, Sun Microsystems, the Sun logo, Solaris, UltraSPARC, Java and all Java based marks, and "The Network is the Computer" are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries. TIBCO, TIBCO Software, TIBCO ActiveEnterprise, TIBCO Designer, TIBCO Enterprise for JMS, TIBCO Rendezvous, TIBCO TurboXML, TIBCO BusinessWorks are the trademarks or registered trademarks of TIBCO Software Inc. in the United States and other countries. WebEx is a trademark of WebEx Communications, Inc.

### **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. 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. MainWin Dedicated Libraries are copyrighted software of Mainsoft 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. Parasolid © UGS Corp. 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. TIBCO ActiveEnterprise, TIBCO Designer, TIBCO Enterprise for JMS, TIBCO Rendezvous, TIBCO TurboXML, TIBCO BusinessWorks are provided by TIBCO Software Inc. Technology "Powered by Groove" is provided by Groove Networks, Inc. Technology "Powered by WebEx" is provided by WebEx Communications, Inc. Oracle 8i run-time and Oracle 9i run-time, Copyright © 2002-2003 Oracle Corporation. Oracle programs provided herein are subject to a restricted use license and can only be used in conjunction with the PTC software they are provided with. Apache Server, Tomcat, Xalan, Xerces, and Jakarta 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>. Adobe Acrobat Reader and Adobe Distiller are copyrighted software of Adobe Systems Inc. and are subject to the Adobe End-User License Agreement as provided by Adobe with those products. 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.infozip.org/pub/infozip/license.html>. 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](mailto:leo@mud.de) or [marcus@mud.de](mailto: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 is licensed under the GNU Library General Public License (LGPL). You may obtain a copy of the source code at <http://www.gtk.org>, which is likewise provided under the GNU LGPL. zlib software Copyright © 1995-2002 Jean-loup Gailly and Mark Adler. OmniORB is distributed under the terms and conditions of the GNU General Public License and GNU Library General Public License. The Java Getopt.jar file, copyright 1987 1997 Free Software Foundation, Inc.; Java Port copyright 1998 by Aaron M. Renn ([arenn@urbanophile.com](mailto:arenn@urbanophile.com)), is redistributed under the GNU LGPL. You may obtain a copy of the source code at <http://www.urbanophile.com/arenn/hacking/download.html>. The source code is likewise provided under the GNU LGPL. This product may include software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org>): Copyright © 1998 2003 The OpenSSL Project. All rights reserved. This product may include cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). Gecko and Mozilla components are subject to the Mozilla Public License Version 1.1 at <http://www.mozilla.org/MPL>. Software distributed under the Mozilla Public License (MPL) is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either expressed or implied. See the MPL for the specific language governing rights and limitations. Mozilla Japanese localization components are subject to the Netscape Public License Version 1.1 (at <http://www.mozilla.org/NPL>). Software distributed under the Netscape Public License (NPL) is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either expressed or implied (see the NPL for the rights and limitations that are governing different languages). The Original Code is Mozilla Communicator client code, released March 31, 1998 and the Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright © 1998 Netscape Communications Corporation. All Rights Reserved. Contributors: Kazu Yamamoto ([kazu@mozilla.gr.jp](mailto:kazu@mozilla.gr.jp)), Ryoichi Furukawa ([furu@mozilla.gr.jp](mailto:furu@mozilla.gr.jp)), Tsukasa Maruyama ([mal@mozilla.gr.jp](mailto:mal@mozilla.gr.jp)), Teiji Matsuba ([matsuba@dream.com](mailto:matsuba@dream.com)).

#### UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227 7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227 19(c)(1)-(2) (JUN'87), as applicable. 010505

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



# Contents

<b>Change Record .....</b>	<b>xvii</b>
<b>About This Guide.....</b>	<b>xxix</b>
Intended Audience.....	xxix
Overview.....	xxix
Chapter Contents .....	xxx
Related Documentation .....	xxxix
Technical Support.....	xxxix
Documentation for PTC Products.....	xxxix
Comments .....	xxxix
Documentation Conventions .....	xxxix
Third-Party Products .....	xxxix
Code Examples .....	xxxix
<b>Administering Runtime Services .....</b>	<b>1-1</b>
Windchill Configuration Properties .....	1-2
The site.xconf File Format and Contents .....	1-3
The xconf.dtd File .....	1-4
Using the System Configurator.....	1-7
Starting and Stopping the Windchill System.....	1-7
Modifying System Properties .....	1-8
Working with Log Files.....	1-9
Using the xconfmanager Utility.....	1-11
xconfmanager Command Syntax .....	1-12
Viewing xconfmanager Help .....	1-16
Setting Property Values and Propagating Your Changes.....	1-16
Listing Property Information .....	1-19
Validating XCONF Files.....	1-20
Other xconfmanager Options.....	1-21
The Windchill Foundation & PDM Home Page.....	1-22
Setting Up Meetings .....	1-23
Accessing Meetings.....	1-23
Configuring E-mail Notifications for Your Site.....	1-24
WebEx Meeting Center Setup .....	1-24

WebEx User Name and E-mail Address .....	1-26
Proxy Server for Connection to WebEx Meeting Center .....	1-26
Administering Organizations .....	1-27
Restricted Organizations .....	1-28
Internal Organizations .....	1-29
Administering Desktop Integration .....	1-32
Configuring Desktop Integration When Enterprise Search is not Used.....	1-32
Setting Desktop Integration Preferences.....	1-33
Setting Up Internet Explorer for Downloads on Windows XP Clients .....	1-34
Running the Windchill ProjectLink Usage Report Utility .....	1-34
Creating Usage Reports.....	1-34
Sample HTML Report.....	1-36
Administering User Preferences .....	1-37
Managing User Preferences.....	1-37
Defining Preference Scopes.....	1-39
Ensuring Proper Backup and Recovery .....	1-42
Maintaining Log Files .....	1-43
Configuring Your Windchill Environment .....	1-43
Configuring a Single Method Server .....	1-44
Configuring a Method Server for Background Queues .....	1-44
Load Balancing for Multiple Method Servers.....	1-45
Configuring Log Files for Multiple Servers .....	1-46
Changing Authentication Text Between Servlet and Windchill Adapter.....	1-47
Administering the Authentication Process .....	1-48
Troubleshooting User Authentication .....	1-49
Double Authentication Can Be Eliminated when Using Internet Explorer .....	1-49
Authentication Configuration Tools .....	1-49
Windchill Scheduler .....	1-50
Windchill Scheduler Automatic Removal of History Items.....	1-50
Other Properties .....	1-51
Purging, Archiving, and Restoring Windchill Information .....	1-51
Windchill Software Maintenance and Best Practices .....	1-53
Installation Process for the Windchill Service Pack.....	1-54
Best Practices for Managing Windchill Installation and Maintenance .....	1-56
About the windchill Command .....	1-57
About the windchill shell.....	1-59
.....	1-60
<b>Administering the Bootstrap Client and JAR Files.....</b>	<b>2-1</b>
Overview .....	2-2



About the Bootstrap Feature .....	2-2
Java Plug-in Cache .....	2-3
Bootstrap Configuration File .....	2-3
Specifying JAR File Cache Location.....	2-3
Controlling Java System Property Settings .....	2-6
Bootstrap Package Versioning .....	2-7
Downloading JAR Files .....	2-8
Administering Codebases.....	2-8
Updating Client JAR Files .....	2-9
JAR Files and Security .....	2-9
Determining Client JAR Contents .....	2-10
<b>Administering External File Vaults .....</b>	<b>3-1</b>
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
Specifying the File Threshold Value .....	3-8
Out-of-the-box Background Queues for External Vaulting .....	3-9
Windchill External Storage Administrator .....	3-10
Adding and Updating Hosts .....	3-11
Adding and Updating Vaults .....	3-12
Deleting a Vault .....	3-12
Adding and Updating Folders .....	3-12
Deleting a Folder .....	3-13
Creating and Updating Mounts .....	3-13
Maintaining Your Vaults .....	3-14
Creating a Vaulting Policy .....	3-15
Examining Existing Rules .....	3-15
Creating New Rules.....	3-16
Modifying and Deleting Rules .....	3-18
Managing Revaulting.....	3-18
Examining Existing Revaulting Schedule Items.....	3-19
Viewing the Results of Revaulting .....	3-20
Creating a Revaulting Schedule Item .....	3-20
Updating a Schedule Item .....	3-22
Viewing a Schedule Item .....	3-22
Canceling a Schedule Item.....	3-22
Deleting a Schedule Item.....	3-22
Forcing Content to Vault .....	3-22

Changing the Location of Files in External Vaults .....	3-24
Moving a Server Using External Vaulting to a Different Host .....	3-25
<b>Administering Content Replication.....</b>	<b>4-1</b>
Overview .....	4-2
Two Types of Content Replication Sites .....	4-4
Modifying the Local Site .....	4-5
Replication Security .....	4-6
Importing Certificates into Sites.....	4-6
Editing Content Replication Properties .....	4-7
Master and Replica Properties .....	4-7
Basic Properties .....	4-9
Out-of-the-box Background Queues for Content Replication .....	4-11
Maximum Queue Values .....	4-11
Content Replication and Windchill Visualization Service .....	4-13
Configuring Properties.....	4-13
To Replicate WVS Viewables.....	4-13
Replication and Compression .....	4-14
Troubleshooting Replication.....	4-15
Improving Content Replication Performance .....	4-15
How the Local Cache Works .....	4-16
Administering a Local Cache Vault .....	4-18
Establishing Mirroring in the Local Cache Vault.....	4-18
Setting the Preferred Content Cache Site .....	4-19
Scheduling Moving Data from Local Cache to the Master Site .....	4-19
Removing Unreferenced Files from a Cache Vault on a Replica Site.....	4-19
Utility to Assist Backups .....	4-23
Log Files.....	4-23
<b>Configuring External File Vaulting or Replication With FvLoader .....</b>	<b>5-1</b>
Overview of Configuration With FvLoader .....	5-2
Configuring External File Vaults or Rules .....	5-3
Supporting Replica Site Vaulting .....	5-3
Removing External File Vaulting Rules or Replication Rules .....	5-4
Listing Domains .....	5-5
The -listContainers Argument Obtains Data.....	5-5
The -listDomains Argument Presents Data .....	5-6
Listing Vaulting Rules .....	5-7
<b>Windchill Import and Export.....</b>	<b>6-1</b>
Overview .....	6-2

Context Considerations in Import and Export.....	6-4
Export Container Availability .....	6-4
Import Container Availability .....	6-5
Controlling the Destinations of Imported Items with Context Mapping Files.....	6-6
Importing and Exporting: Supported Items List .....	6-8
Supported Windchill ProjectLink Items .....	6-9
Supported Windchill PDMLink Items .....	6-13
Supported Windchill Foundation & PDM Items.....	6-29
Configuration Specification Settings.....	6-35
Import and Export of EPMDocuments .....	6-36
Attribute Limitations .....	6-36
Exporting and Importing EPMDocuments as Checked Out.....	6-37
Exporting with the Export User Interface .....	6-37
Exporting from the Export Window .....	6-37
Exporting an Item from its Properties Page .....	6-40
Importing with the Import User Interface .....	6-42
Additional Export and Import Actions .....	6-46
Additional Lock Export Action .....	6-46
Additional Import Actions .....	6-46
Windchill Properties for Export and Import.....	6-47
Windchill Export Properties .....	6-47
Windchill Import Properties.....	6-48
Access Control for Export and Import.....	6-49
Export Access Control Rules .....	6-49
Import Access Control Rules .....	6-52
<b>Administering Content Holders and Content Objects .....</b>	<b>7-1</b>
Overview of Content Holders.....	7-2
Content Handling Configuration .....	7-3
Adding and Updating Data Formats .....	7-4
<b>Configuring and Administering Background Queues .....</b>	<b>8-1</b>
Overview.....	8-2
Queue Entry States .....	8-2
Out-of-the-box Background Queues.....	8-3
CleanUpScheduleQueue .....	8-3
CtScheduleQueue .....	8-4
DataSharingQueue .....	8-4
EMailQueue .....	8-4
forumEventPropogationQueue .....	8-4
Indexing Queue and Bulk Indexing Queue .....	8-4

MarkForDeleteQueue.....	8-5
NotificationQueue.....	8-5
NotificationScheduleQueue.....	8-5
PagingScheduleQueue.....	8-5
PartsLink Queue.....	8-6
ProjectScheduleQueue.....	8-6
PublisherQueue and PublisherQueue1.....	8-6
PurgeHTMLLogMessageQueue.....	8-7
PurgeOrphanedEffAuditsQueue.....	8-7
StatisticsScheduleQueue.....	8-7
WfPropagationQueue.....	8-8
WfScheduleQueue.....	8-8
WfUserWorkQueue.....	8-9
Configuring Background Queues and Related Properties.....	8-9
Configuring Multiple Background MethodServers.....	8-9
Background Queue Properties.....	8-10
Background Queue Log Properties.....	8-12
Other Background Queue-Specific Properties.....	8-13
Regular Queue Maintenance.....	8-14
Maintaining Queues.....	8-14
<b>Customizing and Administering Pro/ENGINEER Wildfire.....</b>	<b>9-1</b>
Customizing Pro/ENGINEER Wildfire.....	9-2
Environment Variables and Config.pro Options for Pro/ENGINEER Wildfire.....	9-2
INI Files.....	9-9
Managing CAD Document and WTPart Naming and Numbering.....	9-12
Managing Incomplete Items.....	9-19
Soft Typing CAD Documents.....	9-20
Mapping Pro/ENGINEER Parameters to Windchill Instance-Based Attributes.....	9-21
Customizing the Parameters in the Download Service.....	9-23
Configuring the Build Rule.....	9-25
Managing ModelCHECK Validation during Check In.....	9-27
Configuring the Initial Collection of Items for Actions.....	9-29
Enabling Support for Custom Parts.....	9-29
Administering Revision.....	9-31
Enabling Display of Rename History and Location History.....	9-34
Customizing AutoAssociate.....	9-34
Customizing the HTML Client Item Selection Page.....	9-37
Defining the Rename Report Mail Server.....	9-43
Generation of Viewables.....	9-44

Managing Secondary Content .....	9-44
Managing Drawing Dependents .....	9-45
Controlling the Display of Internal Pro/ENGINEER Relationships .....	9-45
Clean-up of the Event Manager .....	9-46
Administering Table Views .....	9-47
Configuring Table Scrollbar Display .....	9-47
Configuring the Number of Workspace Rows Displayed .....	9-47
Configuring Automatic Scrolling in the Workspace .....	9-47
System Configuration Recommendations .....	9-48
Running Multiple Servers .....	9-48
Using External File Vaulting .....	9-48
Using Content Replication .....	9-48
Performance Tuning .....	9-48
Setting the Method Server Max Heap Size .....	9-48
Data Compression .....	9-49
Maximizing the Oracle Server/Windchill Method Server Connection .....	9-50
Choosing to Display Family Item Symbols in Folders Table .....	9-50
Other Recommendations .....	9-51
Online Java Performance Guide .....	9-51
Windchill Folder Structure .....	9-51
HTTP Protocol .....	9-51
Default Directory INI Files .....	9-52
Cadhtmlui.ini File .....	9-52
Contentcat.ini File .....	9-55
Newdocument.ini File: .....	9-58
Site Directory INI Files .....	9-59
Autoassociate.ini File .....	9-59
Newdocument.ini File .....	9-60
<b>Windchill Runtime Environment .....</b>	<b>A-1</b>
Web Infrastructure .....	A-2
Java Platform Support .....	A-2
Three-Tier Architecture .....	A-3
Client Software Components .....	A-4
Web Browser .....	A-4
HTML Pages .....	A-5
Java Applets .....	A-5
Server Software Components .....	A-6
HTTP Server .....	A-6
HTTP Gateway .....	A-7

Server Manager.....	A-11
Method Server.....	A-15
Log Files.....	A-18
Database Components .....	A-18
Object Relational Database Management System (ORDBMS) .....	A-18
Full Text Retrieval Indexing Components .....	A-21
File Content Indexing .....	A-21
Publishing.....	A-22
Indexable Objects.....	A-22
Indexing Policies .....	A-22
IndexDelegate Objects .....	A-23
Index Loader .....	A-23
<b>Windchill Considerations for Security Infrastructures .....</b>	<b>B-1</b>
Overview .....	B-2
Protocols .....	B-2
Authentication .....	B-3
URL Generation .....	B-4
Server Codebase Property.....	B-5
Relative and Absolute URLs .....	B-5
Choosing Host Names .....	B-6
RMI .....	B-7
Server hostname Property .....	B-7
Configuration Properties.....	B-7
RMI-over-HTTP .....	B-8
Port 80.....	B-9
Java RMI Servlet.....	B-10
Using HTTPS Protocol .....	B-10
Configuration Considerations.....	B-10
Firewalls .....	B-10
Client-Side Proxy Servers .....	B-11
Server-Side Reverse Proxy Servers .....	B-11
<b>Import and Export Policies, Mapping Rules, and Conflict Messages.....</b>	<b>C-1</b>
XSL-based Policy Files .....	C-2
Policy File Example .....	C-2
Mapping Rules .....	C-4
Mapping Through Special Rules .....	C-4
Mapping Priorities.....	C-5
Client-based Mapping Rules Files.....	C-5
Client-based Mapping Rules File Example .....	C-6

Generalized Mapping Rules File Example.....	C-7
Properties in Mapping Rules Files .....	C-8
Do Not Map Number Attributes for MCAD Documents.....	C-8
About Mapping Rules .....	C-8
COPY Element .....	C-8
COPY_AS Element.....	C-9
IGNORE Element .....	C-10
IGNORE_PARENT Element.....	C-11
Mapping Through XSL Transformation .....	C-11
Java Mapping with the METHOD Element .....	C-12
Hierarchical Instance Based Attribute Definitions, Exporting, and Importing .....	C-12
When to Use Mapping Files for Hierarchical IBAs .....	C-12
How to Write a Mapping File for Hierarchical IBAs .....	C-13
Conflict Messages .....	C-16
Importing RatioDefinition and RatioValue .....	C-16
Administrative Conflicts of Common Business Objects .....	C-17
Administrative Conflicts of Administrative Objects.....	C-19
Dependency Conflicts .....	C-21
Metadata Conflicts .....	C-22
Reforming Custom Modeled Attributes.....	C-24
Example of Two Formats of Mapping Files .....	C-25
Ignoring an Attribute .....	C-26
Changing a Tag to a Different Name .....	C-27
Administrative Conflicts of Common Administrative Objects .....	C-28
<b>Customizing Online Help .....</b>	<b>D-1</b>
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
<b>Customizing Online Tutorials.....</b>	<b>E-1</b>
Online Tutorial Infrastructure .....	E-2
Downloading the Tutorials .....	E-2
Adding a New Tutorial .....	E-3

Creating Tutorials .....	E-4
Adding a New Tutorial to the Tutorial Menu Presented in a Windchill Solution .....	E-4
Regenerating the Index Using the TutorialBuilder Java Class .....	E-7
Linking to a New Tutorial through the User Interface of a Windchill Solution.....	E-7
Editing Existing Tutorials.....	E-8
Editing the Tutorial Index .....	E-9

## Index



# Change Record

The following tables list the major changes made in this guide for Windchill releases.

**Table 1 Changes for Windchill 8.0 M020**

Chapter	Description
Chapter 1, <a href="#">Administering Runtime Services</a>	Added <a href="#">Configuring E-mail Notifications for Your Site</a> section.
Chapter 3, <a href="#">Administering External File Vaults</a>	Added <a href="#">Specifying the File Threshold Value</a> section.  Removed the procedure, Changing from a Multiple Vault to a Single Vault Architecture.
Chapter 4, <a href="#">Administering Content Replication</a>	Added descriptions for the following content replication properties: <ul style="list-style-type: none"><li>• wt.fv.revaultQuerySize</li><li>• wt.fv.master.millisecsToWait</li><li>• wt.fv.master.siteConfigDelivery Attempts</li></ul> Added <a href="#">Resetting Replication</a> section. Updated <a href="#">Resetting Replication for Undelivered Items</a> section.
Chapter 8, <a href="#">Configuring and Administering Background Queues</a>	Added description for <a href="#">CleanUpScheduleQueue</a> to the <a href="#">Out-of-the-box Background Queues</a> section.

Chapter	Description
Chapter 9, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	<p>Added the <a href="#">Administering Revision</a> section.</p> <p>Added description of dm_search_primary_server option to the <a href="#">Config.pro Options</a> section.</p>

**Table 2 Changes for Windchill 8.0 M010**

Chapter	Description
Chapter 1, <a href="#">Administering Runtime Services</a>	Updated this section for 8.0 M010.
Chapter 9, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Updated this section for 8.0 M010.

**Table 3 Changes for 8.0**

Chapter	Description
Administering RetrievalWare Libraries chapter	Removed from this guide and added to the <i>Windchill Installation and Configuration Guide -RetrievalWare</i> .
Chapter 1, <a href="#">Administering Runtime Services</a>	<p>In the <a href="#">Using the xconfmanager Utility</a> section, documented additional xconfmanager parameters for adding and removing values from multi-valued properties and for validating XCONF files.</p> <p>Added the <a href="#">Setting Desktop Integration Preferences</a> section</p> <p>Added examples of site preferences that you can set to the <a href="#">Managing User Preferences</a> section.</p> <p>Added the <a href="#">Setting Up Internet Explorer for Downloads on Windows XP Clients</a> section.</p> <p>Updated the <a href="#">Windchill Software Maintenance and Best Practices</a> section to reference the <i>Windchill Customizer's Guide</i> for specific information about making changes to Windchill code.</p>

Chapter	Description
Chapter 2, <a href="#">Administering the Bootstrap Client and JAR Files</a>	Added <a href="#">Determining Client JAR Contents</a> section.
Chapter 3, <a href="#">Administering External File Vaults</a>	Added warning for creating and mounting folders in the <a href="#">Creating and Updating Mounts</a> section.
Chapter 4, <a href="#">Administering Content Replication</a>	<p>Removed installation and configuration information. This information is now available in the <i>Windchill Installation and Configuration Guide - Windchill</i>.</p> <p>Added procedures for removing unreferenced files in the <a href="#">Removing Unreferenced Files from a Cache Vault on a Replica Site</a> section.</p>
Chapter 5, <a href="#">Configuring External File Vaulting or Replication With FvLoader</a>	Added the VE,vaultName argument in the <a href="#">Configuring External File Vaults or Rules</a> section.
Chapter 6, <a href="#">Windchill Import and Export</a>	Revised the <a href="#">Controlling the Destinations of Imported Items with Context Mapping Files</a> section to improve readability and provide specific information on context mapping files.
Chapter 7, <a href="#">Administering Content Holders and Content Objects</a>	Moved information contained in the <a href="#">Adding and Updating Data Formats</a> section to the <i>Windchill Customizer's Guide</i> .
Chapter 8, <a href="#">Configuring and Administering Background Queues</a>	Added new queues: CtScheduleQueue, Indexing Queue, Bulk Indexing Queue, and PartsLink Queue to the <a href="#">Out-of-the-box Background Queues</a> section.

Chapter	Description
Chapter 9, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	<ul style="list-style-type: none"> <li>Removed the section, Customizing the Naming Service, now that all properties in the section <a href="#">Managing CAD Document and WTPart Naming and Numbering</a> are implemented.</li> <li>Removed the section, Customizing the Event Console -- obsolete due to the new Event Manager functionality.</li> </ul>

Chapter	Description
<p>Chapter 9, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a> (continued)</p>	<ul style="list-style-type: none"> <li>Removed the section, Customizing the Latest Configuration Specification for Checkout -- obsolete due to new Configuration tools.</li> <li>Added a new configuration option (open_simplified_rep_by_default ) to the table in the section <a href="#">Config.pro Options</a>.</li> <li>Added a new section: <a href="#">Configuring the Build Rule</a>.</li> <li>Added a new section: <a href="#">Configuring the Initial Collection of Items for Actions</a>.</li> <li>Added a new section: <a href="#">Enabling Support for Custom Parts</a>.</li> <li>Expanded section, <a href="#">Enabling Support for Custom Parts</a>.</li> <li>Added a new section: <a href="#">Enabling Display of Rename History and Location History</a>.</li> <li>Added a new section: <a href="#">Managing Secondary Content</a>.</li> <li>Added a new section: <a href="#">Managing Drawing Dependents</a>.</li> <li>Added a new section: <a href="#">Clean-up of the Event Manager</a>.</li> <li>Added a new section: <a href="#">Administering Table Views</a>.</li> <li>Added a new section: <a href="#">Configuring Table Scrollbar Display</a>.</li> <li>Added a new section: <a href="#">Configuring the Number of Workspace Rows Displayed</a>.</li> </ul>

Chapter	Description
Chapter 9, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a> (continued)	<ul style="list-style-type: none"> <li>Added a new section: <a href="#">Configuring Automatic Scrolling in the Workspace</a>.</li> <li>Added note in the section, <a href="#">Using External File Vaulting</a>, clarifying when checked in items are visible in the external vault.</li> <li>Updated the section, <a href="#">Data Compression</a>, adding information on Windchill settings.</li> <li>Updated the section <a href="#">Default Directory INI Files</a> with new file: contentcat.ini, and added a new [dependency tracing] section in the file cadxhtmlui.ini.</li> <li>Updated the section <a href="#">Site Directory INI Files</a> with new file: saveas.ini</li> </ul>
Appendix A, <a href="#">Windchill Runtime Environment</a>	Updated <a href="#">Full Text Retrieval Indexing Components</a> section.
Appendix B, <a href="#">Windchill Considerations for Security Infrastructures</a>	Updated <a href="#">Server-Side Reverse Proxy Servers</a> section.
Appendix E, <a href="#">Customizing Online Tutorials</a>	Updated chapter to document new approach for tutorials.

**Table 4 Changes for Release 7.0 M030**

Change	Description
Chapter 4, <a href="#">Administering Content Replication</a>	Renamed and reorganized chapter to create two workflows: one for a lightweight replica server and one for a full-scale replica server.

Change	Description
Chapter 6, <a href="#">Windchill Import and Export</a>	In the <a href="#">Importing and Exporting: Supported Items List</a> section, added list of objects supported by the LoadfromFile framework and/or import/export functionality. Tables for Windchill ProjectLink, Windchill PDMLink, and Windchill Foundation &PDM are available.

**Table 5 Changes for Windchill 7.0 M020**

Change	Description
Chapter 1, <a href="#">Administering Runtime Services</a>	Replaced the WebEx Setup Properties section with an updated section describing how to set up Meeting Center at both the organization and site levels.
Chapter 1, <a href="#">Administering Runtime Services</a>	Updated the Organization ID Type information.
Chapter 1, <a href="#">Administering Runtime Services</a>	New section describing option to eliminate double authentication for Internet Explorer users.
Chapter 1, <a href="#">Administering Runtime Services</a>	Updated maintenance information in the section.
Chapter 2, <a href="#">Administering the Bootstrap Client and JAR Files</a>	Replaced MarkJar Script for Client JAR Files section.
Chapter 3, <a href="#">Administering External File Vaults</a>	Added information on vaulting queues.
Chapter 4, Installing a Lightweight Replica Site	Updated this procedure to reflect implementation of InstallAnywhere installer.
Chapter 4, <a href="#">Administering Content Replication</a>	Added information on content replication queues.
Chapter 4, Replica Configuration	Updated procedures for lightweight replica site to reflect Release 7.0 changes.

Change	Description
Chapter 4, <a href="#">Master and Replica Properties</a>	Added and defined a new property: wt.fv.master.moveItemsBetweenReplicaVaults.
Chapter 8, <a href="#">Configuring and Administering Background Queues</a>	New section describing out-of-the-box queues.
Chapter 8, <a href="#">Configuring and Administering Background Queues</a>	Documented a property for queue notification intervals.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added or updated the names and descriptions of the following config.pro options:  dm_cache_size  dm_secondary_upload  dm_auto_open_zip  dm_cache_mode  regenerate_read_only_objects  dm_overwrite_contents_on_update
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added section that includes INI settings for this action.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added a discussion of the naming and numbering policies, and INI settings.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added section that includes INI settings for this action.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added section that includes INI settings for handling incomplete objects upon check-in.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Added a section describing the property that can remove the display of certain Pro/ENGINEER relationships in the References/Referenced By pages.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Updated reference Cadxhtmlui.ini File section at end of chapter.



Change	Description
Appendix C, <a href="#">Import and Export Policies, Mapping Rules, and Conflict Messages</a>	Added description for potential conflict: Existing Soft Type belongs to a different Logical Identifier Group relative to XML.
Appendix C, <a href="#">Import and Export Policies, Mapping Rules, and Conflict Messages</a>	Added description for potential conflict: Existing Soft Type has different Logical Identifier relative to XML file.
Appendix D, <a href="#">Customizing Online Help</a>	Updated the path to installed WebHelp.
Appendix E, <a href="#">Customizing Online Tutorials</a>	Added this new appendix.

**Table 6 Changes for Windchill 7.0**

Change	Description
Chapter 1, <a href="#">Using the System Configurator</a>	Describes the redesigned interface of the System Configurator.
Chapter 1, <a href="#">Using the xconfmanager Utility</a>	Describes the new utility for editing Windchill property files.
Chapter 1, <a href="#">Setting Up Meetings</a>	Improved description of the setup needed for Meeting Center.
Chapter 1, <a href="#">Administering Organizations</a>	Describes the new features relating to organizations.
Chapter 1, <a href="#">Administering Desktop Integration</a>	Describes the new feature that allows you to create and edit Windchill objects in Microsoft Office applications.
Chapter 1, <a href="#">Running the Windchill ProjectLink Usage Report Utility</a>	Describes the utility that allows you to collect information about Windchill ProjectLink usage.
Chapter 1, <a href="#">Administering User Preferences</a>	Updated to describe changes relating to containers.
Chapter 1, <a href="#">Changing Authentication Text Between Servlet and Windchill Adapter</a>	Added to document how to change the secret text used in authenticating the servlet and Windchill adapter.

Change	Description
Chapter 1, <a href="#">Windchill Software Maintenance and Best Practices</a>	Introduces the service pack installation and best practices for maintenance.
Chapter 4, <a href="#">Vaults and Folders</a>	Presents complete documentation for installing a full-scale replica site and a lightweight replica site.
Chapter 4, <a href="#">Content Replication and Windchill Visualization Service</a>	Explains how to ensure the replication of viewables.
Chapter 4, <a href="#">Configuring Properties</a>	Explains wt.fv.replicationFileSizeThreshold property that controls minimum size of files replicated.
Chapter 4, <a href="#">Replication and Compression</a>	Explains compression in content replication.
Chapter 4, <a href="#">Improving Content Replication Performance</a>	The Local Content Cache and Content Cache Server are now explained in this chapter rather than a separate chapter.
Chapter 5, <a href="#">Configuring External File Vaulting or Replication With FvLoader</a>	Explains how to use the FvLoader utility to configure file vaulting and content replication. This is a new chapter.
Chapter 6, <a href="#">Controlling the Destinations of Imported Items with Context Mapping Files</a>	Explains context mapping file for use in import.
Chapter 6, <a href="#">Import and Export of EPMDocuments</a>	Discusses prerequisites for importing and exporting EPMDocuments.
Chapter 6, <a href="#">Exporting with the Export User Interface</a>	Explains the graphical user interface for Windchill Export .
Chapter 6, <a href="#">Importing with the Import User Interface</a>	Explains the graphical user interface for Windchill Import and the backward compatibility for RatioDefinition and RatioValue.
Chapter 6, <a href="#">Additional Export and Import Actions</a>	Explains Windchill Export and Import actions that do not appear in the graphical user interface.

Change	Description
Chapter 6, <a href="#">Access Control for Export and Import</a>	Suggests access control rules for Windchill Import and Export.
Chapter 8, <a href="#">Queue Entry States</a>	Adds Severe state.
Chapter 8, <a href="#">Background Queue Properties</a>	Describes updated properties.
Chapter 8, <a href="#">Maintaining Queues</a>	Describes updated procedure.
Chapter 9, About Indexing	Updated the example file shown in step 3.
Chapter 10, <a href="#">Customizing and Administering Pro/ENGINEER Wildfire</a>	Presents customization and administration information, and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill Foundation & PDM, Windchill PDMLink, and Windchill ProjectLink. This is a new chapter.
Appendix A, <a href="#">Windchill Runtime Environment</a>	Updates made for Windchill 7.0.
Appendix B, <a href="#">Windchill Considerations for Security Infrastructures</a>	Updates made for Windchill 7.0.
Appendix C, <a href="#">Import and Export Policies, Mapping Rules, and Conflict Messages</a>	Changes made to the appendix title.
Appendix C, <a href="#">Hierarchical Instance Based Attribute Definitions, Exporting, and Importing</a>	Explains preparation for importing hierarchical Instance Based Attribute definitions.
Appendix C, <a href="#">Conflict Messages</a>	Improves documentation of import conflicts.
Appendix D, <a href="#">Customizing Online Help</a>	Explains how to customize Windchill online help. This is a new appendix.



# 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:

	<b>System Administrator</b>
<b>Responsibilities</b>	Keeping the system running. Interfacing with other systems. Administering Windchill applications.
<b>Skills</b>	Understanding Windchill server and client, HTML, HTTP, and database.

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

Chapter 5, Configuring External File Vaulting or Replication With FvLoader, describes the FvLoader utility, which is a shortened version of File Vault Object Loader.

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, 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 9, Customizing and Administering Pro/ENGINEER Wildfire, presents customization and administration information and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill Foundation & 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 C, Import and Export Policies, Mapping Rules, and Conflict Messages, describes methods of mapping attributes during import and export.

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*
- *Windchill Foundation & PDM User's Guide*
- *Windchill PDMLink User's Guide*
- *Windchill ProjectLink User's Guide*
- *Windchill Archive Administrator's Guide*
- *Windchill Customizer's Guide*
- *Windchill Adapter Guide*
- *Windchill Performance Tuning Guide*
- properties.html file

Not all guides are available; only those guides for the products installed are available. If a guide you are interested in reading is not installed at your site, contact whoever did the installation. You can also access the guides from the Reference Documents section of the PTC Web site as described in [Documentation for PTC Products](#).

## 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>

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>

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 and copied to the application when it is installed. In addition, all books [including those 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. You can submit your feedback through the online survey form at the following URL:


[http://www.ptc.com/go/wc\\_pubs\\_feedback](http://www.ptc.com/go/wc_pubs_feedback)

## Documentation Conventions

Windchill documentation uses the following conventions:

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



Convention	Item	Example
	The <b>Caution</b> symbol indicates potentially unsafe situations which may result in minor injury, machine damage or downtime, or corruption or loss of software or data.	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.

## 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.



# 1

## Administering Runtime Services

This chapter provides system administration information related to Windchill runtime services. For a diagram of the complete Windchill directory structure, see the *Windchill Customizer's Guide*.

Topic	Page
Windchill Configuration Properties .....	1-2
Using the System Configurator .....	1-7
Using the xconfmanager Utility .....	1-11
The Windchill Foundation & PDM Home Page .....	1-22
Setting Up Meetings .....	1-23
Administering Organizations .....	1-27
Administering Desktop Integration .....	1-32
Setting Up Internet Explorer for Downloads on Windows XP Clients.....	1-34
Running the Windchill ProjectLink Usage Report Utility .....	1-34
Administering User Preferences.....	1-37
Ensuring Proper Backup and Recovery.....	1-42
Maintaining Log Files .....	1-43
Configuring Your Windchill Environment .....	1-43
Changing Authentication Text Between Servlet and Windchill Adapter .....	1-47
Administering the Authentication Process .....	1-48
Troubleshooting User Authentication .....	1-49
Windchill Scheduler .....	1-50
Windchill Software Maintenance and Best Practices.....	1-53
About the windchill Command .....	1-57
About the windchill shell .....	1-59

# Windchill Configuration Properties

Windchill uses standard Java property files to store many of its settings. The majority of the settings are stored in the property file `wt.properties`, which is located in the Windchill codebase directory. From this location, `wt.properties` is available to be downloaded by clients.

Although `wt.properties` holds most settings, there are other property files located under the codebase directory. These property files are also available to clients. Some property files are stored outside of codebase because they are not needed by clients or should not be available to clients. For example, the `db.properties` (which contains a password to your database instance) is located outside the codebase in the Windchill db directory.

To manage site property settings, PTC no longer recommends that you use a text editor to edit individual property files. Instead, all site-specific 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 System Configurator provides a user interface for updating properties in the most common set of property files.

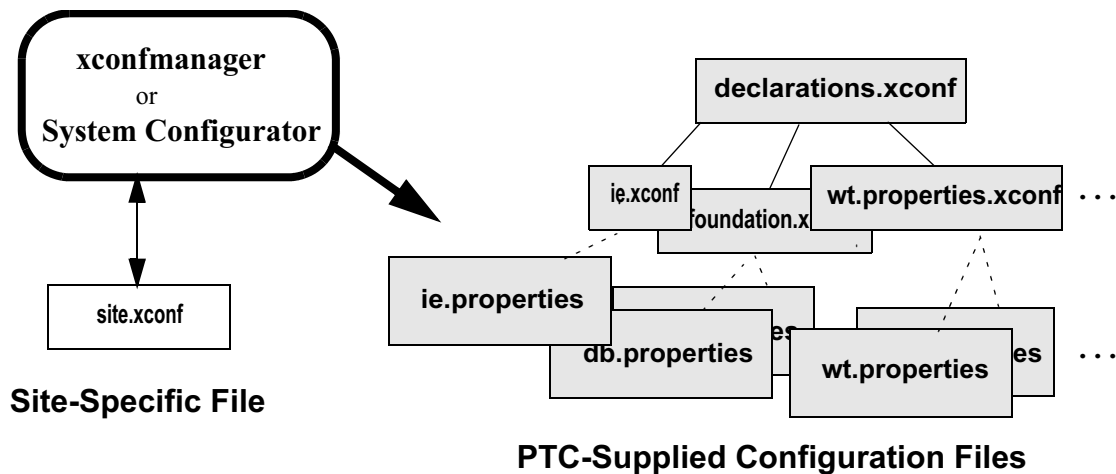
The changes made through either of these utilities are saved in the `site.xconf` file and propagated to respective property files. When you restart your Windchill system, the resulting changes are implemented.

The `site.xconf` file is created and updated by Windchill product installers. These installers also create 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 will be retained automatically when applying PTC-supplied maintenance updates to your installation.

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. Explicitly propagating changes allows you to make multiple command line changes before propagating all 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 AddToProperty element names a property value to add to the end of a multi-valued property. The xconfmanager adds this element to the site.xconf file when you add a property value to the end of a multi-valued property.
- Each RemoveFromProperty element names a property value to remove from a multi-valued property. The xconfmanager adds this element to the site.xconf file when you remove a property value from a multi-valued property.
- 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 choose 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.

**Tip:** If you are attempting to batch script calls to the xconfmanager.bat script using the DOS scripting language, be sure to use call script.bat (and not script.bat).

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 place 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:

```
<!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|AddToProperty|RemoveFromProperty|Service|Resource|ConfigurationRef|ResetP
roperty|UndefineProperty|PropagationAction)*>
<!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;
>

<!ELEMENT AddToProperty EMPTY>
<!ATTLIST AddToProperty
    %name;
    value CDATA #REQUIRED
>

<!ELEMENT RemoveFromProperty EMPTY>
<!ATTLIST RemoveFromProperty
    %name;
    value CDATA #REQUIRED
>

<!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
    order CDATA "0"
    serviceClass CDATA #IMPLIED
    cardinality (duplicate|singleton) "duplicate"
    resource CDATA #IMPLIED
    %overridable;
        %targetFile;
>

<!--ELEMENT ConfigurationRef EMPTY>
<!--ATTLIST ConfigurationRef
    xlink:href CDATA #REQUIRED
>

<!--ELEMENT PropagationAction (ClassPathEntry)*>
<!--ATTLIST PropagationAction
    className CDATA #REQUIRED
>

<!--ELEMENT ClassPathEntry EMPTY>
<!--ATTLIST ClassPathEntry
    dir CDATA #IMPLIED
    file CDATA #IMPLIED
>

```



## 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 Foundation & 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.

Additionally on Windows, you can start and stop Windchill from your desktop or from the **Start** menu (if you chose those options when installing Windchill).

## 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 access the 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 preceded and followed by an asterisk (\*) wild card character 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.

The screenshot shows a web interface titled "Edit Properties". At the top, there is a navigation bar with links: action, federation, notify, pom, presentation (highlighted), register, wt, wvs. Below the navigation bar is a search section with a "Search For:" text box and a "Search" button. The main content area displays a table of properties. The table has three columns: "Name", "Value", and "Default". The "Current View" is set to "Advanced". The table lists three properties: "netmarkets.presentation.author" with value "PTC", "netmarkets.presentation.cssFiles" with an empty value field, and "netmarkets.presentation.website" with value "http://www.ptc.com". Each row has a checkbox in the "Default" column, all of which are unchecked. At the bottom right of the table, there are "OK" and "Reset" buttons.

Name	Value	Default
netmarkets.presentation.author	PTC	<input type="checkbox"/>
netmarkets.presentation.cssFiles		<input type="checkbox"/>
netmarkets.presentation.website	http://www.ptc.com	<input type="checkbox"/>

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 the 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:

HTTPGateway MethodServer ServerManager

Edit Logs

Containing:

Display Results From: ☐ Start ☒ End

Messages:

Search

e:/ptc/Windchill735/logs/MethodServer.log

Tue 4/22/03 10:47:10: Thread-187: WTAdapter Create-TypeInstance: FIELD: the  
Tue 4/22/03 10:47:10: Thread-187: WTAdapter Create-TypeInstance: FORMAT: f  
Tue 4/22/03 10:47:10: Thread-187: WTAdapter Create-TypeInstance: GROUP\_OUT  
Tue 4/22/03 10:56:30: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 105 (1  
Tue 4/22/03 11:10:11: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 1 (0.1  
Tue 4/22/03 11:18:51: NotificationQueuePollingThread: StatementCache: wt.pom.St  
Tue 4/22/03 11:39:27: QueueCheckerThread: StatementCache: wt.pom.StatementC  
Tue 4/22/03 12:26:18: forumEventPropogationQueuePollingThread: StatementCache  
Tue 4/22/03 12:53:28: QueueCheckerThread: StatementCache: wt.pom.StatementC  
Tue 4/22/03 14:09:28: QueueCheckerThread: StatementCache: wt.pom.StatementC  
Tue 4/22/03 14:43:13: WfUserWorkQueuePollingThread: StatementCache: wt.pom.:  
Tue 4/22/03 15:09:28: QueueCheckerThread: StatementCache: wt.pom.StatementC  
Tue 4/22/03 15:43:38: WfPropagationQueuePollingThread: StatementCache: wt.pom  
Tue 4/22/03 17:02:18: NotificationScheduleQueueScheduleingThread: StatementCa  
Tue 4/22/03 17:09:12: forumEventPropogationQueuePollingThread: StatementCache  
Tue 4/22/03 17:33:22: WfPropagationQueuePollingThread: StatementCache: wt.pom  
Tue 4/22/03 18:25:43: WfUserWorkQueuePollingThread: StatementCache: wt.pom.:  
Tue 4/22/03 18:51:04: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 13 (1.  
Tue 4/22/03 19:01:50: MethodSummaryWriter: SUMMARY: 10.0 min, calls = 12 (1.

## 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:

**E-mail Search Results**

Separate multiple recipients in each field with a space character.

E-mail To:

Copy E-mail To:

Subject:

Send

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
- moduleRegistry.properties
- moduleDir.properties
- debug.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 the 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 {-fFhuwvV} {-r <product_root>} {-d <property_names>} {-s  
<property_pair>} {-t <property_file>} {--add <property_pair>} {--remove  
<property_pair>} {--reset <property_names>} {--setfromfile <property_file>}  
{--undefine <property_names>} {-i <declarative_xconf>} {--validateassite  
<site_xconf>} {--validateasdecl <declarative_xconf>} {--validatefilesassite  
<site_list_file>} {--validatefilesasdecl <declar_list_file>} {-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.

The following variables are used in the syntax of multiple parameters:

- *<property\_pair>* is a command-line escaped *name=value* pair that is compatible with the specification for java.util.Properties. For an example, see [Setting Property Values and Propagating Your Changes](#).
- *<property\_names>* is a comma-separated list of property names.
- *<property\_file>* is the relative or full path name of the property file.
- *<declarative\_xconf>* is either a full URL or relative file path to the declarative XCONF file.

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

Parameter Name	Description
--add	<p>Add the specified value at the end of the set of ordered values already defined in the property. Use this option only when the property is declared as a multi-valued property.</p> <p>To determine if property is multi-valued, you can display the current set of values using the -d parameter. The output from this parameter lists the multivalue separator when the property is multi-valued.</p>

Parameter Name	Description
-d or --describe	<p>Lists the values that are currently set and the corresponding XCONF file where each value is set for the specified properties.</p> <p>This option executes after all parameter setting options and the -p option have executed.</p>
-F or --force	<p>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.</p> <p>Use this option in place of -p if you suspect that there are problems with file timestamps or you want to switch between the -w and -u options.</p>
-f or --forcescan	<p>Forces the propagator to ignore its cache of XCONF-to-properties file dependencies. This option is ignored if you specify -F.</p> <p>Use this option in place of -p if you suspect that the cache is out of date.</p>
-h or --help	Displays the help for the xconfmanager command.
-i or --install	<p>Installs a declarative XCONF file that you have created. New declarative XCONF files are used when creating additional property files. When you are adding code in which new properties can be set, you can choose to create a separate property file where the properties are stored. For details on what to put in the declarative XCONF file, see the <i>Windchill Customizer's Guide</i>.</p>
-p or --propagate	<p>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.</p> <p>Updated property files are accessed when the Windchill solution is restarted.</p>

Parameter Name	Description
-r or --productroot	<p>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.</p> <p>The default root directory is the bin directory where the Windchill solution is installed.</p>
--remove	<p>Removes the specified value that is in the set of ordered values defined in the property. Use this option only when the property is declared as a multi-valued property.</p> <p>To determine if a property is multi-valued, you can display the current set of values using the -d parameter. The output from this parameter lists the multivalue separator when the property is multi-valued.</p>
--reset	Resets the site specific value of a property or set of properties to the declared default values.
-s or --set	<p>Sets the named property to a specific value in the site.xconf file.</p> <p>To set multiple properties in the same target property file, use multiple occurrences of this parameter or use the --setfromfile parameter. To set multiple properties that are in different target property files, enter multiple xconfmanager commands, one for each target file.</p> <p>Use this parameter in conjunction with the -t parameter.</p>
--setfromfile	<p>Adds the <i>name=value</i> pairs that are in the specified file to the end of the site.xconf file, thus setting each property named to the specified value. There is no checking done to determine if the value set is the default.</p> <p><i>&lt;property_file&gt;</i> is the file that contains a set of <i>name=value</i> pairs (one pair per line) that indicate the properties and values you want set in one target property file. Each pair sets a value for one property.</p> <p>With this parameter, you can set multiple properties in the same target property file using one xconfmanager command. To set properties that are in different target property files, enter multiple xconfmanager commands, one for each target file.</p> <p>Use this parameter in conjunction with the -t parameter.</p>



Parameter Name	Description
-t or --targetfile	Identifies the property file in which the property value specified in the -s parameter is set or the property values specified in the --setfromfile parameter are set.  Use this parameter in conjunction with either the -s or --setfromfile parameter.
-u w or --unix win	Indicates the platform for which the property files are to be generated. Normally, the current platform settings determine the format of the property files.  Include this parameter when you want to generate property files for a specific platform that is not the current platform.  Specify -u or --unix for UNIX platforms.  Specify -w or --win for Windows platforms.
--undefine	Resets the specified properties such that their values will be null (instead of an empty string) when read through a java.util.Properties instance.
-v	Turns on verbose console output, which shows full exception stack traces.
-V	Turns on debug verbose console output. This option shows full exception stack traces and additional information.
--validateasdecl	Validates a specific file as a declarative XCONF file.  Returns a non-zero result if file cannot be validated.
--validatefilesasdecl	Validates a list of files as declarative XCONF files. The list is contained in the specified file, where each line in the file is either a full URL or relative file path to a declarative XCONF file.  Returns a non-zero result if any of the files cannot be validated.  <declar_list_file> is either a full URL or relative file path to the file containing the list of declarative XCONF files you want to validate.

Parameter Name	Description
--validatefilesassite	<p>Validates a list of files as site-specific XCONF files. The list is contained in the specified file, where each line in the file is either a full URL or relative file path to a site-specific XCONF file.</p> <p>Returns a non-zero result if any of the files cannot be validated.</p> <p><i>&lt;site_list_file&gt;</i> is either a full URL or relative file path to the file containing the list of site-specific XCONF files you want to validate.</p>
--validateassite	<p>Validates a specific file as a site-specific XCONF file.</p> <p>Returns a non-zero result if file cannot be validated.</p> <p><i>&lt;site_xconf&gt;</i> is either a full URL or relative file path to the site-specific XCONF file you want to validate.</p>

**Note:** The xconfmanager executes the -s, --reset, --add, --remove, 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, --add, --remove, 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, --add, --remove, --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.
- Add and remove property values from properties that are multi-valued using the --add and --remove options.

- 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 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 WTemp directory that is under the Windchill installation directory [as defined by \$(wt.home)]:

```
xconfmanager -s wt.temp=$(wt.home)/WTemp -t wt.properties -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\.." -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 UnsetProperty 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).

## Adding and Removing a Property Value to a Multi-valued Property

To add a new classpath entry (d:\MyLibraries\somelibrary.jar) to the Windchill end of the classpath specified in the wt.java.classpath property, execute the following command from the windchill shell:

```
xconfmanager --add wt.java.classpath=d:\MyLibraries\somelibrary.jar -p
```

The value d:\MyLibraries\somelibrary.jar will be added to the end of the ordered set. You do not have to specify the delimiter \$(path.sep) as this will be added to the property value automatically by the xconfmanager.

To remove the classpath entry added in the previous example from the wt.java.classpath property, execute the following command from the windchill shell:

```
xconfmanager --remove wt.java.classpath=d:\MyLibraries\somelibrary.jar -p
```

The value d:\MyLibraries\somelibrary.jar is removed.

**Tip:** The previous example commands do not include the target file (in the -t parameter). The target file is not needed when the property is known to be in only one existing property file.

## 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
```

## Validating XCONF Files

You can use the following options to validate XCONF files:

- Use `--validateassite` to validate a site-specific XCONF file or `--validatefilesassite` to validate a list of site-specific XCONF files.
- Use `--validateasdecl` to validate a declarative XCONF file or `--validatefilesasdecl` to validate a list of declarative XCONF files.

The following section provides examples.

## Validating XCONF Files Examples

To validate a single file as a site-specific XCONF file, run the command:

```
xconfmanager --validateassite=<site_xconf>
```

If the file is valid, then the xconfmanager will issue no output and exit with a return code of zero.

To validate that several files are valid site XCONF files in one invocation, there are two options. You can use the `--validateassite` parameter multiple times. For example,

```
xconfmanager --validateassite=<site_xconf>
--validateassite=<site_xconf>
```

The other option is to create a text file, add a line for each path to a file to be validated, then run the command:

```
xconfmanager --validatefilesassite=<site_list_file>
```

If all the files are considered valid site XCONF files, xconfmanager issues no output and exits with a return code of zero.

You can validate declarative XCONF files in the same manner using the `--validateasdecl` and `--validatefilesasdecl` parameters.

## 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 XCONF 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.
- To generate properties in a format different from the current platform setting, use one of the following:
  - For the UNIX platform format, use -u.
  - For the Windows platform format, use -w.
- To turn on additional console output, use either -v (verbose) or -V (debug verbose).

## The Windchill Foundation & PDM Home Page

Following installation, you can open the Windchill Foundation & 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 Meetings

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, enable or disable Outlook compliant formatting for e-mail notifications, set up the WebEx conferencing URLs and IDs 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 Foundation & PDM, the **Meeting Center** icon appears in the icon bar at the top of the Windchill home page. This icon is identified below:



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.

## Configuring E-mail Notifications for Your Site

To enable or disable Outlook compliant formatting of e-mail notifications, modify the value of the `wt.meeting.outlookStyleNotifications` property using the `xconfmanager` utility, as follows:

- If your company uses Outlook as their standard e-mail client, set the value of this property to *true*. When this property is set to true, e-mail notifications are sent using an Outlook compliant format and are localized in the language of the meeting originator. The e-mail notification contains all the meeting participants as attendees.
- If your company does not use Outlook as their standard e-mail client, set this property to *false*. When this property is set to false, e-mail notifications only contain the originator and recipients as attendees. The e-mail notifications are localized for the recipient.

In the following example, the command is executed from the windchill shell and the value of the `wt.meeting.outlookStyleNotifications` property is set to *true*:

```
xconfmanager -s wt.meeting.outlookStyleNotifications=true  
-t <Windchill>/codebase/wt.properties -p
```

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

## WebEx Meeting Center Setup

To enable web-based meetings in the Meeting Center and to connect to the WebEx server, you have one or more options depending on which Windchill solution you have installed:

- For Windchill Foundation & PDM, Meeting Center is available only through the **Meeting Center** icon. Therefore, only one conferencing URL and ID can be set up. This is done by setting properties in the `wt.properties` file as described in [Meeting Center Setup for Your Site](#).
- For Windchill ProjectLink, users can create meetings from either the **Home** tab or the **Project** tab. In both of these cases, there is an associated organization container in which you can set a unique conferencing URL and ID:
  - The organization container used when a meeting is set up from the **Home** tab is the organization container associated with the meeting creator's organization. If a meeting creator's organization does not have an associated organization container or the conferencing URL and ID are not set in the container, then the general site conferencing URL and ID are used (if set up).
  - The organization container used when a meeting is set up from the **Project** tab is the organization container associated with the current project (not necessarily the organization of the user). If the conferencing

URL and ID are not set in the associated organization container, then the general site conferencing URL and ID are used (if set up).

Setting conferencing URLs and IDs in organization containers is described in [Meeting Center Setup for Individual Organization Containers](#).

Setting a general conferencing URL and ID for your site is done by setting properties in the wt.properties file as described in [Meeting Center Setup for Your Site](#).

- For Windchill PDMLink, users can only create meetings from the **Home** tab. Therefore, the organization container used for determining the conferencing URL and ID for each meeting is the organization container associated with the meeting creator's organization. If a meeting creator's organization does not have an associated organization container or the conferencing URL and ID are not set in the associated organization container, then the general site conferencing URL and ID are used (if set up).

Setting conferencing URLs and IDs in organization containers is described in [Meeting Center Setup for Individual Organization Containers](#).

Setting a general conferencing URL and ID for your site is done by setting properties in the wt.properties file as described in [Meeting Center Setup for Your Site](#).

## Meeting Center Setup for Your Site

To set up a Meeting Center site conferencing URL and ID, add the following properties to the wt.properties file using the xconfmanager.

Property	Description
wt.meeting.centerUrl	Specifies the URL for the WebEx server. For example, http://ptc.webex.com
wt.meeting.partnerId	Specifies the partner ID for the WebEx server.

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.

## Meeting Center Setup for Individual Organization Containers

When an organization container is created, the creator may have included the Meeting Center conferencing URL and ID in the **Conferencing URL** and **Conferencing ID** fields. If the fields were not populated when the container was

created, you can update the organization container and add the Meeting Center conferencing URL and ID.

To add a Meeting Center conferencing URL and ID for a specific organization container, complete the following steps:

1. On the **Organization** tab, display the organization that you want to update.
2. Click .
- The **Update Organization** window appears.
3. Enter information in the following fields:
  - In the **Conferencing URL** field, enter the URL for the WebEx server. For example, enter `http://ptc.webex.com`.
  - In the **Conferencing ID** field, enter the partner ID for the WebEx server.
4. Click **OK** to save your changes.

## 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 principals (consisting of WTOrganization objects and a directory service entry) 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 one initial organization container. This organization container represents your enterprise and is associated with an organization principal. The users associated with this organization container either author product information or in some way are consumers of this information.

As you define your digital product value-chain, additional organization principals can be created to represent your suppliers, contract manufacturers, or design partners. Each of these organization principals may have a unique identifier such as a CAGE or DUNS code. Optionally, the various organization principals 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 principal and possibly identified according to the 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 Foundation & PDM has only one organization container (and corresponding organization principal) that is created during installation. Additional organization principals 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 principals) can be created for each of the business organizations or business units that are collaborating together through the

Windchill solutions. Each organization container 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 container 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 container.

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 container 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 container 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 its 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 principals using the Principal Administrator, the principal 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 have not 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 principal through Principal Administrator, the organization principal 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 containers 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 principals 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 principal or update an existing organization principal to include the following attributes on the principal:

- 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, you can specify a new list by adding the `wt.org.organizationTypes` property to the `wt.properties` file. The `wt.org.organizationTypes` property value is a comma-separated list of coding systems, where each coding system has two parts in the following format:

`<ICD_number>/<coding_system_name>`

where:

`<ICD_number>` is the international code designator number assigned to the coding system. For example, the CAGE ICD number is 0141. For a list of ICD numbers, see the [Registration Authority](#).

`<coding_system_name>` is the name of the coding system.

If the `wt.org.organizationTypes` property does not exist in the `wt.properties` file, then the following `wt.org.organizationTypes` property value is used:

`0141/CAGE,0060/DUNS,0026/ISO6523`

If you add the `wt.org.organizationTypes` property, be sure to include all coding systems that you want to appear in the drop-down list. For example, to include CAGE, DUNS, and GTE/OSI Network, set the `wt.org.organizationTypes` property by entering the following `xconfmanager` command from a windchill shell:

```
xconfmanager -s wt.org.organizationTypes=0141/CAGE,0060/DUNS,0126/GTE/OSI Network
-t wt.properties -p
```

**Note:** After restarting the method server with the above property set, the **Organization ID Type** drop-down list contains **CAGE**, **DUNS**, and **GTE/OSI Network**, but not **ISO6523**.

Windchill does no checking to ensure that the international code designator numbers and coding system names set are valid. It is your responsibility to ensure the use of valid coding systems.

- 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

Windchill Desktop Integration (DTI) allows you to create and edit Windchill objects in Microsoft Office applications. As an administrator, you can:

- Configure the Desktop Integration Enterprise Search settings.
- Set preferences that specify whether Desktop Integration is used for primary content downloads and whether users are prompted to install Desktop Integration on primary content download.

The following sections provide the details for these actions.

### Configuring Desktop Integration When Enterprise Search is not Used

If your site does not use Enterprise Search in Windchill PDMLink, you can configure Desktop Integration so that the user interface does not display the **Keyword** search option.

Desktop Integration uses the wtSearch.xml file for its 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 are made on the server, and the changes are automatically propagated to the client the next time the user connects to the server.

**Note:** Any changes made to the wtSearch.xml file should be treated as a customization to your code. Be sure to follow the coding practices introduced in [Windchill Software Maintenance and Best Practices](#).

Removing the keyword field element from the wtSearch.xml file removes the **Keyword** field from the Desktop Integration search (the **Keyword** field is only used for Enterprise Search functions).

To remove the keyword field element, locate the search form element, which begins with the following tag:

```
<form key="search">
```

Within the search form element, delete the field element that has the key=keyword attribute. This element is in bold type in the following sample search form element:

```
<form key="search">  
  <searchfields key="criteria_string">  
    <field key="number"></field>
```

```

        <field key="name"></field>
        <field key="keyword"></field>
    </searchfields>
    .
    .
    .
</form>

```

Other customizations of this file are not supported.

## Setting Desktop Integration Preferences

The following preferences control the download behavior and usage of Desktop Integration when downloading content from HTML clients:

- /com/ptc/windchill/enterprise/nativeapp/msoi/MSOIDetectionForDownload
- /com/ptc/windchill/enterprise/nativeapp/msoi/MSOIDetectionFailoverPrompt

As is true for all preferences, you can set these preferences for different scopes, including container scopes and the Windchill Enterprise scope, using Preference Administrator. For details, see [Managing User Preferences](#), later in this chapter.

The following sections provide details about these preferences.

### Using Desktop Integration for Primary Content Downloads

The /com/ptc/windchill/enterprise/nativeapp/msoi/MSOIDetectionForDownload preference controls whether Desktop Integration is used for primary content downloads.

The system default for this preference is true.

When this preference is set to true, Desktop Integration is used to download the files rather than the HTML client download behaviors (either HTTP downloads or applet downloads). To use Desktop Integration for interaction with Windchill, this value must be true for the user doing the download. By default, the preference is set to true for all users.

If this preference is set to false, then primary content files that are downloaded use HTML client download behaviors. Then when the downloaded document is opened in an Office application, the document actions under the Windchill menu are grayed out.

### Prompting to Install Desktop Integration

The /com/ptc/windchill/enterprise/nativeapp/msoi/MSOIDetectionFailoverPrompt preference controls whether users are prompted to install Desktop Integration on a primary content download.

The system default for this preference is true.

When this preference is set to true, the client system is checked for the installation of Desktop Integration and, if not found, the user is prompted to install Desktop Integration. If the user cancels this prompt, the user preference for the user is set to false.

If this preference is set to false, users are not prompted to install Desktop Integration and must go to the **Software Downloads** page (available from **Home > Utilities**) to install it.

## Setting Up Internet Explorer for Downloads on Windows XP Clients

Many Windchill applications provide the ability to download files. However, the Microsoft Internet Explorer internet option on Windows XP systems can be set to not allow the downloading of files over the internet. To change this option, instruct users to complete the following steps from Internet Explorer:

1. Click **Tools > Internet Options**
2. From the **Security** tab, click the **Internet** icon.
3. Click **Custom Level**.
4. From the **Downloads** node in the tree, click **Enable** under **File download**.
5. Click **OK** two times.

## 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.

Organization	User ID	Full Name	EMail	Final Access	Project
ptc	ldavinci	Leo Davinci	ldavinci@ptc.com	2002-08-07 15:03:48.0	Waffle Iron
ptc	jmcgee	Jim Bob McGee	jmcgee@ptc.com	2002-08-07 14:57:54.0	Demo Project
ptc	Administrator	admin	WindchillAdministrator...	2002-08-07 10:40:47.0	Golf Course
ptc	pcollins	Pat Collins	pcollins@ptc.com	2002-08-07 14:58:22.0	Demo Project2
ptc	gkahn1	Ghengis Kahn	gkahn1@ptc.com	2002-08-07 15:02:51.0	Golf Course
ptc	sswenson	Scott Swenson	sswenson@ptc.com	2002-08-07 14:56:13.0	Demo Project
ptc	bwilson	Bob Wilson	bwilson@ptc.com	2002-08-07 15:01:00.0	Waffle Iron
ptc had:	8 Users				
Grand	Total:	8 Users			

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.**

---

**Organization: ptc**

Organization	User ID	Name	EMail	Last Access	Project
ptc	smorris	Sarah Morris	smorris@ptc.com	2003-09-28 10:43:53.0	Test Project 2
ptc	nhernandez	Nick Hernandez	nhernandez@ptc.com	2003-10-22 16:17:52.0	Test Project 1
ptc	ameyers	Austin Meyers	ameyers@ptc.com	2003-10-22 13:25:43.0	Test Project 1
ptc	dvalder	Dave Valder	dvalder@ptc.com	2003-10-18 11:21:56.0	Test Project 2
ptc	jpark	Jane Park	jpark@ptc.com	2003-10-12 15:24:52.0	Test Project 2
ptc	jwilson	James Wilson	jwilson@ptc.com	2003-10-21 14:18:59.0	Test Project 2

**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 manage user preferences and define preference scopes.

## 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.

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 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.

The Preference Administrator preference table (located in `../wtcore/jsp/wt/prefs/admin/PrefHelp.jsp`) displays the system default preferences along with descriptions and default values. For example, the preference table provides information on the following preferences:

- The `com/ptc/windchill/enterprise/templates/doc/renameFileFormatOnCreateFromTemplate` site preference allows you to

configure the default name that appears when creating a document from the template. For the options available, see the PrefHelp.jsp.

- By default, the /wt/part/display\_thumbnail\_in\_table preference is set to false and the Action icon to display the thumbnail is not present in the table. If the preference is set to true, the **Action** icon (not the thumbnail itself) is displayed and it can be used to display the thumbnail.
- The /transitions/PROMOTE/\$DEFAULT preference provides the default transition processes for a site. For details on the use of this preference, see the *Windchill Business Administrator's Guide*.
- The ProjectLink/localTimeZone preference sets the time zone for users. When this preference is not set, a user's time zone is taken from system time on the client machine. If you set this preference in the Windchill Enterprise scope, then all users have the same time zone set. The value you can enter for this preference is a Java TimeZone ID, such as "Singapore", "Asia/Tokyo", "America/Los\_Angeles", or "Atlantic/Canary".
- The /com/ptc/windchill/enterprise/search/allSearchTypes preference identifies the types that are searchable. For information about updating searchable types, see the *Windchill Customizer's Guide*.

This is only a small subset of the preferences available. Your Windchill solution provides default settings for preferences that you then can change for the entire site, organization, or individual containers. Users also have the ability to change many preferences from within the Windchill user interface. As part of your initial set up, you should review the preferences and change preferences as required by your site.

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 the Preference Administrator preference 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.

## 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:

```
$DEFAULT,$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:

```
$DEFAULT,$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 getLocalizedString(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.

## 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 database should be backed up on a regular basis. Vendor 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.
- Multiple concurrent instances of Windchill solutions on the same server. For information on setting up multiple instances of Windchill on the same server,

see the Installation Planning Requirements chapter of the *Windchill Info\*Engine Installation and Configuration Guide*.

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
- An LDAP server
- The Convera RetrievalWare search engine (optional)
- 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 `wt.properties` configuration starts up two method servers. One is dedicated to running background queues:

```
#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
```

Background queues can also be grouped to execute on specific method servers; see the [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 wt.properties file control load-balancing behavior.

### Server Selection

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

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

The getServer(String service) and getNextServer(String service, Remote server) methods in the server selector interface deal with load balancing. The 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.

Server Selector	Description
wt.manager.StandardServerSelector	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.
wt.manager.BalancedServerSelector	Returns the first server on a getServer(). Returns the least recently used server on a getNextServer.

## 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.ServerLoadException` 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:

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

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

## Configuring Log Files for Multiple Servers

To measure the distribution of transactions across all method servers, it can be useful to configure separate log files for each method server. To help set up unique log files, PTC provides a utility class that can parse the launch command and substitute numeric identifiers for tokens.

To configure the launch utility, use the `xconfmanager` utility or System Configurator to set the following property in the `wt.properties` file:

```
wt.manager.cmd.MethodServerLauncher=wt.method.MethodServerLauncher
```

Setting this property instructs the server launcher to delegate method server launching to the utility class.

To configure each method server to write to separate log files on a Windows system, use the `xconfmanager` utility or System Configurator to modify the value of the `wt.manager.cmd.MethodServer` property in the `wt.properties` file as follows:

```
wt.manager.cmd.MethodServer=\
cmd.exe /C start "MethodServer {0}" \
"${wt.java.cmd}" -classpath "${wt.java.classpath}" \
-Xms32m Xmx64m -Xnoclassgc wt.method.MethodServerMain \
wt.method.access.log.file=${wt.logs.dir}\\access{0}.csv\
wt.method.log.file=${wt.logs.dir}\\MethodServer{0}.log
```



As a result, each method server runs in a separate console window with a numeric identifier appearing in the title bar. The same numeric identifier is appended to both the method server log and access.csv files.

The UNIX equivalent can be achieved by launching each method server from a script. For example, you can set the wt.manager.cmd.MethodServer property to the launch\_ms.ksh script:

```
wt.manager.cmd.MethodServer=/opt/ptc/Windchill/launch_ms.ksh {0}
```

Where the launch\_ms.ksh script is:

```
#!/bin/ksh

/usr/java/bin/java -Xms32m -Xmx64m -Xnoclassgc -noverify
wt.method.MethodServerMain \
wt.method.access.log.file=/opt/ptc/Windchill/logs/access$1.log
\ wt.method.log.file=/opt/ptc/Windchill/logs/MethodServer$1.log
\ > /opt/ptc/Windchill/logs/heap$1.log 2>&1
```

## Changing Authentication Text Between Servlet and Windchill Adapter

As part of the installation, Windchill solutions automatically set the value for the secret.text2 and secret.text properties in the ie.properties file to a value of "mySecret" to establish a more secure authentication process between your servlet and the Windchill adapter. PTC suggests that you reset these properties to a unique value of your choice. 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 and later releases, 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, similar to a password, that is used to sign outgoing requests and validate incoming requests. Similar to other service-specific Info\*Engine properties, the property names must include a prefix of the runtime service name of the service to which the property applies. If it is unclear what the full name of your property should be, examine the contents of the

<Windchill>/codebase/WEB-INF/ie.properties file to see the automatically generated properties created by the installers. For security reasons, PTC suggests that the secret properties be placed only in ie.properties files and not in LDAP configuration. When multiple, separate installations are involved, and you wish to allow communications between services, the secret property values for all services involved must be duplicated between all ie.properties files so that systems generating requests can properly sign them.

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. PTC recommends using a secret.text value between 6 and 18 characters. The ie.properties 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 -p -s  
"<runTimeServiceName>.secret.text2=<your_secret_value>" -t  
"<Windchill>\codebase\WEB-INF\ieStructProperties.txt"
```

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 applied to the configuration of any remote services that will make requests into your Windchill system. Likewise, any similar configuration for remote Windchill systems, or out-of-process adapters, that your system will communicate with must be duplicated in your ie.properties file.

## Administering the Authentication Process

The Windchill architecture is designed to rely on Web server authentication to provide authenticated user names. Therefore, access controls maintained on the Web server determine access privileges to an authenticated Windchill URL or SysAdm URL based on a user name and password obtained by the Web browser.

The HTTP authentication implementation, described in more detail in the *Windchill Customizer'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 name 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 names, unless you have changed the value of this property to false.

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

## Troubleshooting User Authentication

### Double Authentication Can Be Eliminated when Using Internet Explorer

If Internet Explorer users do not select the **Save this password in your password list** option when first logging in, they are required to log in again when accessing a Java applet within Windchill.

To eliminate the need to log in twice, enter the correct user name and password for the first Windchill login in the Internet Explorer authentication window. Then, select **Save this password in your password list**. Selecting this option allows the Web browser to remember the credentials and users will not have to authenticate to the Java applet.

The **Save this password in your password list** option is available with Internet Explorer version 6.0 and greater when using JRE 1.4.2\_02 or greater.

**Note:** There is no system setting that takes care of this issue; you should inform users that they should select the option whenever they log in.

### Authentication Configuration Tools

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><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>
```

```

</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 name.

## 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 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:

Property	Description
<code>wt.scheduler.purgeHistoryItems</code>	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.
<code>wt.scheduler.purgeHistoryItemsInterval</code>	Specifies the number of days between the cleanups of history items thirty days old.

wt.scheduler.purgeHistoryItemsOlderThan	Specifies the age in days of the items that will be purged.
---	---

## Other Properties

Other properties related to Windchill Scheduler operations are the following:

Property	Description
wt.scheduler.verbose	Specifies whether to run the Windchill Scheduler in verbose mode. The default is false.
wt.scheduler.log.properties	Specifies whether to print out the Scheduler-specific properties on startup.
wt.scheduler.log.filename	Specifies the file to which the log file is written. The default is \$(wt.logs.dir)\$(dir.sep)Scheduler.log.
wt.scheduler.log.enabled	Specifies whether to save the log of Windchill Scheduler in a separate file. The default is false.
wt.scheduler.log.append	Specifies whether to append to the end of the existing log file. The default is true.

## Purging, Archiving, and Restoring Windchill Information

Purge is a new capability for collecting and removing documents and meta data from Windchill. Archive and restore are new capabilities for collecting, storing, and restoring information managed within a Windchill environment.

Administrators need to create jobs for Purge by specifying the query to collect the objects for purging. Archive Purge has a rich wizard-based facility for collecting objects for Purge. These objects can be optionally archived. Archive Purge uses the object collection framework for collecting objects. The following BaseCriterionDef objects are created OOTB as a part of the Object Collection that correspond to each of the Criterion Types available.

- context-id: Select the context to search in
- folder-membership: Find objects that are members of a folder
- object-type: Select objects of the types specified.
- lifecycle-state: Find object in lifecycle state
- view-reference: Find object in a given view

- version-id: Select specific versions
- iteration-id: Select the specified iterations.
- version-at-maturity: Find object version at the given maturity level
- created-by: Select objects based on the creator.
- create-date: Find object created before, between or after the input date(s)

The BaseCriterionDef that is used for processing dependencies is archive-dependency-map.

For the data to be archived, the administrator can also specify options to be included in the archive:

- Versions and Iterations -- Include either the latest iterations of the latest version, or the latest iterations of all versions.
- CAD Dependents -- Include either required dependencies only, or all dependencies.

Windchill loads four wt.dataops.objectcol.RelationshipMap objects that use the above loaded BaseCriterionDef object (archive-dependency-map) and correspond to each of the 4 options offered by Windchill (given the combination of two options viz. CAD Dependencies and Version and Iterations with two choices each), as part of the base data loading.

The following RelationshipMap objects are created OOTB and used for collecting CAD dependencies:

- latest-of-latest-n-reqd: Collect latest iterations of the latest version of required dependencies.
- latest-of-latest-n-all: Collect latest iterations of the latest version of all dependencies.
- latest-of-all-n-reqd: Collect latest iterations of the all version of required dependencies.
- latest-of-all-n-all: Collect latest iterations of the all version of all dependencies.

The following rules are created for the OOTB RelationshipMap objects:

Domain	Context	Type	Principal	Grant Permissions
Default	Site	RelationshipMap	ALL	Read
Default	Site	RelationshipMap	Administrators	Full Control(All)
System	Site	BaseCriterionDef	ALL	Read

Domain	Context	Type	Principal	Grant Permissions
System	Site	BaseCriterionDef	Administrators	Full Control(All)

The collection framework can be customized to allow a collection of a different set of dependencies than the ones provided OOTB. For further details refer to the chapter entitled, "Customizing Archive, Purge, and Restore" in the *Windchill Customizer's Guide*.

## Windchill Software Maintenance and Best Practices

Normal maintenance corrections and updates to the products of Windchill 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 Support Web site (<http://www.ptc.com/support/support.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 release. Datecode values for maintenance releases are of the form Mnnn where nnn is numeric, typically increasing by increments of 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, it is 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 Customizer's Guide*. 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 and 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 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 executes<sup>1</sup>, 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 three modes:

- **Changed Files Only**

Selecting this installation type copies all updated files to your system, but does not overwrite site-modified files and does not run configuration actions such as updating property files and building JAR files. Using this type, you can install the Windchill Service Pack files 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 do a complete installation to validate that the PTC updates, site modifications, and customizations operate together. After the validation is complete, you collect all the site customizations together for easy deployment to production.

- **Updates for Site-Modified Files to Safe Area**

Copies only the updated files that correspond to the files in the *<Windchill>/wtSafeArea/siteMod* directory are copied to the *ptcCurrent* directory.

Selecting this installation type does not do a complete installation and no installation registry entry is made. Instead, it gives you access to the updates for site-modified files so that you can manually update your modified files.

---

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.



On the test system, you can then do a complete installation to validate that the PTC updates, site modifications, and customizations operate together. After the validation is complete, you collect all the site customizations together for easy deployment to production.

- **Complete**

Copies PTC updates to site-modified files into the <Windchill>/wtSafeArea/ptcCurrent directory and copies all other updated PTC files into the selected directory, overwriting existing files where applicable. It also copies selected files from the wtSafeArea/siteMod directory to the selected directory. Select this type only after updating your site-modified files. First select either the **Changed Files Only** or **Updates for Site-Modified Files to Safe Area** type to obtain PTC updates for site-modified files.

Selecting this type also runs the configuration actions after files are updated.

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 have 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<sup>2</sup>. 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

---

2. After you perform a first time installation any 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 you add a new locale to your system through the Windchill Language Pack.

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 which a Windchill 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 in the *Windchill Customizer's Guide*. The topics in the guide 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.
- Properly managing modifications to existing HTML templates and creating new ones to replace standard PTC templates.
- 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.

## 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 ensure 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 `<Windchill>/bin/windchill` file. Locate 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:

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

Arguments (optional)	Description
-w, --wthome=DIR	<p>Sets the Windchill home directory. Default is the parent directory containing the windchill script.</p> <p><b>Note:</b> On UNIX systems where you have multiple instances of Windchill installed under the same user account, settings made to WT_HOME and SQLPATH environment variables by using this -w option are overridden by any settings to these same variables in the user's .cshrc, .login, and .profile shell initialization files.</p>
--java=JAVA_EXE	<p>The Java executable. Default is the wt.java.cmd variable value specified in the \$WT_HOME/code-base/wt.properties file.</p>
-cp, --classpath=PATH	<p>Java classpath. Default is the wt.java.classpath variable value specified in the \$WT_HOME/code-base/wt.properties file.</p>
--javaargs=JAVAARGS	<p>Java command line arguments.</p>

## windchill Actions

Action	Description
shell	Sets up a Windchill environment in a new instance of the currently running shell.
start	Starts the Windchill server.
stop	Stops the Windchill server.
status	Retrieves the status of the Windchill server.
version	Displays the Windchill installation version.
properties <resource> [,...][?key[&key2]...]	<p>Displays the properties as seen by Windchill for the given resource with substitution and other actions executed. It can be limited to a given set of keys.</p> <p>For example:</p> <p>windchill properties wt.properties — lists all wt.properties</p> <p>windchill properties wt.properties?wt.server.codebase — lists server codebase</p> <p>windchill properties wt.properties?wt.env.* — lists all the environment variables use by windchill shell</p> <p>windchill properties — with no arguments generates the help report</p>
CLASS [CLASS_ARGS]	<p>Run a Windchill class with optional class arguments. For example:</p> <p>windchill wt.load.Developer -UAOps</p>

## 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.



# 2

## Administering the Bootstrap Client and JAR Files

Topic	Page
Overview .....	2-2
About the Bootstrap Feature.....	2-2
Java Plug-in Cache .....	2-3
Bootstrap Configuration File.....	2-3
Bootstrap Package Versioning .....	2-7
Downloading JAR Files .....	2-8
Administering Codebases.....	2-8

## 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 Plug-in Cache

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

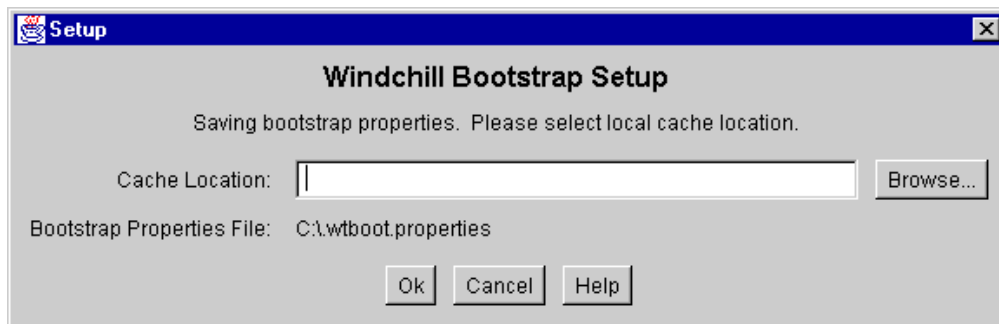
For information about the Java Plug-in, see the *Windchill Info\*Engine Installation and Configuration Guide*.

## 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

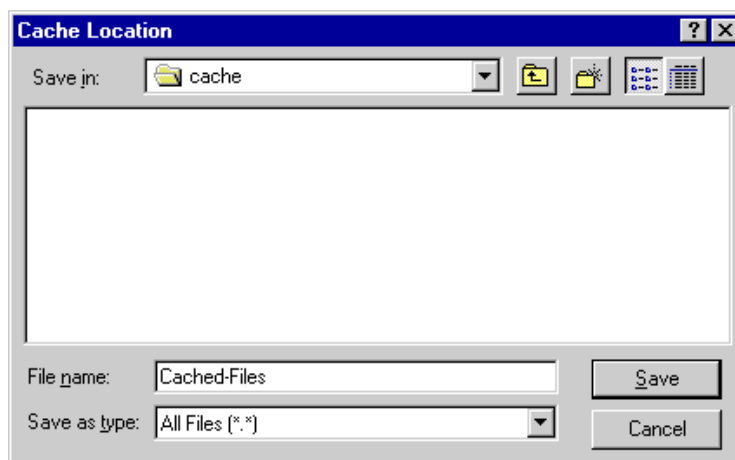
The first time you use the bootstrap feature, you are prompted for the location in which to cache JAR files.



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:

Property	Description
allowUserInteraction	Turns bootstrap user interaction on or off. Set to false, the autoDownload/autoInflate becomes important. Default is true.
autoDownload	Automatically downloads JAR files being cached by bootstrap. The user will be prompted. Default is false.
autoInflate	Automatically inflates the downloaded JAR files. The user will be prompted. Default is false.
cacheDir (required)	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.

Property	Description
captureFile	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  boot_preload  parameter when bootstrapping an applet or application.
captureFileStackTrace	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.
checkVersion	Accesses a particular server codebase and checks for a more recent version of the bootstrap package. Default is true.
enabled	Enables and disables bootstrap. Default is true.
rmiFailoverTimeout	Specifies, in milliseconds, the length of time for which the wt.boot.WTRMIMasterSocketFactory 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 wt.boot.rmiFailoverTimeout. Default is 10,000 (10 seconds).
rmiSocketFactory	Specifies the fully qualified class name of an RMI socket factory to be installed. Default is wt.boot.WTRMIMasterSocketFactory.
setProperty.xxx	Triggers the setting of arbitrary Java system properties, through a property naming convention. Any bootstrap property that has a name starting with setProperty specifies a setting for the system property identified by the remainder of the name. For example, setProperty.user.language=FR sets the user.language property to French.
showClasspath	Displays the Java class path. This can be used when debugging applets to see where the classes are being loaded from.
showMissingFiles	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.

Property	Description
useFullHostNames	Specifies if host names 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.

## 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

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:



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.

## Updating Client JAR Files

Client JAR files need to have the same versions of files that are located in *<Windchill>/codebase*. Site changes to displayed text and properties can require that JAR files downloaded to clients are updated.

**Note:** When the client JAR files are updated, end users download them as the applications detect the previously downloaded JAR files are out-of-date.

To ensure that the JAR files maintained through the MakeJar.xml script are updated correctly, you should add the following to the *<Windchill>/codebase/jarContents/Cust.bom*:

- Paths for the compiled resources (\*.ser and/or \*.class files) of the files you change
- Paths of site-modified property files

To verify that all site-modified property files are listed in Cust.bom, you can compare targetFile entries in site.xconf with the files listed in Cust.bom. Any files listed in targetFile entries that are not in Cust.bom should be added to Cust.bom. For example, if the site.xconf file has an entry for the following:

```
targetFile="codebase/wt/change2/change2.properties"
```

Then, ensure that *codebase/jarContents/Cust.bom* contains the following entry:

```
wt/change2/change2.properties
```

To rebuild the client JAR files that are managed by jarContents and jarManifest specifications, execute the following command from a windchill shell:

```
ant -f codebase/MakeJar.xml custUpdate
```

You must also ensure that the following client JAR files are updated:

- JAR files used by workgroup managers
- JAR files used by the Optegra Gateway
- JAR files created by site personnel that use \*.jar.config JAR files

The following command, executed from a windchill shell, ensures that these client JAR files are updated:

```
ant -f bin/swmaint.xml make_jar.config_jars
```

For information about the swmaint.xml script, see the *Windchill Customizer's Guide*.

## 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).

## **Determining Client JAR Contents**

For optimal performance, all resources (for example, .class and .properties files) needed by a Windchill applet should be contained within one of the client JAR files it uses. Each resource not found within one of an applet's client JARs that is needed by the applet requires a separate round-trip network request and the resource is not compressed as it would be if it was part of a client JAR. This leads to poor performance on slow or wide-area networks. Providing the correct content in a client JAR file helps ensure optimal performance of the applet.

Despite PTC's best efforts, the client JARs for PTC applets may not contain all resources that they require. Similarly, if you have any customized applets, their JARs may not contain all of the custom resources they require. If you suspect that the client JAR file used by an applet does not have the correct contents, you can determine which missing resources should be added through the use of HTTP Request Log utility.

For information about the HTTP Request Log utility, see *Windchill Customizer's Guide*.



# 3

## Administering External File Vaults

Topic	Page
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
Out-of-the-box Background Queues for External Vaulting .....	3-9
Windchill External Storage Administrator .....	3-10
Creating a Vaulting Policy .....	3-15
Managing Revaulting .....	3-18
Changing the Location of Files in External Vaults .....	3-24
Moving a Server Using External Vaulting to a Different Host.....	3-25

## 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:

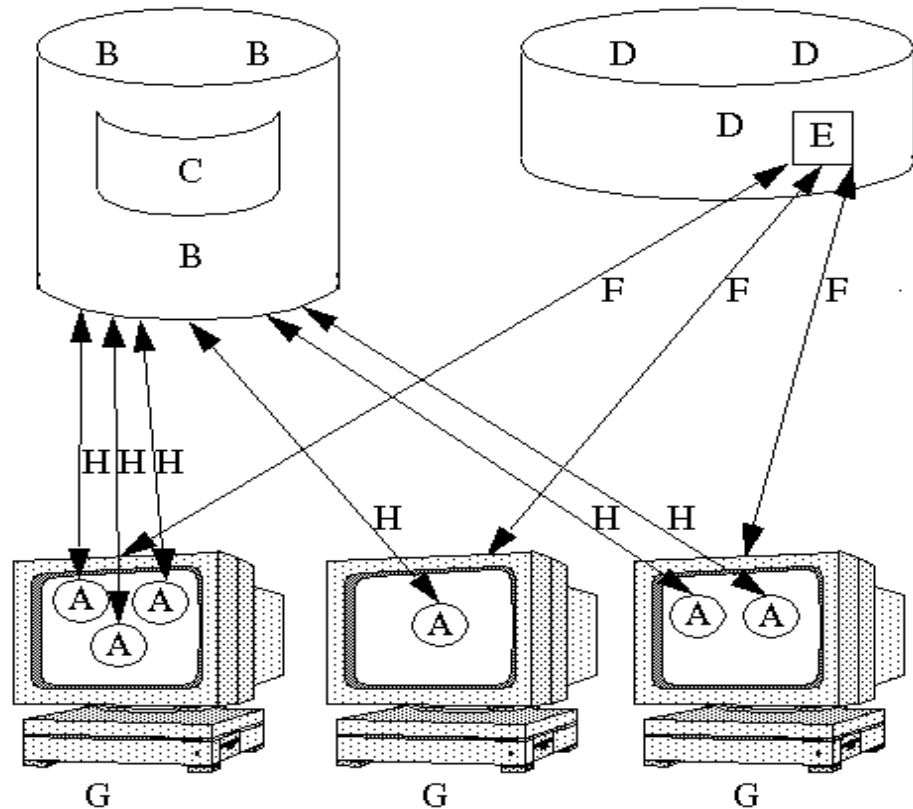
- **External File Vaulting** -- File vaulting allows you to store Windchill data outside the Windchill database in logical containers called *vaults*, each of which can refer to multiple physical memory locations called *folders*. Multiple hosts can work together in file vaulting to form *sites* or *clusters*. 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.
- **Import and Export** -- 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 items: 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 item. Using file vaulting, you can specify that, for a particular type of item 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 a Vendor database.
- C. BLOB -- The 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 item, 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 to enable faster file upload for certain applications without, or prior to, the System Administrator setting up a custom cache vault.

The central cache vault is initially located in the following temporary directory:

*<Windchill>/vaults/defaultcachevault*

**Note:** A write-enabled cache vault is required to create a document or CAD document in Windchill.

### To relocate the cache vault by relocating its folder

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.

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 (default), 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.

**Note:** Files are uploaded to the cache vault and the vault is revaulted according to established file vaulting rules. If `wt.fv.uploadtocache.revaultOnCommit=true` is set, revaulting occurs immediately after the upload transaction is finished. Otherwise, the content files are revaulted during the scheduled revaulting session. PTC recommends purging unreferenced files for the cache vault at the end of the backup procedure.

**Note:** 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.

Property	Description
wt.fv.verbose.properties	Determines whether or not file vault properties are outputted upon startup. The default is false.
wt.fv.verbose	Specifies whether the system outputs log information specific to the file vault feature. The default is false.
wt.fv.read.buffer_size	The size of the buffer for uploading files directed to a vault. The default is 8192 (8 KB).
wt.fv.cleanup.buf_size	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).
wt.fv.log.enabled	Enables the file vault log file. The default is false.
wt.fv.log.append	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.
wt.fv.log.filename	The name of the file vault log. The default is \$(wt.logs.dir)\FileVault.log.
wt.fv.log.mountInfoFile	The name of the log file for mounting information. The default is \$(wt.logs.dir)\MountInfo.log.
wt.fv.revaultOnChange	Determines whether or not the revaulting is performed in the background for the items changing their domain and/or life cycle state. The default is true.

Property	Description
wt.fv.revaultQuerySize	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.
wt.fv.uploadtocache. revaultOnCommit	<p>Determines whether or not the documents modified during the transaction will be added to the <i>RevaultCacheQueue</i> upon the completion of the transaction.</p> <p>The default is false. With property value is set to false, direct and immediate vaulting of WTDdocuments does not occur. (Setting the value to true may result in performance issues during upload.) When set to false, a revaulting schedule must be implemented to revault files.</p>
wt.fv.purgeUnreferenced FvItemsInterval	Determines the periodicity in days of the execution of the cleanup of unreferenced items. The default is 1.
wt.fv.purgeUnreferenced FvItemsOlderThan	Determines the age in days of unreferenced items which will be subject to cleanup. The default is 30.
wt.fv.forceContentToVault	Determines whether a single vault will be used for all content vaulting. The default is false. See Forcing Content to Vault.
wt.fv.useFvFileThreshold	If true, the property wt.fv.fvFileThreshold is effective. If false, wt.fv.fvFileThreshold has no effect.
wt.fv.fvFileThreshold	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 "n." 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. Refer to the following section for more information.

## Specifying the File Threshold Value

**Note:** In the following discussion, the term *directory* should be considered equivalent to *folder* in a Windows environment.

In some operating systems, the file system performance of directory access degrades significantly as the number of files in the directory grows beyond a certain threshold. This affects both reading the contents of the directory and adding files to the directory.

To ensure adequate Windchill performance in such environments, the ability to enforce a limit on the number of files per vault folder has been introduced. This applies to all non-DBMS Windchill storage locations for Windchill-managed application files, such as CAD files or text documents. In a new Windchill installation, the limit on application files per folder is set to a default value of 7000 files. This setting may be enabled or disabled through the following properties in the Windchill `wt.properties` file for each Windchill installation:

```
wt.fv.useFvFileThreshold=true
```

If the limit enforcement is enabled (*true*), then the actual file limit is set via the property:

```
wt.fv.fvFileThreshold=N
```

where *N* is the maximum number of files allowed per directory.

When this limit is set, and a currently writable folder reaches the limit on the number of files it can hold, Windchill automatically makes that folder read-only, and selects the next available folder to be writable.

**Note:** A folder may also be made read-only if there is no storage space left in the folder's underlying file system directory.

For performance reasons, the limit on the number of files per folder has a plus or minus 10% band. For example, if the limit is set to 10000, then the folder will become read-only at somewhere between 9000 and 11000 files.

If the file limit is enforced, processing time will be added to all file upload operations, degrading Windchill performance. Therefore, at the time Windchill is installed, the Windchill administrator should consult the specific operating system documentation to determine whether this limit is advisable for their Windchill system. If the limit is advisable, then it should be set to an appropriate value for the operating system. If, however, it is determined that enforcement of the limit is not advisable for the particular operating system, then the Windchill limit can be disabled.

Even with the Windchill limit disabled, the administrator should monitor the growth of the number of files in the Windchill folders. Eventually, a point will be reached when it is appropriate to manually discontinue additions to a currently writable folder by making it read-only, and switch to the next available folder for



further file uploads. For operating systems in which this point is a large enough number of files (for example, 100000), it may be beneficial to avoid the performance penalty of automatic enforcement by Windchill, and instead manually monitor folder growth.

## **Out-of-the-box Background Queues for External Vaulting**

The following sections describe the external vaulting queues that are established when your Windchill solution is installed.

### **PurgeScheduleHistoryQueue**

The PurgeScheduleHistoryQueue is used by external vaulting and content replication to track and clean up information about the old (outdated) schedules for revaulting and content replication. The queue contains only one item at any time.

A failed queue entry is an indication that old schedule histories could not be removed.

Checking for failed entries in this queue is not needed. If the entry in the queue is in the failed state, it will be recreated during the next startup of the method server.

### **PurgeUnreferencedFvItemQueue**

The PurgeUnreferencedFvItemQueue is used by external vaulting and Upload to Cache to track and clean up information about the outdated or failed attempts to perform upload of content to cache on the master site. The queue contains only one item at any time.

A failed queue entry is an indication that some outdated information about uploading to cache could not be removed.

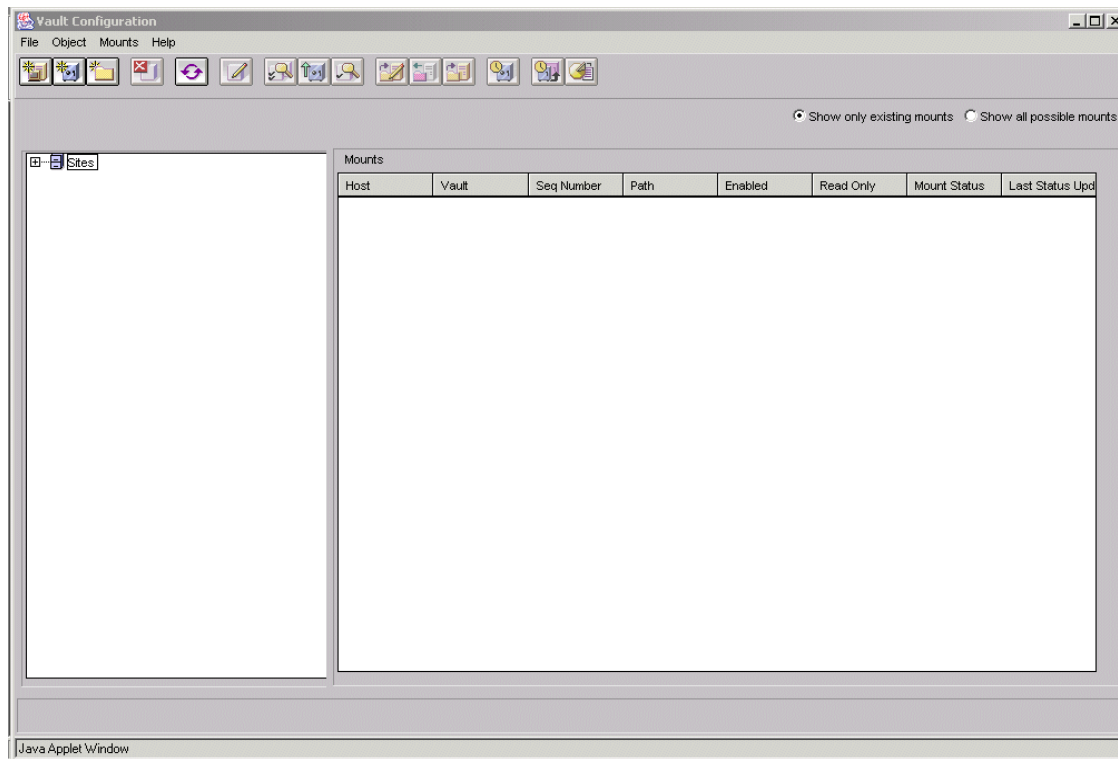
Checking for failed entries in this queue is not needed. If the entry in the queue is in the failed state, it will be recreated during the next startup of the method server.

# 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:



The left panel of the **Vault Configuration** window displays a tree view of the site item, which includes all of the hosts, vaults, and folders that have been defined for the site. (Each folder must have its own unique directory.)

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

When the option 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 open 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 items
- Clean up your vaults by removing unreferenced files.
- Validate a single mount or all mounts of an item.

## 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.

**Note:** 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.

**Note:** A write-enabled cache vault is required to create a document or CAD document in Windchill.

## 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. Complete 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 box.
  - b. From **Object** menu select **Update**.
  - c. In the **Update Folder** dialog box, 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 item. 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 (for example, in a cluster), all mount paths associated with one folder 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. Each vault folder must have its own unique directory.

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.



**Caution:** Each folder must have its own unique directory to store the content. For example, two folders should not have the same physical location. Failure to do so may result in data loss.

## 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 item. 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 items:

- 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 Policy Administrator 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 item with which they are associated.

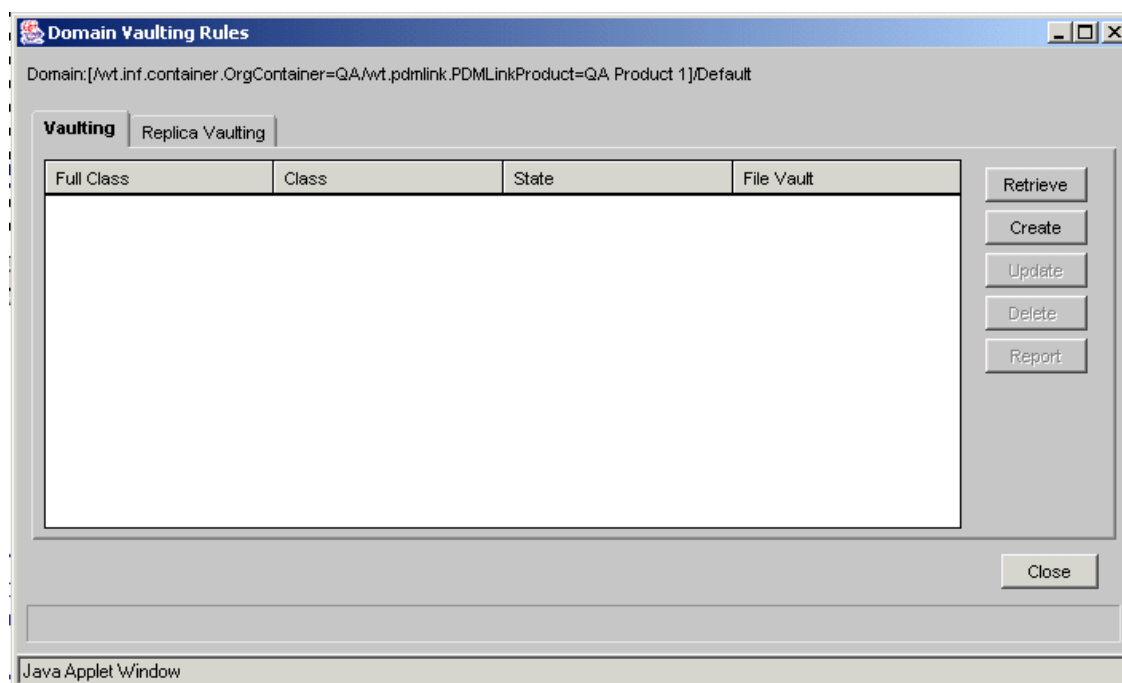
Windchill domains can be created in a hierarchical fashion, with some domains being children of other domains. It is important to note that a domain does not inherit the vaulting rules of its parent domain. Vaulting rules must be explicitly defined at each level of a domain hierarchy.

## 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.



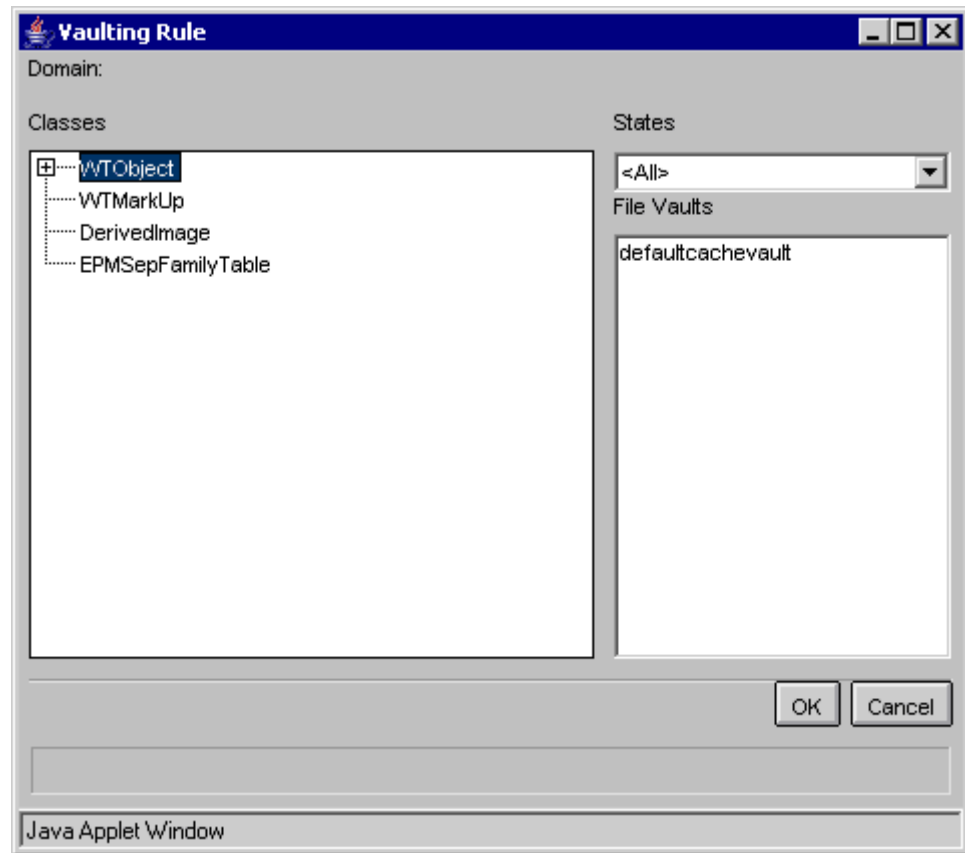
## Creating New Rules

When you create vaulting rules, you can use all items implementing the interface `wt.content.ContentHolder` that are `DomainAdministered` as well as `LifeCycleAdministered`. Additional items you can use are the following:

- `wt.viewmarkup.DerivedImage`
- `wt.viewmarkup.WTMarkUp`
- `wt.epm.familytable.EPMSepFamily Table`



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



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 item type and life cycle state combination can be linked to only one file vault.

**Note:** Do not apply the external file vault rule on the defaultcachevault.

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 item (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 item 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 item 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 includes 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 item storage to be in a BLOB. Revaulting may also be needed when a change occurs in the domain or life cycle state of an item that holds content files. The revaulting process for such item changes can be done in the background, which is administered by a property, wt.fv.revaultOnChange (default=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.

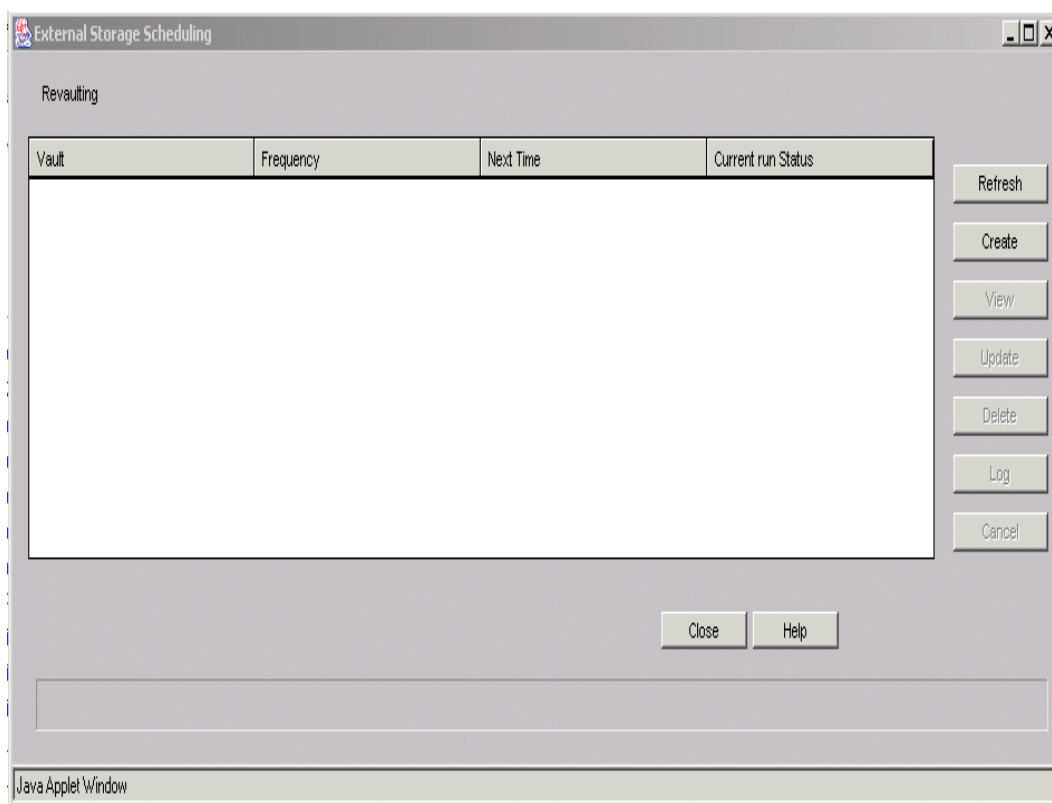
Revaulting has the potential to be a resource-intensive activity. Therefore, it needs to be managed. To designate a vault for revaulting, it is necessary to schedule when it is to be done. Creating a schedule item for the vault does this. At the scheduled time, the revaulting process is launched and 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.

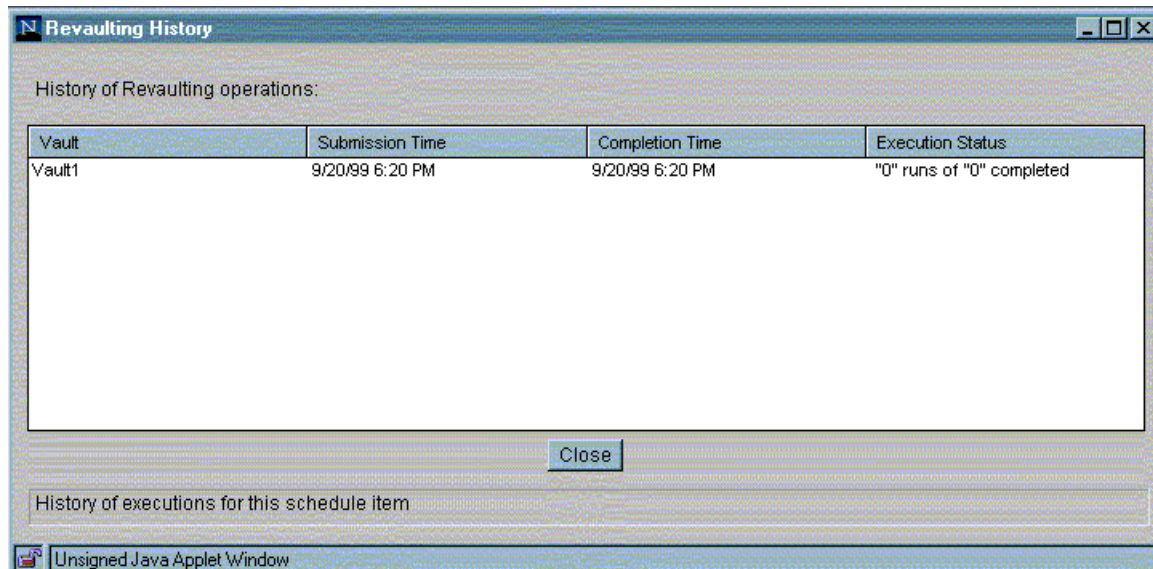


A list of vaults is displayed, for which revaulting has been scheduled. Each vault displays the frequency of revaulting, the time of the next revaulting run, and the status of the current run. A status of READY means that the run has been scheduled. Click **Refresh** to update the contents of the list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



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**.

**Tip:** 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.

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

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 **Close** to close the window without saving any current changes. Schedule items saved to the repository with the **Apply** button cannot 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 schedule 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 Method Server 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 Method Server 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 revaulting 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.

**Note:** A write-enabled cache vault is required in order to create a document or CAD document in Windchill.

- 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 BLOBs

Forcing content to a single vault increases performance by preventing content files from being stored in 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:  

```
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.  

```
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 by adding, for example, 100 folders.
8. Initiate revaulting to the designated cache vault so that any remaining data is moved from BLOBs to the vault.

## 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:

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 item 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** check box 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 database 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 item stored in the external vault.

# 4

## Administering Content Replication

Topic	Page
Overview .....	4-2
Modifying the Local Site.....	4-5
Replication Security .....	4-6
Editing Content Replication Properties.....	4-7
Out-of-the-box Background Queues for Content Replication .....	4-11
Content Replication and Windchill Visualization Service.....	4-13
Replication and Compression.....	4-14
Improving Content Replication Performance.....	4-15
Administering a Local Cache Vault .....	4-18

## Overview

Windchill content replication increases the productivity of Windchill users by reducing their time to access content data. The users access content data stored on more rapidly accessible external vaults known as replica vaults. Replica vaults store content data that has been replicated from slower external vaults or from the Windchill 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 site is a group of hosts with one URL. For the purpose of content replication, a site can play the role of master site, replica site, or both. When a site is playing the role of a master site, content can be replicated from Binary Large Object (LOB) storage, 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 vault/folder configuration information for each of its replica sites. Replica sites retrieve vault configuration information on startup or an update of the information is pushed from the master site on its startup or sent explicitly by the master site administrator.

A replica site is meant to provide Windchill users with local access to content data in replicated vaults. The data in each replicated vault can come from only one master site, and attempts to disregard this rule could 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 have a connection to an DBMS. 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 upload/download content. The advantage of running in this lightweight mode is that no database instance is needed and most of Windchill services are shut down, providing for easier maintenance, improving performance and startup time.

## Data Security

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. By using the master site's public key, the replica service verifies that all the URLs from which to download originate from the master site. The same checking procedure is used during the replication process to ensure 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 ensure correct key validation. A difference between the clocks of more than five minutes may

prevent 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 the incomplete designs for the part, so two different vaults would be appropriate:

1. WTPart, all-states, collab-domain > Vault\_on\_site1
2. WTPart, complete, collab-domain > Vault\_on\_site2

These rules provide engineers with local access to the content for all life cycle states of the part, and sales personnel with local access to complete parts.

**Note:** Windchill domains can be created in a hierarchical fashion, with some domains being children of other domains. However, 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.

## Scheduling

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 an item needs to be created for each replica vault. By creating a schedule item, you can ensure 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 copy of content from master to replica occurs only once unless replication is reset. 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 another site. 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.

## Vaults and Folders

When a Windchill user creates information, such as a part or document, content files can be associated with that item. The file vaulting feature of Windchill allows you to specify whether, for a particular type of item in a specific life-cycle state, content files should be stored with the item in the Windchill database or stored in an external container called a vault.

Each vault contains folders, which correspond to storage locations (for example, directories) on the host system. Based upon the Windchill administrator-defined vaulting rules, an uploaded file is stored in the file system location represented by the vault and folder.

For content replication (lightweight and full-scale), remote vaults and folders must be created on the replica site.

The folder must be mounted to the hosts. A mount is the association between a folder and a host. When you create or update a mount, you specify a storage location on a host system.

## Two Types of Content Replication Sites

There are two types of replication (replica) sites to allow you to maximize performance and the use of resources: the lightweight replica site and the full-scale replica site.

- Lightweight replica site -- A partial, dedicated Windchill installation, without database access, which can only manage replicated content copied from other Windchill sites or content uploaded to the local cache.
- Full-scale replica site -- A complete, independent Windchill installation, with its own database, that can serve as a replica site for another Windchill installation by storing some of its file contents.

For information on installing and configuring a content replication site, refer to the *Windchill Installation and Configuration Guide -- Windchill*.

## Modifying the Local Site

Starting file vault service generates a local site named Master, and modifying the name or URL of that site is possible, but not normally required.

When file vault service starts, a default local site is automatically generated, so you do not need to create a local site. You can modify the parameters of the automatically generated local site. The automatically generated master site has the name *Master* and its URL is the value of the property `wt.httpgw.url.anonymous` in the `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 ensures that the automatically generated site labeled **This Installation** can continue serving its role in the event of a change in the value of the property `wt.httpgw.url.anonymous` in the `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 the name or URL data for any site, but you should ensure that all folders accessible via the old URL are accessible via the new URL and that these folders are properly mounted to all hosts of the site and these hosts' names are correct.

## 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.j  
ks com.ptc.foo.jar
```



## Editing Content Replication Properties

Editing properties 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 must add or edit properties with the `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 `wt.properties` files 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.

Property	Master	Replica	Description
<code>wt.fv.master.verbose</code> Properties	X		Print out master service properties on method server start up
<code>wt.fv.master.verbose</code>	X		Verbose setting for master service. Print out to Method Server. log
<code>wt.fv.master.log.enabled</code>	X		Enable logging of master service verbose information to a file specified in <code>wt.fv.master.log.filename</code>
<code>wt.fv.master.log.append</code>	X		Append new logging to the end of a file specified in <code>wt.fv.master.log.filename</code>
<code>wt.fv.master.log.filename</code>	X		Log master service verbose information

wt.fv.revaultQuerySize	X		Specify maximum number of items to be replicated and retrieved in one query to database
wt.fv.replica.verbose		X	Print out replica service properties on method server start up
wt.fv.replica.log.enabled		X	Enable logging of replica service verbose information to a file specified in wt.fv.replica.log.filename
wt.fv.replica.log.append		X	Append new logging to the end of a file specified in wt.fv.replica.log.filename
wt.fv.replica.log.filename		X	Log replica service verbose information
wt.fv.replicationFileSizeThreshold	X		Specify the minimum size of file to be eligible for replication
wt.fv.master.moveItemsBetweenReplicaVaults	X	X	Specify one copy of content on replica site.
wt.fv.master.millisecsToWait	X		Specify the duration of the delay before the master site resends replica configuration information.
wt.fv.master.siteConfigDeliveryAttempts	X		Specify the number of times the master site attempts to resend the replica configuration information.

**Note:** The wt.fv.replicationFileSizeThreshold may not appear in the file, but it is nonetheless existing and effective. For an explanation of this property, see [Content Replication and Windchill Visualization Service](#) on page 4-13.

## About the moveItemsBetweenReplicaVaults Property

When content replication is configured, the replication process from the master site to the replica sites is performed in several steps.

Before the delivery of the items to the remote sites actually occurs, the following two queries are involved:

- Windchill checks which items need to be moved between replica vaults. This may occur for some of the following reasons:
  - The corresponding WObject has been moved from one domain to another.
  - Its life cycle state has changed.
  - The replication rules have been modified.
- Windchill checks which items need to be replicated from the master to the replica site(s). Only items that have been replicated are moved. Those items created with the preference set to replica are not moved unless they also exist on the master.

The first query can be very time consuming. The property, wt.fv.master.moveItemsBetweenReplicaVaults, allows administrators to turn the first query off.

If the wt.fv.master.moveItemsBetweenReplicaVaults property is set to True, it allows copying between vaults on the same replica site. Set the property to False to improve site performance.

**Note:** PTC recommends one vault per replica server for content replication and two vaults per replica server for upload caches (only one of them is actively used at any point in time).

## 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.

**Note:** 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.

Property	Sender	Receiver	Explanation
wt.intersvrcom.verbose	X	X	Set in installation. Do not alter. default: false
wt.intersvrcom.ultraLight	X	X	Set in installation. Do not alter. default: false
wt.intersvrcom.security.graceTimePeriod	X	X	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)
wt.intersvrcom.security.URL Authentication	X	X	Do not set to false unless in debug mode. default: true
wt.intersvrcom.security.useProxyForClients	X		This value must be set to true if the sender connects internet through a proxy. default: true
wt.wrmf.verbose	X		default: false
wt.wrmf.delivery.deleteDeliveredItem		X	If set to true, all delivered Shipping Item will be deleted from the database. default: true
wt.wrmf.transport.httptransport.supportInterruption	X		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
wt.wrmf.transport.outbox.pipe.<1 or 2 or 3>	X		Sets a transportation type. See the values of this property in the wt.properties file for the correspondence of integers with transportation types.
wt.wrmf.delivery.TrackingNumberGenerator	X		
wt.intersvrcom.transport.site	X		

# Out-of-the-box Background Queues for Content Replication

The following sections describe the queues that are established when your Windchill solution is installed.

## DeliveryStatusOnStartup

The DeliveryStatusOnStartup queue is used by content replication for the removal of failed or undelivered messages. The queue starts shortly after method server start up and works through all of the failed or undelivered messages.

References to the internal content replication messaging system are placed in the queue entries. If there are failed entries, this means that not all failed messages could be deleted. The failed entries can be removed from the queue. Next time the method server is restarted, all needed information will be gathered again.

There are two scenarios when the DeliveryStatusOnStartup queue is heavily used:

- When the master site method server is restarted after the replica site becomes unavailable during a content replication run.
- When the master site method server is restarted after it was restarted during a content replication run.

## PurgeScheduleHistoryQueue

The PurgeScheduleHistoryQueue is used by external vaulting and content replication to track and clean up information about the old (outdated) schedules for revaulting and content replication. The queue contains only one item at any time.

A failed queue entry is an indication that old schedule histories could not be removed.

Checking for failed entries in this queue is not needed. If the entry in the queue is in the failed state, it will be recreated during the next startup of the method server.

## 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.

Property	Description
<code>wt.queue.max.processQueues</code>	Sets the maximum number of process queues (24) that the queue service creates before throwing an exception. Default is 12.
<code>wt.queue.max.scheduleQueues</code>	Sets the maximum number of schedule queues (24) that the queue service creates before throwing an exception. Default is 12.

# 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
```

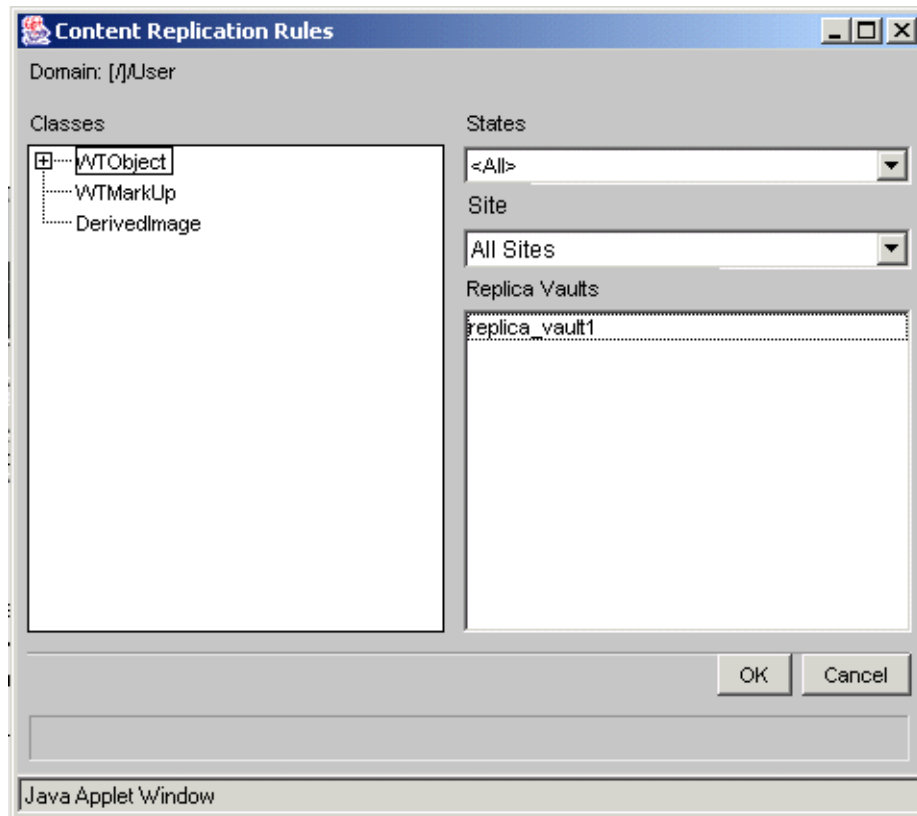
```
wvs.enabled=true
```

The default value of the 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** dialog box 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** dialog box 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** dialog box. In this dialog box, 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 menu, only the vaults for that site are displayed in the dialog box.

## Replication and Compression

By default, content replication uses compression. To stop the compression, you set the property `wt.intersvrcom.transport.site` to `false` in the `wt.properties` file.

**Note:** 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 to `false`:

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

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

```
wt.intersvrcom.transport.site.replica1.useGzip=false
```



**Note:** If the data transfer is within a LAN, then setting `wt.intersvrcom.transport.site.<Site Name>.useGzip` to true will require more time to transfer data. This is due to increased IO operations while zipping. You should therefore set the property to false within a LAN. Data transfers across LANs with the above property set to true is slower because of the data compression during transfer.

## Troubleshooting Replication

There are two log files for troubleshooting replication: `ReplicaSite.log` on the replica site and `MasterSite.log` on the master site.

These files are controlled by the following properties in the `wt.properties` file. You should edit these properties using the `xconfmanager`.

```
wt.fv.replica.log.filename=$(wt.logs.dir)$(dir.sep)ReplicaSite.log
wt.fv.master.log.filename=$(wt.logs.dir)$(dir.sep)MasterSite.log
```

## Improving Content Replication Performance

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

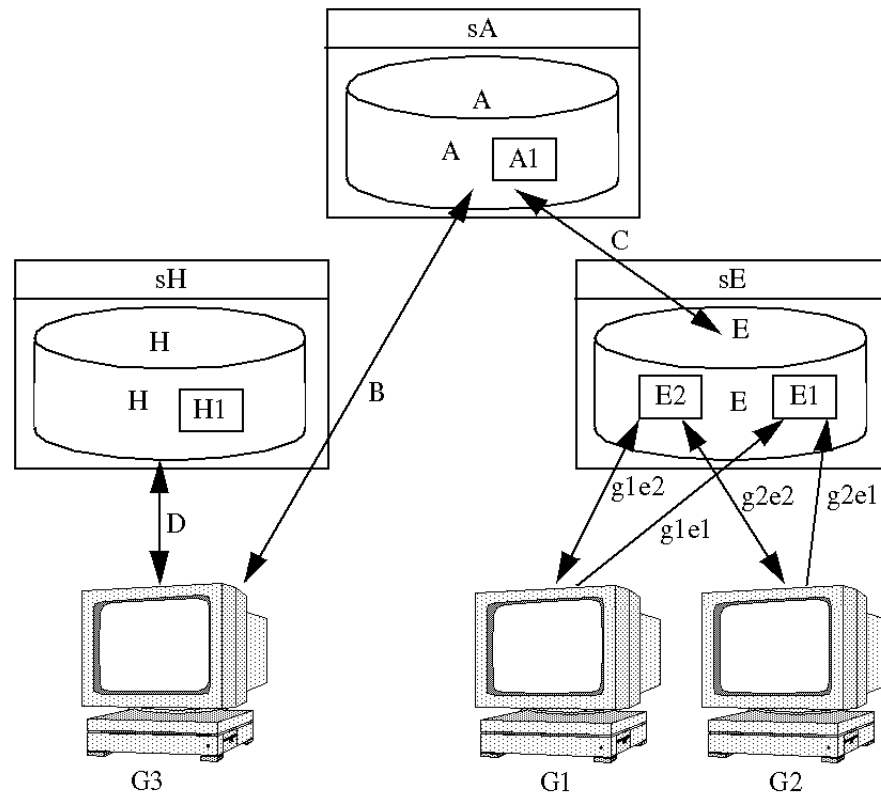
- **Local Upload** -- Places a user's uploaded content in the local cache vault as an intermediate location prior to transfer to a master site upon checkin.
- **Content Cache Server (CCS)** -- Creates a location for rapid access to frequently requested content data.

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.

## Administering a Local Cache Vault

You can specify one vault in a site to perform the local cache vault role in the replica site.

To enable the local cache function in a vault, select the **Designated for Cache** check box while creating or updating a vault in the replica site:

- When you are creating a vault, the check box 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 user's guide for your Windchill solution provides further information on accessing and setting user preferences.

## 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 [Scheduling](#) in this chapter for an explanation of scheduling.

## Removing Unreferenced Files from a Cache Vault on a Replica Site

An unreferenced file is a content file that no longer has a valid association to a Windchill item. For example, a file is no longer referenced when all items to which it was associated have been deleted, or when a user deletes the association to the file when updating the item.

The following procedure describes the process to remove files that are no longer referenced in Windchill from a cache vault on the replica site. This procedure will also remove references to replicated content that was uploaded to the replica site via an upload to replica cache. The procedure is not applicable to any existing replication rules for the vault. If any of the contents uploaded to the replica cache must permanently reside on the replica site (e.g., for the purpose of faster downloads, etc.), then replication rules for these objects must be created.

### To remove unreferenced files from a cache vault

1. Using the xconfmanager, on both the master and replica sites, set the following property in the wt.properties file:  
  
wt.fv.master.moveItemsBetweenReplicaVaults=false  
  
If the property was previously set to true, you must restart both the master and replica site now.
2. Delete all of the currently existing content replication schedules for the replica cache vault (that is, delete all of the existing schedules for existingReplicaCacheVault).
  - a. Open from the **Schedule Content Replication** window:
    - i. On the **Site** tab, click **Utilities**.
    - ii. Click **Replication Administrator**.
    - iii. Click **Schedule Content Replication**.
  - b. In the **Schedule Content Replication** window, select a schedule and click **Delete**.
  - c. Repeat the above step for all existing schedules.
3. Mark all existing folders of existingReplicaCacheVault as read-only.
  - a. In the **Vault Configuration** window, select a vault from the **Vault** list.
  - b. Select the **Read Only** check box.
  - c. Click **OK**.
4. Create as many folders as there were in the original vault and mount the folders. Make sure the folders are enabled and not read-only. These folders will accept content uploads while this operation is in progress. The steps to create each folder are as follows:
  - a. In the **Vault Configuration** window, click **File > New > New Folder** to display the **New Folder** dialog box.
  - b. Select a vault from the **Vault** list.  
  
**Note:** Do not select the **Read Only** check box as this will make the folder unavailable to store uploaded files.
  - c. Type a unique name for the folder in the **Name** box.
  - d. Click **OK**.
  - e. In the **Vault Configuration** window's left pane, expand a cabinet that holds folders, and select the folder's icon.

- f. Click **Object > Mount**.  
The **New Mount** dialog box appears.
  - g. Specify a path in the **Path** box.
  - h. Click **OK**.
  - i. With the new folder selected in the **Vault Configuration** window, click **Object > Toggle Enable**.
  - j. Repeat the above steps for each folder you need to create.
5. Create a new replica vault called tempReplicaVault. (Do NOT mark the vault as designated for cache.)
    - a. In the **Vault Configuration** window, click **File > New > New Vault** to display the **New Vault** dialog box.
    - b. Select the replica site server from the **Site** list.
    - c. Do not select the **Read Only** check box as this will make the vault unavailable to store uploaded/replicated files.
    - d. Type a unique name for the vault in the **Name** box.
    - e. Click **OK**.
  6. Move all of the original folders from existingReplicaVault to tempReplicaVault. Do not move the folders created in step 4.
  7. Broadcast the configuration to the replica site, where the vaults existingReplicaVault and tempReplicaVault reside.  
  
With the replica server selected in the **Vault Configuration** window, click **File > Broadcast Configuration**.
  8. Schedule an immediate/run-once content replication session for the vault tempReplicaVault.
    - a. Open from the **Schedule Content Replication** window:
      - i. On the **Site** tab, click **Utilities**.
      - ii. Click **Replication Administrator**.
      - iii. Click **Schedule Content Replication**.
    - b. In the **Schedule Content Replication** window, click **Create** to display the **Content Replication Scheduler** dialog box.
    - c. In the **Replica Vault** box at the top of the dialog box, select the vault that will receive the replicated content.
    - d. In the **Time and Frequency** groups, specify **Immediate-Once** replication.

- e. Click **OK**.

The **Schedule Content Replication** dialog box displays the new schedule item.

Wait for the schedule to complete. This content replication session will pull the content, previously uploaded to the replica cache and residing in one of the folders which was moved in step 6, to the master site.

9. In the wt.properties file, set the following property on both the master and replica sites using the xconfmanager utility:

```
wt.fv.master.moveItemsBetweenReplicaVaults=true
```

10. Restart the replica site and verify that it is running.

11. Restart the master site and verify that it is running.

12. Schedule an immediate/run-once content replication session for the vault existingReplicaCacheVault. (See step 8 above for instructions.)

Wait for the schedule to complete. This content replication session will copy all referenced contents from the vault tempReplicaVault to the vault existingReplicaCacheVault and store it in the folders created in step 4.

13. For each folder that belongs to the vault tempReplicaVault:

- a. Write down the mount path for the folder.
- b. Delete the folder from Windchill using the **Vault Configuration** window. (To access the **Vault Configuration** window, in the **External Storage Administrator** or the **Replication Administrator**, click **Vault Configuration**.)
- c. If folder deletion succeeds, it is safe to remove the physical directory on the disk to which the mount was pointing.

14. If the vault tempReplicaVault contains no more folders, delete the vault.

15. In the wt.properties file, set the following property on both the master and replica sites using the xconfmanager utility:

```
wt.fv.master.moveItemsBetweenReplicaVaults=false
```

16. Restart both the master site and replica site.

## Resetting Replication

Clicking **Object > Reset Replication** removes references to ALL previously replicated content for a target site. All content which has replication rules defined for it will be replicated to the site at the next run of content replication. The replica site may still contain content which has been uploaded to its cache. References to such content are NOT removed. This option should be used with extreme caution before the site is scheduled to be unregistered.



## Resetting Replication for Undelivered Items

When replication finishes, the message **Delivery Complete** appears. This message means that the master site has successfully sent a message to the replica site. The message does not guarantee that replication was executed on the replica site.

If a **Failed to Replicate** message appears, then you should reset the replication for undelivered items for the target vault and schedule the replication again.

Selecting **Reset Replication Undelivered Items** removes references to the replicated content which did not replicate properly during the previous runs of content replication.

These references are deleted automatically after a period of 30 days. If you want to re-replicate the **Failed to Replicate** content sooner than 30 days, this option may be used.

### To reset replication for undelivered items

1. Open the **Vault Configuration** window:
  - a. On the **Site** tab, click **Utilities**.
  - b. Click **Replication Administrator**.
  - c. Click **Vault Configuration**.
2. Select the target vault.
3. Click **Object > Reset Replication for Undelivered Items**.

## 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 Content Replication Properties](#) in this chapter for an explanation of the log file properties.



# 5

## Configuring External File Vaulting or Replication With FvLoader

Topic	Page
Overview of Configuration With FvLoader .....	5-2
Configuring External File Vaults or Rules.....	5-3
Supporting Replica Site Vaulting.....	5-3
Removing External File Vaulting Rules or Replication Rules.....	5-4
Listing Domains .....	5-5
Listing Vaulting Rules.....	5-7

## Overview of Configuration With FvLoader

PTC supplies a class that you can use as a utility. 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.

FvLoader can be used for the following tasks:

- 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.
- To list domains -- accomplished through command line. This is a two-step process. See [Listing Domains](#).
- To list vaulting policy rules -- accomplished through command line. You can use the output for batch deletion or recreation of policy rules. See [Listing Vaulting Rules](#).
- To support 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 the 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. Contact PTC's Global Services for information on implementing local replication. [Do not keep the CSV file in edit mode while executing FvLoader commands](#).

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_Path>
```

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 (if you type 0, the Remote Replica Site is displayed, and, if you type 1, then the Local Master site is displayed)
- R,vaultName,fullClassName,fullDomainPath,lifeCycleStateName -- Creates a file vaulting policy rule that concerns the following:
  - File vault with the name vaultName
  - 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.
  - Domain with the full external path fullDomainPath. For an explanation of where to get the fullDomainPath please see the section, [Listing Domains](#).
  - Life cycle state with the name lifeCycleStateName.
- VE,vaultName -- Enables the vault with the name vaultName.
- 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 cycle 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)

- Replica file vault with the name `replicaVaultName`
- 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.
- Domain with the full external path `domainPath`. For an explanation of where to get the `fullDomainPath` please see the section, [Listing Domains](#).
- Life cycle state with the name `lifeCycleStateName`
- `RemoveLocalR,vaultName,fullClassName,fullDomainPath,lifeCycleStateName`  
-- Removes an existing external vaulting rule (same arguments as for rule creation)
  - File vault with the name `vaultName`,
  - 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.
  - Domain with the full external path `fullDomainPath`. For an explanation of where to get the `fullDomainPath` please see the section, [Listing Domains](#).
  - 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:\>

```

### Piping Example:

```
C:\> java wt.fv.FvLoader - listContainers | tee c:\temp\cont.txt
```

## 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

[/wt.inf.container.OrgContainer=PTC/wt.inf.library.WTLibrary=
Windchill PDM]/
    ChangeItems

C:\>

```

## 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

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

###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:\>

```

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**

```

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

```

**Example: Create command for the same rule**

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

**Note:** Do not keep the CSV file in edit mode while executing FvLoader commands.



# 6

## Windchill Import and Export

Topic	Page
Overview .....	6-2
Context Considerations in Import and Export.....	6-4
Importing and Exporting: Supported Items List.....	6-8
Configuration Specification Settings.....	6-35
Import and Export of EPMDocuments.....	6-36
Exporting with the Export User Interface .....	6-37
Importing with the Import User Interface .....	6-42
Additional Export and Import Actions .....	6-46
Windchill Properties for Export and Import.....	6-47
Windchill Export Properties .....	6-47
Windchill Import Properties .....	6-48
Access Control for Export and Import .....	6-49

## Overview

Windchill Import and Export can assist you in moving Windchill Foundation & PDM content and metadata to and from Windchill Foundation & PDM, Windchill PDMLink, and Windchill ProjectLink sites by placing the data in JAR files. The JAR file contains XML files representing the item's metadata and content files.

Windchill Export allows you to compress the data in any of the following Windchill items into a JAR file:

- Items in Cabinets and Folders -- The content of items supported by Windchill Export that is located in cabinets and folders is compressed. Folders and cabinets are not supported, but for each supported item, 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 item 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 items: 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 item 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 items 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 will not import an item that already exists in the same namespace in the local Windchill database. Uniqueness is evaluated on the basis of its unique identifier and/or business entity. For most business items such as WTPart, WTDocument and EPMDocument instances, the unique identifier is known as the UFID (unique federated identifier) that is composed of the local ID, the domain, and the site. Different item types may have different uniqueness

identifiers, for example, an instance based attribute (IBA) or soft type definition item can be identified by its name and its path.

An item'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 item 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 window, or a policy or rule file is used to change either the UFID or the business identity of the import item.

Both the export and import processes can refer to mapping rule files that transform data in the JAR file. In addition context mapping files enable the specification of item 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 window during import or export. If this is not supplied from the window, 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 items within logical entities called containers. The container concept is used to separate items that belong to different working contexts.

This topic discusses the following in its sub-sections:

- Access to items and the import or export of items 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 items:

- WTPart, WTDocument, Reference Document, and End Item which is also known as WTProduct.

Any IBA values and IBA definitions must be exported with the item 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 items 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** window will be available to administrators of site, organization, product, and library contexts through the **Export/Import Manager** on the **Utility** tab. This window allows administrators to search within a context for items to export. The search returns items created within the context container. To export data from a project context, the user should use the export action from the **Project Details** page.



**Note:** In order to export an item, you must have READ access to the item and all its attributes. For example, you must have READ access to the Team of an item, where the Team template name would be one of the exported attributes.

## 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 action 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 items 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 items 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 the import action at this level:

- Only business items 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 items in the target set. When the import into folder action is performed, all foldered items 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 Items with Context Mapping Files

Normally, all items are imported into the target context where the import process launched. If you want to override this behavior, in Solution Name you can use context mapping. Due to security considerations, in Solution Name, importing is always to a single project container.

**Note:** In Solution Name, business items are contained in the classic container called Windchill PDM.

For more information on containers, refer to the chapter, Administering Containers, in the *Windchill Business Administrator's Guide*.

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:

```
<?xml version="1.0" encoding="UTF-8" ?>
<container-info>
  <container>
    <source-container>Original containerReference of the item at
      the export site</source-container>

    <target-container>containerReference of the context where
      the item must be imported to at the import site</target-
      container>
  </container>

  <container>
    <source-container>Original containerReference of the item at
      the export site</source-container>

    <target-container>containerReference of the context where
      the item must be imported to at the import site</target-
      container>
  </container>

  <container>
    <source-container>Original containerReference of the item at
      the export site</source-container>

    <target-container>containerReference of the context where
      the item must be imported to at the import site</target-
      container>
  </container>
</container-info>
```

There can be more than one <container> elements in the mapping file, as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<container-info>
  <container>
    <source-
      container>wt.inf.container.DefaultOrgContainer=DefaultOrg/w
      t.inf.container.ClassicContainer=Windchill PDM</source-
      container>

    <target-
      container>wt.inf.container.OrgContainer=Windchill_RD/wt.inf
      .library.WTLibrary=Windchill PDM</target-container>
  </container>
</container-info>
```

## Importing and Exporting: Supported Items List

This section includes tables that list the items that are supported or not supported by import/export functionality. Import/export technology is not solely supported by the Import/Export Manager. Refer to the Comments column for additional information on the window that may support import/export.

The following tables are included:

- [Supported Windchill ProjectLink Items](#)
- [Supported Windchill PDMLink Items](#)
- [Supported Windchill Foundation & PDM Items](#)

Refer to the appropriate table to determine whether an item is supported by loader technology and/or import/export.

## Supported Windchill ProjectLink Items

The following table lists the Windchill ProjectLink items that can be loaded or imported/exported. The column, Can Be Exported?, specifies whether it is possible to export this item to a file using the Windchill Import/Export Manager, the Life Cycle/Workflow Manager, as part of a template, or in some other manner.

Windchill ProjectLink Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Folder</b>			
Folder structure	Yes	Yes	
Folder access rules	Yes	Yes	
Links	Yes	Yes	Parts and documents supported only
Imported attachments	No	No	
<b>Document</b>			
Document metadata	Yes	Yes	
Template Document	No	No	Exported as WTDDocument
Document content	Yes	Yes	
Document forum	No	No	
Document notebooks	No	No	Also called references
Document meetings	No	No	
Document templates	Yes	Yes	
Document routings	No	No	Only the name of the current routing for the document and the state of the document are exported with the document. It is assumed that a life cycle template with that name exists in the target system.

Windchill ProjectLink Item	Supported by a Loader?	Can Be Exported?	Comments
Document related tasks	No	No	
Document change history	No	No	
Document access rules	Yes	Yes	Role based access rules
<b>Part</b>			
Part metadata	Yes	Yes	
Part viewable	No	No	
Part Markup	No	No	
Part thumbnail	No	No	
Part EPM docs	No	Yes	
Part structure	Yes	Yes	
Part forum	No	No	
Part notebooks	No	No	Also called references
Part meetings	No	No	
Part templates	No	No	
Part routings	No	No	
Part related tasks	No	No	
Part change history	No	No	
Part access rules	No	Yes	
<b>Project</b>			
Project files	Yes	Yes	
Project metadata	Yes	Yes	Not all project metadata is exported
Project forum	No	No	
Project template	Yes	Yes	

Windchill ProjectLink Item	Supported by a Loader?	Can Be Exported?	Comments
Project subproject list	No	No	
Project change history	No	No	
Project life cycle templates	Yes	No	
Project life cycle instance	No	No	
Project workflow templates	Yes	No	
Project workflow instances	No	No	
<b>Team</b>			
Team metadata	Yes	Yes	
Team roles (including dynamic)	Yes	Yes	
Team members	Yes	Yes	
<b>Activity</b>			
Activity metadata	Yes	Yes	
Activity deliverables	Yes	Yes	
Activity forum	No	No	
Activity resources	Yes	Yes	
Activity tasks	No	No	
Activity notebooks	No	No	Also called references
<b>Summary activity</b>			
Summary metadata	Yes	Yes	
Summary child links	Yes	Yes	
Summary forum	No	No	
Summary notebooks	No	No	Also called references
Summary resources list	Yes	Yes	
Summary task list	No	No	

Windchill ProjectLink Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Subproject</b>			
Subproject metadata	No	No	
Subproject project notebooks	No	No	Also called references
<b>Milestone</b>			
Milestone metadata	Yes	Yes	
Milestone deliverable links	Yes	Yes	
Milestone forum	No	No	
Milestone notebooks	No	No	Also called references
<b>Meeting</b>			
Meeting metadata	No	No	
Meeting discussion	No	No	
Meeting action items	No	No	
Meeting notebooks	No	No	Also called references
Meeting subject list	No	No	
<b>Discussion</b>			
Discussion topics	No	No	
Discussion postings	No	No	
Discussion posting links	No	No	
Discussion posting attachments	No	No	
<b>User Notebook</b>			
Reference folder structure	No	No	
Reference links	No	No	
Reference attachments	No	No	



Windchill ProjectLink Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Action item</b>			
Action item metadata	No	No	
Action item forum	No	No	
Action item notebooks	No	No	Also called references
Action item meeting links	No	No	

## Supported Windchill PDMLink Items

The following table lists the Windchill PDMLink items that can be loaded or imported/exported. The column, Can Be Exported?, specifies whether it is possible to export this item to a file using the Windchill Import/Export Manager, the Life Cycle/Workflow Manager, as part of a template, or in some other manner.

**Note:** \* indicates items that are not supported in Windchill PDMLink.

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Document</b>			
Metadata/attributes	Yes	Yes	
Multiple versions and iterations	Yes	Yes	
Primary content file	Yes	Yes	Loader is LoadContent
Primary content URL	Yes	Yes	Loader is LoadContent
No primary content	Yes	Yes	
Secondary content file(s)	Yes	Yes	Loader is LoadContent
Secondary content URLs	Yes	Yes	Loader is LoadContent
Structure	No	No	
Config spec	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Baseline	No	No	
Doc-Doc reference links	No	No	
Doc-Part reference links	Yes	Yes	Loader is LoadPart
Doc-Part described by links	Yes	Yes	Loader is LoadPart
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	
Location (context)	No	No	Imported/loaded into a target container
Associated change objects	No	No	
Associated meetings	No	No	
Associated discussions	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated life cycle	Yes	Yes	<p>Associated life cycles are supported, but the target system must have the corresponding Life Cycle Template.</p> <p>When a document is exported, the name of the life cycle template it is hooked to is exported. The actual life cycle template is not exported. The assumption is that the target system has a life cycle template with the same name so that the document gets linked with the life cycle template.</p> <p>You can directly export/import a life cycle template from the Life Cycle Administrator, but this is independent of documents/parts etc.</p>
Associated processes	No	No	
Associated template	No	No	
Subscriptions	No	No	
Template documents	Yes	Yes	Loaded as part of a Container Template
<b>Part</b>	<b>LoadPart</b>		
Metadata	Yes	Yes	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Multiple versions and iterations	Yes	Yes	
Associated viewable (Representation)	Yes	No	
Associated markup	No	No	
Associated thumbnail	No	No	
Associated EPM docs	No	Yes	
Product Structure	Yes	Yes	
Config Spec	No	No	
Views	Yes	Yes	Views are supported, but the target system must have the corresponding View item
Baselines	No	No	
Effectivity	Yes	No	Loader is LoadEffectivity
Substitutes/Alternatives	No	No	
Reference Designators	Yes	Yes	
Line Numbers	Yes	No	
Associated Described by Docs	Yes	Yes	
Associated Reference Docs	Yes	Yes	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associate life cycle	Yes	Yes	<p>Associated life cycles are supported, but the target system must have the corresponding Life Cycle Template.</p> <p>When a document is exported, the name of the life cycle template it is hooked to is exported. The actual life cycle template is not exported. The assumption is that the target system has a life cycle template with the same name so that the document gets linked with the life cycle template.</p> <p>You can directly export/import a life cycle template from the Life Cycle Administrator, but this is independent of documents/parts etc.</p>
Associated processes	No	No	
Associated annotations	No	No	
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Location (context)	No	No	Imported/loaded into a target container
Associated change objects	No	No	
Save As history	No	No	
Associated IBAs	Yes	Yes	Loader is LoadValue
Discussions	No	No	
Subscriptions	No	No	
Meetings	No	No	
Part templates	No	No	
<b>EPM Docs</b>			
Metadata	No*	Yes	
Multiple versions and iterations	No*	Yes	
Associated Parts	No*	Yes	
Associated Products	No*	Yes	
Structure/Links	No*	Yes	
Latest	No*	No	
As stored	No*	No	
Baselines	No*	No	
(Member, Reference, Contained In Variant links)	No*	Yes	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associate life cycle	No*	Yes	<p>Associated life cycles are supported, but the target system must have the corresponding Life Cycle Template.</p> <p>When a document is exported, the name of the life cycle template it is hooked to is exported. The actual life cycle template is not exported. The assumption is that the target system has a life cycle template with the same name so that the document gets linked with the life cycle template.</p> <p>You can directly export/import a life cycle template from the Life Cycle Administrator, but this is independent of documents/parts etc.</p>
Associated processes	No*	No	
Version history	No*	No	
Iteration history	No*	No	
Life cycle history	No*	No	
Location (context)	No*	No	Imported into a target container
Associated change objects	No*	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Save As history	No*	No	
Discussions	No*	No	
Subscriptions	No*	No	
Meetings	No*	No	
EPMDoc template	No*	No	
<b>Serialized Parts</b>			
Metadata	No	Yes	
Serialized Part instances	No	No	
<b>Product/Library container</b>	<b>LoadContainer</b>		
Metadata	Yes	Yes	Imported/loaded into a Container Template
Container content (all objects)	No	No	
Container team	Yes	Yes	Imported/loaded as part of a Container Template
Access control rules	Yes	Yes	Imported/loaded as as part of a Container Template
Container template	Yes	Yes	
Life cycle templates	Yes	No	
Workflow templates	Yes	No	Loader is LoadWorkflow
Version scheme	No	No	
Numbering sequence	No	No	
<b>Product/End Item</b>	<b>LoadPart</b>		
Metadata	Yes	Yes	
Multiple versions and iterations	Yes	Yes	



Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated viewable (Representation)	Yes	No	
Associated markup	No	No	
Associated thumbnail	No		
Associated EPM docs	No	Yes	
Structure	Yes	Yes	
Config Spec	No		
Views	Yes	Yes	Views are supported, but the target system must have the corresponding View item
Baselines	No	No	
Effectivity	Yes	No	
Substitutes/Alternatives	No	No	
Reference Designator	Yes	Yes	
Line Number	Yes	No	
Product Configurations	No	No	
Product Instances	No	No	
Associated annotations	No		
Associated Described by Docs	Yes	Yes	
Associated Reference Docs	Yes	Yes	
Associate life cycle	Yes	Yes	
Associated processes	No	No	
Version history	No	No	
Iteration history	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Life cycle history	No	No	
Location (context)	No	No	Imported/loaded into a target container
Associated change objects	No	No	
Save As history	No	No	
Associated IBAs	Yes	Yes	
Discussions	No	No	
Subscriptions	No	No	
Meetings	No	No	
End Item templates	No	No	
<b>Change Issue/Problem Report</b>			
Metadata*	No	No	
Associated life cycle*	No	No	
Associated processes*	No	No	
Associated content/attachments*	No	No	
Associated change request*	No	No	
Life cycle history*	No	No	
Affected products*	No	No	
Affected data*	No	No	
Subscriptions*	No	No	
<b>Change Request</b>			
Metadata*	No	No	
Associated life cycle*	No	No	
Associated processes*	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated content/attachments*	No	No	
Associated change issues*	No	No	
Associated change investigation*	No	No	
Associated change proposal*	No	No	
Associated change order*	No	No	
Life cycle history*	No	No	
Affected products*	No	No	
Affected data*	No	No	
Subscriptions*	No	No	
<b>Change Order/ECN</b>			
Metadata*	No	No	
Associated life cycle*	No	No	
Associated processes*	No	No	
Associated content/attachments*	No	No	
Associated change requests*	No	No	
Associated change activities*	No	No	
Life cycle history*	No	No	
Effectivity*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
<b>Change Activity</b>			
Metadata*	No	No	
Associated content/attachments*	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated original data version*	No	No	
Associated new data version*	No	No	
Associated change order/ECN*	No	No	
Life cycle history*	No	No	
Effectivity	No	No	
Discussions	No	No	
Subscriptions	No	No	
<b>Change Investigation</b>			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated analysis activity*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
<b>Change Proposal</b>			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated analysis activity*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
<b>Analysis Activity</b>			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated data*	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated change proposals*	No	No	
Associated change investigations*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
<b>Worklist</b>			
Tasks*	No	No	
Metadata*	No	No	
Associated change objects*	No	No	
Associated processes*	No	No	
Associated Data*	No	No	
<b>Cabinets &amp; Folders</b>			
Metadata	Yes	Yes	Imported/loaded as part of a Container Template
Structure	Yes	Yes	Imported/loaded as part of a Container Template
Associated domains	Yes	Yes	Associated domains are supported but the target system must have the corresponding Domain
Access rules	Yes	Yes	Imported/loaded as part of a Container Template
Content (all objects)	N/A	Yes	Only parts, documents, and CAD documents are supported by import/export

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Links	N/A	N/A	
<b>Meeting</b>			
Metadata	No	No	
Discussion	No	No	
Action items	No	No	
References	No	No	
Subject list	No	No	
<b>Discussion</b>			
Topics	No	No	
Postings	No	No	
Posting links	No	No	
Posting attachments	No	No	
<b>Organization</b>	<b>LoadContainer</b>		
Metadata	Yes	No	
<b>User</b>	<b>LoadUser</b>		
Metadata	Yes	No	
<b>Group</b>	<b>LoadUser</b>		
Metadata	Yes	No	
Members	Yes	No	
<b>Domain</b>	<b>LoadUser</b>		Requires createDomain method
Metadata	Yes	No	
Structure	Yes	No	
Access control rules	Yes	No	
Indexing policies	No	No	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Notification policies	No	No	
<b>Team Templates</b>	<b>LoadTeam</b>		
Metadata	Yes	No	
Roles (including dynamic)	Yes	No	
<b>Team Instances</b>			
Metadata	No	No	
Roles (including dynamic)	No	No	
Members	No	No	
Groups	No	No	
<b>Life Cycle Templates</b>	<b>LoadLifecycle</b>		
Metadata	Yes	Yes	Import/export is possible using the Life Cycle Administrator
States (Phases)	Yes	Yes	
Roles & Participants	Yes	Yes	Loader is LoadProject
Access control rules	Yes	Yes	

Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
Associated process	No	No	<p>The name of the workflow template is imported/export, not the template. It is assumed that the target system has a workflow with the same name.</p> <p>The LoadLifecycle loader also only loads the template and the names of workflow templates for phases and gates.</p> <p>LoadLifecycle is typically run after LoadWorkflow, so the associations between a phase and a workflow template are created on the LoadLifecycle load.</p>
Promotion criteria	No	No	
Iteration history	No	No	
<b>Workflow Templates</b>	<b>LoadWorkflow</b>		
Metadata	Yes	Yes	Import/export is possible using the Workflow Administrator
Process	No	No	
<b>Workflow Instances</b>			
Metadata	No	No	
Process status	No	No	
Associated data	No	No	



Windchill PDMLink Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Report Templates</b>			
Metadata	Yes	No	
<b>Miscellaneous Admin</b>			
Preference settings	Yes	No	
Replication policies	No	No	
Type definitions	Yes	Yes	Loader is LoadValue
Attribute definitions	Yes	Yes	Loader is LoadValue

## Supported Windchill Foundation & PDM Items

The following table lists the Windchill Foundation & PDM items that can be loaded or imported/exported. The column, Can Be Exported?, specifies whether it is possible to export this item to a file using the Windchill Import/Export Manager, the Life Cycle/Workflow Manager, as part of a template, or in some other manner.

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
<b>Document</b>	<b>LoadDoc</b>		
Metadata/attributes	Yes	Yes	
Multiple versions and iterations	Yes	Yes	
Primary content file	Yes	Yes	Loader is LoadContent
Primary content URL	Yes	Yes	Loader is LoadContent
No primary content	Yes	Yes	
Secondary content file(s)	Yes	Yes	Loader is LoadContent

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
Secondary content URLs	Yes	Yes	Loader is LoadContent
Structure	No	No	
Config spec	No	No	
Baseline	No	No	
Doc-Doc reference links	No	No	
Doc-Part reference links	Yes	Yes	Loader is LoadPart
Doc-Part described by links	Yes	Yes	Loader is LoadPart
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	
Associated change objects	No	No	
Associated life cycle	Yes	Yes	Life cycle attribute is supported, but the target system must have the corresponding Life Cycle Template
Associated processes	No	No	
<b>Part</b>	<b>LoadPart</b>		
Metadata	Yes	Yes	
Multiple versions and iterations	Yes	Yes	
Associated viewable (Representation)	Yes	No	
Associated markup	No	No	
Associated thumbnail	No	No	
Associated EPM docs	No	Yes	

<b>Windchill Foundation &amp; PDM Item</b>	<b>Supported by a Loader?</b>	<b>Can Be Exported?</b>	<b>Comments</b>
Product Structure	Yes	Yes	
Config Spec	No	No	
Views	Yes	Yes	Views are supported, but the target system must have the corresponding View item
Baselines	No	No	
Effectivity	Yes	No	Loader is LoadEffectivity
Substitutes/Alternatives	No	No	
Reference Designators	Yes	Yes	
Line Numbers	Yes	No	
Associated Described by Docs	Yes	Yes	
Associated Reference Docs	Yes	Yes	
Associate life cycle	Yes	Yes	Life cycle attribute is supported, but the target system must have the corresponding Life Cycle Template
Associated processes	No	No	
Associated annotations	No	No	
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	
Associated change objects	No	No	
Save As history	No	No	

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
Associated IBAs	Yes	Yes	Loader is LoadValue
<b>EPM Docs</b>			
Metadata	No	Yes	
Multiple versions and iterations	No	Yes	
Associated Parts	No	Yes	
Associated Products	No	Yes	
Structure/Links	No	Yes	
Latest	No	No	
As stored	No	No	
Baselines	No	No	
(Member, Reference, Contained In Variant links)	No	Yes	
Associate life cycle	No	Yes	Life cycle attribute is supported, but the target system must have the corresponding Life Cycle Template
Associated processes	No	No	
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	
Associated change objects	No	No	
<b>Serialized Parts</b>			
Metadata	No	Yes	
Serialized Part instances	No	No	

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
Product/End Item	LoadPart		
Metadata	Yes	Yes	
Multiple versions and iterations	Yes	Yes	
Associated viewable (Representation)	Yes	No	
Associated markup	No	No	
Associated thumbnail	No		
Associated EPM docs	No	Yes	
Structure	Yes	Yes	
Config Spec	No		
Views	Yes	Yes	Views are supported, but the target system must have the corresponding View item
Baselines	No	No	
Effectivity	Yes	No	
Substitutes/Alternatives	No	No	
Reference Designator	Yes	Yes	
Line Number	Yes	No	
Product Configurations	No	No	
Product Instances	No	No	
Associated annotations	No		
Associated Described by Docs	Yes	Yes	
Associated Reference Docs	Yes	Yes	

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
Associate life cycle	Yes	Yes	
Associated processes	No	No	
Version history	No	No	
Iteration history	No	No	
Life cycle history	No	No	
Associated change objects	No	No	
Associated IBAs	Yes	Yes	
<b>Cabinets &amp; Folders</b>			
Metadata	Yes	Yes	Loaded/exported as part of a Container Template
Structure	Yes	Yes	Loaded/exported as part of a Container Template
Associated domains	Yes	Yes	Domain attribute is supported but the target system must have the corresponding Domain
Access rules	Yes	Yes	Loaded/exported as part of a Container Template
Content (all objects)	N/A	Yes	Only parts, documents, and CAD documents are supported
Links	N/A	N/A	
<b>User</b>	<b>LoadUser</b>		
Metadata	Yes	No	
<b>Group</b>	<b>LoadUser</b>		

Windchill Foundation & PDM Item	Supported by a Loader?	Can Be Exported?	Comments
Metadata	Yes	No	
Members	Yes	No	
<b>Domain</b>	<b>LoadUser</b>		Requires createDomain method
Metadata	Yes	No	
Structure	Yes	No	
Access control rules	Yes	No	
<b>Misc Admin</b>			
Preference settings	Yes	No	
Replication policies	No	No	
Type definitions	Yes	Yes	Loader is LoadValue
Attribute definitions	Yes	Yes	Loader is LoadValue; exported with IBAHolder items, not separately

## 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 items 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 items 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 item with new identities
- Substitute for an existing item
- Ignore item

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 items, including:

- EPMMemberLink
- EPMReferenceLink
- EPMVariantLink
- EPMContainedIn
- 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 item, are an administrator of the item'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** 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 item/folder action list.

## Exporting from the Export Window

To display the **Export** window, perform either of the following two sets of actions:

1. Click **System Administration** on the **Windchill** home page to display the **System Administration** page. Click **Import/Export Manager**.

Or

Click **Site Map** on the **Windchill** home page. In the System Administration section of the **Site Map**, click **Import/Export Manager**.

2. Click **Export** to display the **Export** window.

Export

Source Context: "Windchill PDM"

Export File Name:  Browse...

Export Rule File:  Browse...

☐ Export Policy File :  Browse...

☒ Export Action : None

☐ Detailed Log

Configuration Specification

Set Config Spec

Objects for Export

Objects

Add... Delete

Filters

Add... Delete

Export Status Log

Preview Submit Save log... Exit Cancel Help

Java Applet Window

3. 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.
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** option 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.xml`

- Select the **Export Action** option and then select from the export action drop-down list (actions will be applied to all items 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 item will become checked out by you. The software attempts to check out items that are necessary to describe an item 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. Set the configuration specification in the **Configuration Specification** section of the window to specify a configuration specification, baseline, or effectivity for the exported item. Setting the configuration specification is optional.

Set Configuration Specification for navigating Product Structure

☒ Latest View: [dropdown] State: [dropdown]  
☒ Include parts in my personal cabinet

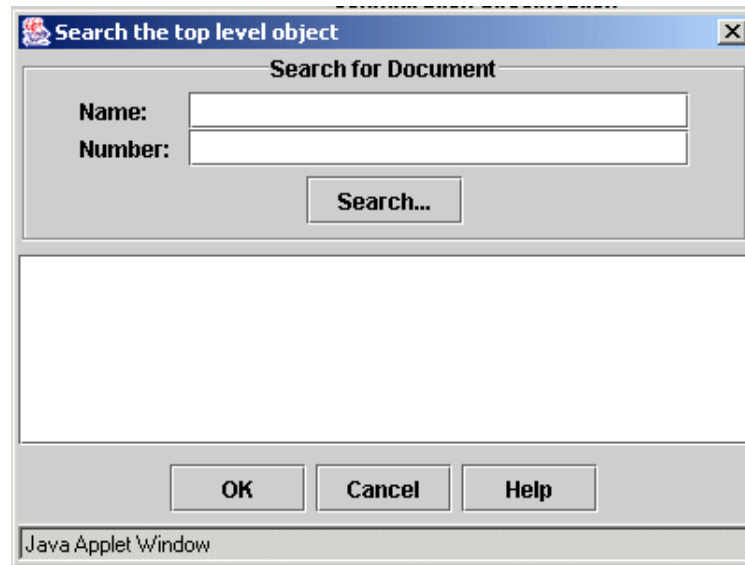
☐ Baseline Name: [text field] Search...

☐ Effectivity View: [dropdown]  
Effective Date: 10/14/2003  
Effectivity Context: [text field] Search...  
Type: [text field]  
Value: [text field]

OK Cancel Help

Java Applet Window

7. Click **Add** in the **Objects** section of the window to select data for export. You can remove items from the list by selecting them and clicking **Delete**. You select a type of item and then display a window for selecting the item. The following graphic shows the appearance of the window if you selected a document as the type of item to add.



8. Click **Add** in the **Filters** section of the window to specify a time period that defines the items for export. Adding filters is an optional step. You can filter items 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.

**Note:** If you have multiple filters for filtering by different time ranges, it should be noted that the filters combine together and may collectively filter all times.

9. If you need to, click **Preview** to understand what will be exported. With **Detailed Log** not checked, you can see how many items 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 the *Windchill System Administrator's Guide* for information that can help in resolving conflicts.

## Exporting an Item 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 items for export.

1. Navigate to the properties page of the item 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 item 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** option 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.xml`
  - Select the **Export Action** option and then select from the export action drop-down list (actions will be applied to all items 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 item will become checked out by you. The software attempts to check out items that are necessary to describe an item 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 item.
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 items for export. Adding filters is an optional step. You can filter items 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.

**Note:** If you have multiple filters for filtering by different time ranges, it should be noted that the filters combine together and may collectively filter all times.

9. If you need to, click **Preview** to understand what will be exported. With **Detailed Log** not checked, you can see how many items 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 to 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 items from various source contexts. Then, if an import item exists in the target database, import policies or import actions selected from the **Import** window are applied to determine how that item is modified.

**Note:** The Access Control List (ACL) applies to import operations. For example, a user without Revise privileges for a particular item type cannot defeat ACL control by using the import action, **Import as new version**. If an import attempt specifies an item 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 items 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 items 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 items in the target set. When the import into folder action is performed, all foldered items are placed into the target folder.

To display the **Import** window, perform either of the following two sets of actions:

1. Click **System Administration** on the **Windchill** home page to display the **System Administration** page. Click **Import/Export Manager**.

Or

Click **Site Map** on the **Windchill** home page. In the **System Administrator** list on the **Site Map** click **Import/Export Manager**.

2. 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.

Import

Default Target Context: "Windchill PDM"

Import File Name:  Browse...

Import Settings

Import Rule File:  Browse...

Context-mapping file:  Browse...

☐ Import Policy File:  Browse...

☒ Import Action: Default

☐ Resolve Overridable Conflicts

☐ Detailed Log

Import Status Log

Preview Submit Save log... Exit Cancel Help

3. 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.

4. 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. Specifying a mapping rule file is optional.
5. You may specify a context mapping file in the local file system to identify into which target context the import file items are placed. If you do not specify a context mapping file, items 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 Items with Context Mapping Files](#).
6. You may optionally specify import policies by one of the two following methods.
  - Select the **Import policy File** option 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.xml.
  - Select the **Import Action** option and then select from the import action drop-down list (actions will be applied to all items in the import file):
    - **Default** -- If the import item matches the full item identity of an existing database item, the existing item is picked and no import takes place. If the import item 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 item is newer than the existing db item, the import process will create an item with the next available iteration on the latest iteration. If the item does not exist in the target database, applying this action will create a new item. If you are sure that a new iteration should be created for all items that can be iterated, this option is appropriate.
    - **Import as new version** -- If there is a version of the import item in your site's database, the import process will create an item with the next available version and the first iteration. Otherwise, a new item will be created. If you are sure that a new version should be created for all items that can be versioned, this option is appropriate.
    - **Import as checked out** -- If there is a version of the import item in your site's database, the import process will create a checked out (working) copy of the existing item with the same version as in the import XML file. The newly imported item will be kept checked out until you check it in. If you are sure that all existing items should be checked out for all items 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 item without iterating the item.
  - Update checked out item in place -- This option will replace the content, attributes, and links of the checked out item.
7. Select or clear the **Resolve Overridable Conflicts** check box. This check box controls the value of the property `wt.ixb.import.overrideConflicts`. Most import operations fail if this check box 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 **Resolve Overridable Conflicts** will only resolve the overridable conflicts and cannot 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:** Since Windchill 7.0, `RatioDefinition` and `RatioValue` may cause conflicts. These types of data, if included in an export from 6.2.6 or earlier, result in an overridable import conflict in Windchill 7.0 or later. If you override the conflict, the data is imported as `FloatDefinition` and `FloatValue`.

8. Click **Preview** to understand what will be imported. With the **Detailed Log** cleared, you can see how many items 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 action does not perform actual import, nor does it report all conflicts, especially those from runtime.
9. 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 item in the database with the same UFID or the same name, number, version, and iteration as the item in the XML file. If such an item 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 item 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 item.

The format of new information that must be provided in the Import Policy file is the following:

```
<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 item 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 item in the XML file for an item in the database that has the name, number, version, and iteration provided in the Import Policy file. If such an item does not 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 item in the XML file. This action does not 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 ensure 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 items) 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 item.
- 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 item.
- wt.iba.definition.hierachicaldefinition.enabled -- Beginning in Windchill 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, the 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 item action for which you do not have privileges, the entire transaction will fail.

For example, a user without Revise privileges for a particular item type cannot defeat ACL control by using the import action, **Import as 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 Foundation & 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 Items

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	Cabinet	All	non-administrator user	Full Control (All)

## Export Rules for WTParts

Domain	Context	Type	State	Principal	Grant Permissions
System	Site	View	All	non-administrator user	Full Control (All)
Marketing	Windchill PDM	WTPart	All	non-administrator user	Read

## Export Rule for WTParts With Policy File for Lock and Checked Out

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTPart	All	non-administrator user	Read/Modify

## Export Rule for WTDocuments

### Export Rule for WTDocuments With Policy File for Lock and Checked Out

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTDocument	All	non-administrator user	Read

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTDocument	All	non-administrator user	Read/Modify

#### Export Rule for EPMDocuments

<b>Domain</b>	<b>Context</b>	<b>Type</b>	<b>State</b>	<b>Principal</b>	<b>Grant Permissions</b>
Marketing	Windchill PDM	EPMDocument	All	non-administrator user	Read

#### Export Rule for EPMDocuments With Policy File for Lock and Checked Out

<b>Domain</b>	<b>Context</b>	<b>Type</b>	<b>State</b>	<b>Principal</b>	<b>Grant Permissions</b>
Marketing	Windchill PDM	EPMDocument	All	non-administrator user	Read/Modify

#### Export Rule for Nested Folders

<b>Domain</b>	<b>Context</b>	<b>Type</b>	<b>State</b>	<b>Principal</b>	<b>Grant Permissions</b>
Marketing	Windchill PDM	SubFolder	All	non-administrator user	Read

## Import Access Control Rules

### Import Rule for All Items

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	Cabinet	All	non-administrator user	Full Control (All)

### Import Rules for WTParts

Domain	Context	Type	State	Principal	Grant Permissions
System	Site	View	All	non-administrator user	Read/Modify/Create
Marketing	Windchill PDM	WTPart	All	non-administrator user	Read/Modify/Create

### Import Rule for New Versions of WTParts

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTPart	All	non-administrator user	Read/Modify/Create/Revise

### Import Rule for WTDocuments

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTDocument	All	non-administrator user	Read/Modify/Create



### Import Rule for New Versions of WTDocuments

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	WTDocument	All	non-administrator user	Read/Modify/Create/Revise

### Import Rule for EPMDocument

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	EPMDocument	All	non-administrator user	Read/Modify/Create

### Import Rule for Nested Folders

Domain	Context	Type	State	Principal	Grant Permissions
Marketing	Windchill PDM	SubFolder	All	non-administrator user	Read/Modify/Create



# 7

## Administering Content Holders and Content Objects

Topic	Page
Overview of Content Holders.....	7-2
Content Handling Configuration .....	7-3
Adding and Updating Data Formats.....	7-4

## 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.

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:

Property	Description
wt.clients.debug=false	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.
wt.content.DEBUG=false	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.
wt.content.httpClass= wt.content.ContentHttp	Identifies the class that processes HTTP requests for upload and download operations. Currently, this value is not configurable and should not be changed.
wt.content.temp=\$(wt.temp)	Identifies a temporary directory to which files will be written to upon upload for intermediate processing. This property is not currently used.
wt.content.uploadImpl=rmi	Identifies the communication protocol used when uploading content from a Windchill applet. Currently, this value is not configurable and should not be changed.
wt.content.validEmptyFile=false	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 <b>false</b> . This value should be set to <b>true</b> only if Windchill needs to store files from some other application or process that actively utilizes 0-byte placeholder files.

Property	Description
wt.doc.primaryContentRequired=true	(Windchill PDMLink only) Identifies whether primary file is a required field for Windchill PDMLink documents. Since the reason Windchill PDMLink documents exist is to hold file content, the default value is <b>true</b> . This value should be set to <b>false</b> only if there is a reason for Windchill PDMLink documents to exist without any primary content.

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.

**Note:** Any changes made to existing data formats or any new data formats that you add should be treated as customizations to your code. Be sure to follow the coding practices introduced in [Windchill Software Maintenance and Best Practices](#).

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.

For details how to customize data formats, see the *Windchill Customizer's Guide*.

# 8

## Configuring and Administering Background Queues

Topic	Page
Overview .....	8-2
Queue Entry States .....	8-2
Out-of-the-box Background Queues .....	8-3
Configuring Background Queues and Related Properties.....	8-9
Regular Queue Maintenance .....	8-14

## 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](#).

## Queue Entry States

Each entry in a background queue corresponds to a processing task. An entry can be in one of the following states:

State	Description
Ready	Corresponds to the initial state of the task. Only entries in the <b>Ready</b> state are selected for execution, based on the order in which they were inserted into the queue (first in, first out).
Suspended	Indicates that an entry is to remain in the queue, but is not eligible for execution until you change its state to <b>Ready</b> . Changing a state can be accomplished from the <b>Queue Manager</b> tab in the System Configurator.
Executing	Indicates that an entry is selected for execution. From the <b>Executing</b> state, the entry goes to either the Completed or the Failed state.
Completed	Indicates that the task was successfully executed. You can purge queues of <b>Completed</b> entries so that the size of a queue does not exceed the storage capacity of the database.
Failed	Indicates that an error occurred during execution. You can purge queues of <b>Failed</b> entries so that the size of a queue does not exceed the storage capacity of the database.



State	Description
Reschedule	Indicates that the queue entry has been executed and automatically rescheduled.
Severe	Indicates that an unexpected problem occurred when the entry was executing. Setting this state also stops the queue from which the entry was executing.

## Out-of-the-box Background Queues

The following sections describe the queues that are established when your Windchill solution is installed. When optional products are installed, additional queues can be established. These queues are described as part of the product description. For example, the queues used for file vaulting are described in the [Administering External File Vaults](#) chapter and the queues used for content replication are described in the [Administering Content Replication](#) chapter.

### CleanUpScheduleQueue

The CleanUpScheduleQueue is used by the StandardRecentlyVisitedService, StandardPurgeService, and the StandardCollectionService services:

- The StandardRecentlyVisitedService uses this queue to remove the oldest items from all recently visited lists when the total number of list items exceeds the value specified for the wt.recent.objectStackSize property located in the wt.properties file. The default value for this property is 100.

A QueueEntry is scheduled to run at midnight the first time the StandardRecentlyVisitedService is started. Every time the QueueEntry runs it re-schedules itself to run again at midnight on the following day.

- The StandardPurgeService uses this queue to cleanup canceled purge jobs. The QueueEntry is only created if the wt.queue.executeQueues property is set to true. By default this property is set to true.

A QueueEntry is scheduled to run every midnight. When the queue runs it deletes cancelled purge jobs older than a day and which also have a status of Awaiting preview.

**Note:** Purge jobs with a status of Awaiting preview are not visible to users viewing the user interface.

- StandardCollectionService uses this queue to disable queries with invalid criterion. This QueueEntry is only created if the both the wt.queue.executeQueues and wt.dataops.objectcol.cleanUpEnabled properties are set to true. By default both of these properties are set to true.

A QueueEntry is scheduled to run at midnight once every seven days. It disables queries with invalid criterion. For example, a query that has a folder

reference that has been deleted. Disabled queries are not visible to users viewing the Query Manager user interface.

## **CtScheduleQueue**

The CtScheduleQueue is used when you have turned on the automatic scheduling of team updates as described in *Windchill Business Administrator's Guide*. By default the scheduling is turned off. Use the `wt.inf.team.useScheduledRefreshGroups` and `wt.inf.team.useScheduledRecomputeProperties` to turn scheduling on.

## **DataSharingQueue**

The DataSharingQueue is used when there is sharing of data between containers. In some cases, the update of the sharing status involves heavy processing. For example, if a folder is shared, all foldered objects are also shared, recursively. This queue is used to process this type of updating in the background. The queue is used so that user response time is not affected by shared update processing.

## **EmailQueue**

The EmailQueue is used by the mail service to queue requests to send mail.

## **forumEventPropogationQueue**

The forumEventPropogationQueue is used by Discussion Forums to delete the following objects:

- Discussion Topics (`wt.workflow.forum.DiscussionTopic`)
- Discussion Posting and replies (`wt.workflow.forum.DiscussionPosting`)
- Attached content to Discussion Postings (`wt.workflow.notebook.Bookmark`)

Failed queue entries can mean one of the following:

- A topic in a Discussion Forum has not been successfully deleted.
- A posting or a reply to a posting in the Discussion Forum has not been successfully deleted
- An attachment to a posting has not been successfully deleted.

You should check this queue each day to ensure that the queue is working properly. If there is heavy use of Discussion Forums, you may want to check the queue more often. For each failed queue entry, reset the queue entry state to ready.

## **Indexing Queue and Bulk Indexing Queue**

Indexing Queue and Bulk Indexing Queue are optional queues that are set up when you install Convera RetrievalWare. These queues are used by the

RetrievalWare as part of the indexing process. You should periodically check the Indexing Queue and Bulk Indexing Queue for failed entries.

If a RetrievalWare queue entry has a **Severe** status, ensure that RetrievalWare is configured correctly and is running. Then reset the queue entry to **Ready**. Resetting the queue entry restarts the indexing process.

## MarkForDeleteQueue

The MarkForDeleteQueue is used by Windchill ProjectLink to implement the marking of projects and their contents as deleted. When a project is marked as deleted (using the Delete action), it no longer appears in any project member's My Projects list. The marking process is done in the background to improve user response time.

The system automatically sends a notification to the project managers group if the deletion of a project fails. The notification includes the exception message that caused the failure. The project manager should investigate the reason for the failure, correct it, and reattempt a project deletion by selecting the project Delete action again.

You should periodically check the MarkForDeleteQueue for failed entries. If an entry has a failed status, the only action needed is to remove it from the queue.

## NotificationQueue

The NotificationQueue is used by the notification service to queue requests to generate and send notifications (policy and subscription based notifications).

## NotificationScheduleQueue

The NotificationScheduleQueue is used by the notification service. Requests are queued to expire subscriptions. The subscription UI allows users to set an expiration date for subscriptions, and the queued request is what removes the subscriptions at expiration.

## PagingScheduleQueue

The PagingScheduleQueue is used by Local Search to clean up temporary results that are stored in the database. The results are associated with search requests of each user. There is only one entry regardless of the number of users executing Local Search.

A failed entry means that the temporary results will not be cleaned up and may impact performance if the data grows too large. If the queue is able to successfully create a new entry, then a subsequent execution will clean up all data (including the data from previous attempts).

You should check this queue each day to ensure that the temporary results stored in the database are being cleaned up. Intermittent failures are not critical

because the successful processing will clean up all data from previous failures. However, all failures should be investigated and reported through Technical Support.

## **PartsLink Queue**

The PartsLink Queue is an optional queue that is set up when PartsLink is installed. The queue is used by the Windchill Partslink Service to publish the parts which have been classified to the Partslink Server.

You should periodically check the PartsLink Queue for failed entries. If an entry has a failed status, the only action needed is to reset the queue entry to **Ready**.

## **ProjectScheduleQueue**

The ProjectScheduleQueue is used by Project Management for sending events corresponding to missed, approaching, and passed deadlines.

There is no need to periodically check this queue; a failure in this queue is extremely unlikely. If there are complaints that project management deadline messages are not being sent, this queue should be checked. However, notification failures are more likely to be caused by problems with the e-mail server.

## **PublisherQueue and PublisherQueue1**

The PublisherQueue and PublisherQueue1 are used by Visualization Services to manage the publishing of CAD data. These queues are created when the first user publishes a job through Windchill Visualization Services.

All submitted publish jobs are added to PublisherQueue. When the job executes in this queue, it is added to a free publishing queue. Publishing queues are named PublisherQueueN (where N is a sequential integer starting at 1). By default, PublisherQueue1 is created; additional publisher queues (for example, PublisherQueue2, PublisherQueue3, and so on) can be created with the Windchill Queue Manager.

At many customer sites, publisher queues are heavily used and execute jobs that run for a long time (possibly, hours). Typically, CAD data is published following checkin. The objects stored in the queue entries are `com.ptc.wvs.server.publish.PublishJob` objects.

The details of publish jobs can be viewed from the publish monitor. From the publish monitor, you can see the queue entries that log information about publishing failures. If a job has failed, use the information in the publish monitor to investigate why it failed and, after fixing the problem, resubmit the job for publishing.

## PurgeHTMLLogMessageQueue

The PurgeHTMLLogMessageQueue is used to periodically purge log messages displayed on the event console that are older than a specified number of days. By default, messages that are older than 5 days are removed. However, you can change the default by specifying a new value in the `com.ptc.windchill.cadx.purgeLogMessagesOlderThan` property. This property resides in the `wt.properties` file.

The queue does not have any queue entries. The queue, when running, calls a method that periodically removes the old messages.

There is no need to periodically check this queue. If users complain that old messages are not being deleted, verify that the queue is running.

## PurgeOrphanedEffAuditsQueue

The PurgeOrphanedEffAuditsQueue is used by the Effectivity service to clean up audit objects.

Effectivity audit objects are created to track the creation and factual deletion of effectivity objects. A factual deletion means that an effectivity record is marked as deleted, but is retained as historical information. When effectivity objects are actually deleted, audit objects can become unreferenced, thus, having no further purpose. Checking for corresponding audit objects each time that an effectivity object is actually deleted is time consuming; therefore, this non-urgent clean up is done on a scheduled basis (by default, once a day). To change the schedule for the clean up, change the interval time set in the `wt.eff.EffChangeAudit.purgeInterval` property, which is located in the `wt.properties` file. Express the interval time minutes; the default value is 1440 minutes (one day).

**Note:** If the value of the `wt.eff.EffChangeAudit.purgeInterval` property is set to zero or a negative value, the clean up will not be performed.

Failed PurgeOrphanedEffAuditsQueue entries mean an attempt to query for and delete the audit items has failed. Checking for failed PurgeOrphanedEffAuditsQueue entries is not needed on a regular basis. Each queue entry has identical functionality; therefore, if failed entries are noticed, they can be deleted (since they will be replaced by future ones). If the problem occurs chronically, check the system configuration and consider filing a problem report with Technical Support. Also, since the Effectivity service is programmed to create this queue on startup (if it does not exist), a problematic instance of the queue can simply be deleted, along with all of its entries (regardless of their status).

## StatisticsScheduleQueue

The StatisticsScheduleQueue is used by the soft type/attribute query service to gather statistics related to Instance Based Attributes. These statistics are used when optimizing queries.

A failed queue entry implies that statistics were not collected. If statistical data is not up-to-date, then the soft type/attribute query performance may not be optimized.

The frequency for gathering the statistics is controlled by the `com.ptc.core.query.optimize.statisticsBasedRankGenerator.queueInvokeTime` property setting. The default is once a day. See `properties.html` for details.

## **WfPropagationQueue**

The `WfPropagationQueue` is used by workflow (and its associated tasks) to propagate all state changes to Workflow objects. This includes any routing expressions and transition expressions associated with those state changes.

The workflow queues are likely to be used heavily when many concurrent Windchill users are completing workflow tasks. Also, they are likely to be used heavily in scenarios where Windchill business objects are created that use life cycles which in turn use workflows.

Failed entries in workflow queues mean that something failed to process correctly. In most cases, a failed workflow queue entry corresponds to a stack trace in the method server log. Analyze the queue entry failure by checking the method server log to determine the cause of the failure. Sometimes, the message listed in the queue manager can be enough to determine the cause of the failure.

You should examine workflow queues when the workflow activities associated with the queues do not appear to be executing properly. Additionally, it is good practice to periodically check the queues to clean out old entries.

## **WfScheduleQueue**

The `WfScheduleQueue` is used by workflow (and its associated tasks) to queue all timed events. Deadline checks for any workflow object with a deadline set and expression-based Synchronization Robots are executed within this queue.

The workflow queues are likely to be used heavily when many concurrent Windchill users are completing workflow tasks. Also, they are likely to be used heavily in scenarios where Windchill business objects are created that use life cycles which in turn use workflows.

Failed entries in workflow queues mean that something failed to process correctly. In most cases, a failed workflow queue entry corresponds to a stack trace in the method server log. Analyze the queue entry failure by checking the method server log to determine the cause of the failure. Sometimes, the message listed in the queue manager can be enough to determine the cause of the failure.

You should examine workflow queues when the workflow activities associated with the queues do not appear to be executing properly. Additionally, it is good practice to periodically check the queues to clean out old entries.

## WfUserWorkQueue

The WfUserWorkQueue is used by workflow (and its associated tasks) to instantiate workflow robots and execute workflow robot actions.

The workflow queues are likely to be used heavily when many concurrent Windchill users are completing workflow tasks. Also, they are likely to be used heavily in scenarios where Windchill business objects are created that use life cycles which in turn use workflows.

Failed entries in workflow queues mean that something failed to process correctly. In most cases, a failed workflow queue entry corresponds to a stack trace in the method server log. Analyze the queue entry failure by checking the method server log to determine the cause of the failure. Sometimes, the message listed in the queue manager can be enough to determine the cause of the failure.

You should examine workflow queues when the workflow activities associated with the queues do not appear to be executing properly. Additionally, it is a good practice to periodically check the queues to clean out old entries.

## Configuring Background Queues and Related Properties

This section describes how to configure multiple background MethodServers, the properties for background queues, background queue logs, and other related properties. All the properties in the tables that follow are defined in the wt.properties file.

### Configuring Multiple Background MethodServers

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 being 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 `appsvr1`, `appsvr2`, and `appsvr3`. 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.

## Background Queue Properties

Use the properties described in the following table to configure the background queues:

Property	Description
<code>wt.queue.&lt;queuename&gt;</code>	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 <a href="#">Overview</a> .
<code>wt.queue.defaultInterval</code>	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 <i>polling interval</i> . The queue begins processing again when the polling interval has elapsed.  Default is 60.



Property	Description
wt.queue.execEntriesCount	Sets the number of entries queried from the queue to be executed. Default is 6.
wt.queue.executeQueues	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.
wt.queue.max.processQueues	Sets the maximum number of process queues that the queue service creates before throwing an exception. Default is 12.
wt.queue.max.scheduleQueues	Sets the maximum number of schedule queues that the queue service creates before throwing an exception. Default is 12.
wt.queue.queueGroup	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> .
wt.queue.queueGroup.default	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> .
wt.queue.queueMonitor.sleep	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.
wt.queue.removeCompleted	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).
wt.queue."+queueName+" .removeFailedEntries	Determines whether failed entries are automatically removed. Default is false (to not remove).
wt.queue."+queueName+" .exceptionRetries	Determines whether failed execution entries are retried. These retries occur back to back, with no wait time. Default is 0.

## 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.

The properties in the table below provide information related to queue architecture and schedule queue type:

Property	Description
wt.queue.log.enabled	Determines whether queue events are logged. This is a global property, so the log created contains information about all existing queues.  Default is false (to not log queue events) .
wt.queue.log.file	Sets the name of the queue log file.  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.
wt.queue.log.append	Determines whether queue log entries append to or overwrite the existing log file.  Default is true.
wt.queue.pollingQueueThread.verbose	Provides debug information specific to the actual polling threads.  Default is false.
wt.queue.processingQueue.verbose	Provides general processing queue information that can be used to debug problems.  Default is false.
wt.queue.processingQueue.execEntries.verbose	Provides debug information related to the execution of a set of process queue entries.  Default is false.
wt.queue.queueWatcher.verbose	Provides information related to the control of a specific queue. Each queue has an associated queue watcher.  Default is false.
wt.queue.queueWatcher.update.verbose	Provides queue state update debug.  Default is false.

Property	Description
wt.queue.scheduleQueue.verbose	Provides general schedule queue information that can be used to debug problems. Default is false.
wt.queue.scheduleQueue.execEntries.verbose	Provides debug information related to the execution of a set of schedule queue entries. Default is false.
wt.queue.scheduleQueue.execEntry.verbose	Provides debug regarding the execution of individual schedule queue entry. Default is false.
wt.queue.scheduleQueueEntry.verbose	Provides execute debug for schedule queue entries. Default is false.
wt.queue.scheduleQueueThread.verbose	Provides debug information related to the actual scheduling threads. Default is false.

## Other Background Queue-Specific Properties

There are other Windchill properties specific to queuing. The following table includes three examples:

Property	Description
wt.index.defaultQueueInterval	Specifies the number of seconds in the time-out interval of the index queue polling thread. Default is 60.
wt.index.useQueue	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.
wt.queue.notifyInterval	Specifies the interval of queue failure notification. The notification is done through e-mail. Default is 3600 seconds (1 hour).

See [properties.html](#) for descriptions of all available properties.

To add supported properties to a property file (in this case, the wt.property file) or change values for existing properties, use the xconfmanager utility. For details, see [Using the xconfmanager Utility](#).

## 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 Queues

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.



# 9

## Customizing and Administering Pro/ENGINEER Wildfire

This chapter presents customization and administration information and recommendations for using Pro/ENGINEER Wildfire integrated with Windchill Foundation & 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 item naming, automated part creation, supporting custom parts, and customizing the user interface. In addition, recommendations for system configuration and performance tuning are offered.

Topic	Page
Customizing Pro/ENGINEER Wildfire .....	9-2
System Configuration Recommendations .....	9-48
Performance Tuning .....	9-48
Other Recommendations .....	9-51
Default Directory INI Files .....	9-52
Site Directory INI Files .....	9-59

# 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 items prior to an upload to the server, and to keep copies of items 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:

Variable	Values	Description
PTC_WF_ROOT	/path/to/dir, Default on UNIX = ~/wf Default on Windows = [User Profile]\Application Data\PTC\ProENGINEER\ Wildfire\	Overrides the default location of .wf directory. Setting this environment variable will cause Pro/ENGINEER to use the new location as a location for the cache.  <b>Note:</b> 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.  <b>Note:</b> 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:

Config.pro Option	Values	Description
disable_search_path_check	no [default] yes	<p>Controls whether or not the search path is checked for name conflicts when creating, renaming, or copying models.</p> <p>When set to "yes," disables the check of the search path for a naming conflict when a new file is created. This can speed up file creation by postponing the search path check (which includes the entire commonspace) until an upload is performed.</p>
dm_auto_open_zip	yes (default) no	<p>Defines how Pro/ENGINEER will handle zip files.</p> <p>If set to "yes", then Pro/ENGINEER opens the zip file and attempts to retrieve items in the zip file. If the zip file contains more than one file (for example, in the case of assemblies), Pro/E will first attempt to open an item in the zip file that has the same name as the zip file itself. If it finds one, it will open it, if not, it will display the contents of the zip file in the <b>File&gt;Open</b> dialog box.</p> <p>If set to "No", Pro/ENGINEER treats a zip file like a directory, and displays the contents of the zip file in the <b>File &gt; Open</b> dialog box, allowing the user to pick the file from the zip that he or she wants to retrieve into session.</p>

Config.pro Option	Values	Description
dm_cache_mode	all (default) modified none	<p>Defines how Pro/ENGINEER writes in-session data to persistent storage. It can take one of 3 values:</p> <p>All -- Pro/ENGINEER makes maximum use of the available cache to optimize file transfer performance. Whenever possible, Pro/ENGINEER caches data that is sent to or retrieved from the server.</p> <p>Modified -- Only files modified by Pro/ENGINEER are saved to the cache. Files that are downloaded from the server are not cached. This will slow down retrieval times.</p> <p>None -- No files are written to the local cache upon Pro/ENGINEER <b>Save</b> (all data is uploaded directly to the server) and no files are cached on download. Using this option puts Pro/ENGINEER into "streaming" mode that minimizes the cache but slows down save and retrieval operations.</p>
dm_cache_size	Integer [default = 400]	<p>Sets the size (in MB) of the cache allocated to each workspace on the client hard disk.</p> <p>Recommendation: If possible, set the cache size large enough to accommodate the largest anticipated data set (the downloaded content, and the locally modified content prior to upload should be counted separately).</p>
dm_checkout_on_the_fly	checkout (default) continue	<p>If set to the default value "checkout," the default action for the <b>Conflicts</b> ("checkout-on-the-fly") dialog box is "Check Out Now."</p> <p>If set to "continue," the default action for the <b>Conflicts</b> dialog box is "Continue."</p>

Config.pro Option	Values	Description
dm_http_compression_level	Integer, from 0 (no compression) to 9 (maximum compression) [default = 0]	<p>Sets the level of compression for data upload and download.</p> <p>Although compression speeds up transfer over the network, it uses server CPU and client CPU to perform the compress and decompress operations. In a local area network, where network transfers are rapid, compressing and decompressing data can result in lesser throughput. On a wide area network with lower bandwidth compression can lead to higher throughput. Since this is set per client, PTC recommends that clients in a LAN use a value of 0 (the default) and clients in a WAN use a value of 2 or 3.</p>
dm_network_request_size	integer >0 [default = 100000]	<p>Determines the maximum size, in bytes, of an HTTP upload request when uploading content files to Windchill.</p> <p>A small value (say 8000) would mean a lot of small HTTP requests to the method server containing the model files which may add overhead, but because the local Wildfire file buffers are filled quickly, the upload starts sooner.</p> <p>A much larger value (say 800000000) may allow the uploading of the entire data set in a single HTTP request, but it could take a while for the client to write the files from local disk to its internal buffer before streaming the content to the server. In addition, because of apparent size limitations of the Microsoft HTTP API you may experience random upload failures with very large file size data sets in Windows.</p>

Config.pro Option	Values	Description
dm_network_retries	integer >0 [default = 10]	<p>Sets the number of attempts to connect to a Windchill server before the connection is considered broken.</p> <p>(This is a hidden config option that was introduced in Pro/ENGINEER Wildfire date code 2003210.)</p> <p>Recommended setting: default</p> <p><b>Note:</b> 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.</p>
dm_network_threads	integer >0 [default = 3]	<p>Sets the number of concurrent threads Pro/ENGINEER uses for uploading and downloading data to and from a Windchill server.</p> <p>In most cases, increasing the number of threads in a LAN environment will not improve performance, as the disk will then become the bottleneck. Even in a WAN environment, settings greater than the default are unlikely to improve throughput significantly.</p> <p>(This is a hidden config option that was introduced in Pro/ENGINEER Wildfire date code 2003210.)</p> <p>Recommended setting: default</p>
dm_overwrite_contents_on_update	no (default) yes	<p>Specifies behavior during Update action.</p> <p>If set to "no," does not overwrite the locally modified contents for out-of-date items, but updates their metadata only</p> <p>If set to "yes," overwrites the locally modified or out-of-date items with the ones in the server in addition to updating their metadata</p>
dm_remember_server	yes [default] no	<p>If this option is set to "yes," the last primary server/workspace of a Pro/ENGINEER session is set automatically for the next Pro/ENGINEER session.</p>

Config.pro Option	Values	Description
dm_search_primary_server	yes [default] no	If this option is set to "yes," during retrieval, the system searches the primary server for dependencies not found in the workspace
dm_secondary_upload	automatic [default] explicit	Defines the behavior of saving to a secondary server (See also dm_upload_objects).
dm_upload_objects	explicit [default] automatic	<p>Defines the behavior of the <b>Save</b> command in Pro/ENGINEER.</p> <p>If this option is set to explicit, the Pro/ENGINEER <b>File &gt; Save</b> command will write data to the cache. The user must then explicitly send that data to the server (using either <b>File &gt; Save and Upload</b> or <b>File &gt; Checkin</b>). If this option is set to automatic, <b>File &gt; Save</b> in Pro/ENGINEER will also upload the Pro/ENGINEER files to the server.</p>
let_proe_rename_pdm_objects	no [default] yes	<p>Determines whether an item retrieved from a PDM database can be renamed in a Pro/ENGINEER session</p> <p>An item rename in Pro/ENGINEER is seen only by parents in session. The item is seen as a new item when saved to the workspace.</p> <p>This option can be used to replace a standard sub-assembly with a copy of itself with a unique name.</p>
open_simplified_rep_by_default	no (default) yes <i>&lt;name_of_simplified_rep&gt;</i>	<p>Specifies whether to prompt user to select a simplified representation when opening a Pro/ENGINEER file</p> <p>If set to "yes," user is prompted to open a simplified representation when opening a Pro/ENGINEER file</p> <p>If set to the name of a simplified representation, the system opens the simplified representation without prompting the user.</p>

Config.pro Option	Values	Description
regenerate_read_only_objects	yes (default) no	Specifies whether read-only items (items not checked out) are regenerated.  Set to "yes," it specifies that read-only parts with relationships to an explicitly modified assembly are modified implicitly upon regeneration of the assembly. (Explicit changes to a checked in item cause the <b>Conflicts</b> dialog box to appear.)  By setting to "no," you may avoid having read only workspace items marked as modified. This, in turn, can reduce the number of files required for the checkout of an associated assembly
save_model_display	wireframe, shading_low, shading_high, shading_lod	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.
save_objects	changed_and_specified [default] all changed changed_and_updated	Determines when an item and its dependent items (such as a part used in an assembly) are stored.
web_browser_homepage	string value	Sets the location of Pro/ENGINEER browser homepage.

**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 (<server\_alias> is the name you assign to the server in the Server Registry):

```
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 item 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 item (for example, save or requesting checkout) are done in the context of the primary server, not the location from which the item 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 WTPart, 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:

`<Windchill>/codebase/com/ptc/windchill/cadx/cfg/`

INI files are installed by default under the following subdirectories:

- `<Windchill>/codebase/com/ptc/windchill/cadx/cfg/site/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.

- `<Windchill>/codebase/com/ptc/windchill/cadx/cfg/default/cadxhtmlui.ini, contentcat.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
CADAssemblyDocNumber=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=00008_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 Foundation & 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.

## INI File Recommendations

### 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
```

### Managing CAD Document and WTPart Naming and Numbering

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

## Policy-Managed Naming and Numbering

Pro/ENGINEER Wildfire supports four policies to determine how newly-created items (either CAD documents or WTParts) are named and numbered. The four policies can be described briefly as follows:

- Auto-numbering
  - The CAD document Number is provided by the CADDoc number generator (either OOTB or your customization)
  - If the system parameter PTC\_COMMON\_NAME is created during creation of the Pro/ENGINEER model file, its value is copied to the CAD document Name. If Common Name is left blank in Pro/ENGINEER, the default value for the CAD document Name is copied from the Pro/ENGINEER model name. The file extension (.prt) can be optionally dropped (controlled by a property).
  - The WTPart Number is provided by the WTPart Number generator
  - The default value for the WTPart Name is copied from the current value of the CAD document Name at the time the WTPart is created. If a file extension is present in the CAD document name, it can be optionally dropped when set in WTPart (controlled by a property).
  - In any create and edit user interface, the CADDoc and WTPart NAME field is editable
  - In any create and edit user interface, the NUMBER field is not editable.
  - Auto numbering is the OOTB naming and numbering policy.
- Name-driven
  - The CAD document Number is copied from the Pro/ENGINEER Model Name (the file extension can be dropped – controlled by a property)
  - If the system parameter PTC\_COMMON\_NAME is created during creation of the Pro/ENGINEER model file, its value is copied to the CAD document Name. If Common Name is left blank in Pro/ENGINEER, the the default value for the CAD document Name is copied from the Pro/ENGINEER model name (file extension can be dropped – controlled by a property)
  - The default value for the WTPart Number is copied from the CAD document Number (the file extension can be dropped – controlled by a property)
  - The default value for the WTPart Name is copied from the current value of the CAD document Name at the time the WTPart is created (the file extension can be dropped – controlled by a property)
  - In any create and edit user interface, the CAD document and WTPart NAME and NUMBER fields are editable by the user.

- Parameter-driven
  - The CAD document Number is copied from the value of the Pro/ENGINEER designated parameter identified by the INI key PROENumberingParameter.
  - The value for the CAD document Name is copied from the Pro/E designated parameter identified by the INI key PROENamingParameter.
  - The value for the WTPart Number is copied from the value of the IBA identified by the INI key AutoAssociateNumberingParameter.
  - The value for the WTPart Name is copied from the value of the IBA identified by the INI key AutoAssociateNamingParameter.
  - In any create and edit user interface, the CAD document and WTPart NAME and NUMBER fields are editable by the user.

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

**Note:** If the designated parameters change after the creation of items, neither the associations nor the names of CAD documents or WTParts change.

- Custom

The purpose of the custom policy is to provide users with a way to modify the system behavior.

The Object Initialization Rules administrator, available on the Windchill ProjectLink or Windchill PDMLink **Utilities** tab, provides a way to specify default values for the attributes of a specific item type. The default values are then used when the Windchill solution creates items of that type. These specifications are called *rules*. Each rule can contain default values for one item type. The rules that are set only apply when the Windchill solution that is used to create an item does not set a corresponding value. Rules can be setup to provide auto-number generation but they can also be setup to provide custom behavior. Rules are also set per container, allowing this way to have different naming/numbering policies on different containers.

**Note:** Regardless of the naming and numbering policy used, when creating a new item, system uniqueness constraints require that the CAD document attributes Number and File Name must both be unique within a Windchill Foundation & PDM site, within a Windchill PDMLink site, or within each Windchill ProjectLink project.

## Identifying the Current Naming and Numbering Policy

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

- If auto-numbering is set in Rules, then the policy is auto-numbering.
- If custom behavior is implemented in Rules, then the policy is custom.
- If neither auto-numbering or custom behavior is set and the parameter-driven properties are set in cadxhtmlui.ini, then policy is parameter-driven.
- Otherwise, the policy is name-driven.

## INI Properties for Naming and Numbering

The properties for parameter-driven policy in cadxhtmlui.ini are the following for auto-associate:

AutoAssociateNamingParameter=<Some String Parameter>

AutoAssociateNumberingParameter=<Some String Parameter>

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

PROENamingParameter=<Some String Parameter>

PROENumberingParameter=<Some String Parameter>

By default, none of these four properties has a value.

**Note:** PROENumberingParameter can not be used with family table parts that have more than one level of nested instances. If PROENumberingParameter is used with such family table parts, upload fails with a valid number uniqueness exception. When PROENumberingParameter is used with family table parts that have only one level of instances, values of this parameter need to be different for each instance. You can do this by adding this parameter as a family table column and providing a different number value for each instance.

When set to true (default is false), the following properties specify dropping the file extensions (such as, ".prt" or ".asm") when naming and numbering new items during auto associate:

AutoAssociateDropNameFileExtension=

AutoAssociateDropNumberFileExtension=

When set to true (default is false), the following properties specify dropping file extensions during upload:

UploadDropNameFileExtension=

UploadDropNumberFileExtension=

## Customizing the Naming Service

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

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

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

- Name:
  - 1.Naming service customization
  - 2.Explicitly assigned Common Name through the Pro/ENGINEER **File > New** dialog box
  - 3.Name parameter (PROENamingParameter=)
  - 4.File Name (The option "UploadDropNameFileExtension=" will take in effect only if Name is assigned based on File Name (CAD Name))
- Number:
  - 1.Naming service customization
  - 2.Number parameter (PROENumberingParameter=)
  - 3.File Name (The option "UploadDropNumberFileExtension=" will take in effect only if Name is assigned based on File Name [CAD Name])

Use the following steps to customize the Naming service:

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

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

public interface EPMDocumentNamingDelegate
{
    public void validateDocumentIdentifier(DocIdentifier
docIdentifier);
}
```

The definition of Class DocIdentifier is as follows:

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

import java.util.HashMap;

public class DocIdentifier
{
```

```

    {
        private String m_modelName;
        private String m_docName;
        private String m_docNumber;
        private HashMap m_parameters;
    }

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

    {
        m_modelName = modelName;
        m_docName= docName;
        m_docNumber= docNumber;
        m_parameters= params;
    }

    /** get the CAD Name for the model */
    public String getModelName()

    {
        return m_modelName;
    }

    /** get  the EPMDocument name for the model */
    public String getDocName()

    {
        return m_docName;
    }

    /** set the EPMDocument name for the model */
    public void setDocName(String docname)

    {
        m_docName = docname;
    }

    /** set the EPMDocument number for the model */
    public void setDocNumber(String docnumber)

    {
        m_docNumber = docnumber;
    }

```

```

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

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

```

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

```

public void validateDocumentIdentifier(DocIdentifier
docIdentifier)

```

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

Use the DocIdentifier.getModelName() to get the CAD Name of the EPMDocument that this DocIdentifier object represents.

- The Pro/ENGINEER designated parameters may be used to set EPMDocument numbering/naming.

Use the DocIdentifier.getParameters() to get the associated parameters.

Use the “set” methods on the DocIdentifier to set the new name/number values. The Upload Service will use these suggestions if they are feasible.

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

```

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

<Option cardinality="singleton" requestor="wt.epm.EPMDocument"
serviceClass="com.ptc.windchill.uwgm.proesrv.c11n.EPMDefaultDocument
NamingDelegate"/>

</Service>

```



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

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

4. Restart the method server.

## Managing Incomplete Items

Incomplete dependent items (commonly referred to as "ghosts") are those that are referenced by, but not available to a parent item currently in session. While Pro/ENGINEER Wildfire can display incomplete dependents in the workspace and upload them to the server, Windchill does not allow the check-in of incomplete items. A key in the cadxhtmlui.ini file,

GhostIgnoreDependencyOption, can be set to one of four values to allow check-in of an assembly by ignoring certain dependencies or to disallow the ignoring of dependencies (and thereby disallow the check-in of assemblies containing unresolved incomplete dependent items).

If GhostIgnoreDependencyOption is set to one of the following values, the behaviors described result:

- If set to IgnoreOptionalDependencies (default), any Pro/ENGINEER internal and non-required dependencies are ignored.
- If set to IgnoreOptionalReferenceDependencies, any reference dependencies are ignored.
- If set to IgnoreProeInternalDependenciesOnly, only internal Pro/ENGINEER dependencies are ignored.
- If set to DoNotAllowIgnore, no incomplete dependencies can be ignored.

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

## Soft Typing CAD Documents

### About Soft Typing

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

The Windchill counterpart to restricted value parameters is the soft typing feature. Soft typing is accessible through the **Type Manager** and allows you add constraints (such as a value or range of values) to an instance-based attribute. Additionally, you can use the **Type Manager** to add attributes to the EPMDocument type and its soft types: both the CAD document and Dynamic document (Arbortext document) soft types.

For CAD documents, there is one system-provided soft type, the Workgroup Manager CAD Document, that an administrator can modify to add attributes that can have different values for each iteration of the item that an administrator can modify. This soft type cannot be deleted nor sub-typed further; however, there are additional soft types, related to CAD documents. They are the following:

- Workgroup Manager CAD Document Master soft type (on CAD Document Master type)

Attributes that are added to this soft type have only one value for all iterations. Changing the value of an attribute on a CAD Document Master soft type changes that value for all iterations. This type of attribute is the Windchill equivalent of a Pro/INTRALINK non-versioned attribute.

- Workgroup Manager CAD Document Uses Link soft type (on CAD Document Uses Link type)

Attributes that are added to this soft type are specific to the use of an iteration of an item. For example, if there are four bolts of the same type (bolt.prt) in an assembly, and each bolt needs to be tightened to a specific torque, you can add torque to the Uses Link soft type and then apply a different value to each occurrence of the bolt in the assembly. (In contrast, if you instead add this attribute to the CAD Document soft type, then all bolts in all assemblies would have the same torque wherever they are used.)

- Workgroup Manager CAD Document Reference Link soft type (on CAD Document Reference Link type)

Attributes that are added to this soft type apply to reference links (again, not to the CAD document, itself).

For details on soft typing, selecting attributes and setting constraints, refer to the *Windchill Business Administrator's Guide*.

## 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:

- Implicit parameter-to-attribute mapping
- Explicit parameter-to-attribute mapping

### 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.

**Note:** For CAD documents, there is one system-provided type that an administrator can modify. This type cannot be deleted nor sub-typed. In order to assign any attributes to CAD documents, the administrator must go into the Type Manager and add attributes to the default CAD document soft type.

### Explicit Parameter-to-Attribute Mapping

Two specific cases where explicit mapping can be helpful are the following:

- When parameter names in Pro/ENGINEER are in multi-byte characters (multi-byte character names are not supported in Windchill)
- When you are dealing with older files with legacy parameter names

**Note:** In older versions of Pro/ENGINEER (prior to Wildfire 2.0) the system does not keep track of the parameter-to-attribute mapping of each iteration of an item. This could lead to loss of historical data if the mapping file is changed between early and later iterations. For example, consider the situation where "LENGTH" in a Pro/ENGINEER file is mapped to "length" in Windchill for the first three iterations of an item (A.1, A.2, A.3), but the mapping file is then changed so that "LENGTH" is mapped to "diameter" in Windchill when iterations A.4, and A.5 are created. Upon attempting to retrieve A.1, the system looks into the mapping file, trying to find "diameter" instead of "length". Because it cannot find "length," there is no way to show the value for "LENGTH" in Pro/ENGINEER for iteration A.1.

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:

```
<Windchill>/codebase/com/ptc/windchill/uwgm/proesrv/attribute
```

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
```

**Note:** For CAD documents, there is one system-provided type that an administrator can modify. This type cannot be deleted nor sub-typed. In order to assign any attributes to CAD documents, the administrator must go into the Type Manager and add attributes to the default CAD document soft type.

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.

## 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 (succeeds with a warning written to the Event Manager). All conflicts are reported in the Event Manager 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.

## Resolving Type Conflicts Between Pro/ENGINEER Parameters and Windchill IBAs

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

```
<Service context="default" name="wt.epm.attributes.EPMAAttributeDelegate"
targetFile="codebase/service.properties">

  <Option cardinality="singleton" requestor="wt.iba.value.IBAHolder"
selector="PROE"
serviceClass="wt.epm.attributes.EPMAAttributeDelegateWithWarnings"/>

</Service>
```

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

## Customizing the Parameters in the Download Service

Windchill provides a server-side delegate that can be used to insert parameters into a 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:

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

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

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

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

</Service>
```

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

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

3. Restart the method server.

## Configuring the Build Rule

The epm.xconf file (codebase/wt.properties) contains properties that can be set to control the following during execution of the build rule:

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

## Controlling Attribute Publishing

You can set a property to define the attributes that are published to a build target.

For build targets that do not belong to a team template, or belong to a team template without a defined property, a default property is defined, as follows:

wt.epm.build.attributes.default=<delimiter-separated list of attributes>

An asterisk (\*) in place of the list of attributes is the property's default value, to indicate that all attributes are to be published.

**Note:** The property wt.epm.build.attributes.delimiter defines the delimiter that separates the listed attributes. A comma (,) is the default value.

For a specific team template, the syntax is as follows:

wt.epm.build.attributes.teamtemplate.<container path>.<teamtemplate name>=<delimiter-separated list of attributes>

**Note:** Change comma separators within the container path to periods (.) with the following space removed. Change spaces within the container name to underscores (\_)

Example:

```
wt.epm.build.attributes.teamtemplate.Site.Organization_Design,  
Library_Stock_parts.TestTemplate=Color,Length,DiameterInner,DiameterOuter
```

For the single container (context) of Windchill Foundation & PDM, no container path is included.

Example:

```
wt.epm.build.attributes.teamtemplate.TestTemplate=Color,Length,DiameterInner,  
DiameterOuter
```

## Specifying Usage Links

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

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

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

## Vetoing Operations Based on Owner Application

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

- Add link
- Change folder (Move)
- Check-in
- Checkout
- Delete IBA
- Delete link
- Delete (item)
- Modify property
- Revise

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

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

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

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



wt.epm.veto.change.folder=OPTEGRAGATEWAY,PROPDMGATEWAY,WORKMANAGERGATEWAY,IDEASTDM.

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

## Enabling As Stored Configurations

As stored configurations are enabled by the property wt.epm.workspaces.createAsStored=true.

The property wt.epm.workspaces.createAsStored.AuthoringApplications enumerates a list of authoring applications that allow a CAD document to be added to an as stored baseline. For example:

wt.epm.workspaces.createAsStored.AuthoringApplications=PROE

**Note:** The property wt.epm.workspaces.createAsStored.AuthoringApplications.delimiter defines the character used to delimit the list of owning applications. The default value is comma (.).

## Managing ModelCHECK Validation during Check In

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

## Configuring ModelCHECK

To enable the gatekeeper functionality, you must first edit the ModelCHECK configuration to enable ModelCHECK to add the required parameters to the data. From the **ModelCHECK Configuration** dialog box within Pro/ENGINEER, edit the initialization file (config\_init.mc) and change the following items:

- Set **ADD\_DATE\_PARM** to **Y**

This creates a parameter called MODEL\_CHECK in the model files of all models that are checked. This parameter contains the date and time when ModelCHECK was last run.

- Set **ADD\_ERR\_PARM** to **Y**

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

- Set **ADD\_CONFIG\_PARM** to **Y**

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

- Set **ADD\_MODE\_PARM** to **Y**

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

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

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

## Windchill Configuration

After configuring ModelCHECK, configure Windchill as follows:

- Set the property "ModelCheckValidation" to "true" in \$WT\_HOME/codebase/com/ptc/windchill/cadx/propfiles/cadxcommon.properties
- Set the appropriate modelcheck option in the [modelcheck] section of \$WT\_HOME/codebase/com/ptc/windchill/cadx/cfg/default/cadxhtmlui.ini
  - NumErrors - Maximum number of errors allowed (integer). The default value is zero.
  - NumHours - Number of hours since the last ModelCheck run (integer). The default value is 24.
  - MConfig <comma separated string>- Type of configuration used followed by actual set of MC Configuration files to be used for validation. Set to "Medium"configuration by default

Example:

MConfig=Medium,check/default\_checks.mch,start/nostart.mcs,constant/inch.mcn.

- Alternatively, you can modify the existing way of specifying the MConfig files by defining a config files for each of the lifecycle states in predefined syntax. If the lifecycle used for any particular check-in transaction does not match any of the listed lifecycles, gateway will assume default config files for validating the model being checked in.

- The syntax for defining MCCConfig to list the lifecycles can be as follows:

Example:

```
MCCConfig=<Lifecycle_1>:<mch_file1>,<mcs_file1>,<mcn_file1>
<Lifecycle_2>:<mch_file2>,<mcs_file2>,<mcn_file2>.....
```

**Note:** See <Pro/ENGINEER>/modchk/text/usascii/config/setconf.mcc for a listing of analysis types and the corresponding configuration files.

- Create IBAs with following names and types in the Attribute Administrator:
  - MC\_ERRORS--(integer)
  - MODEL\_CHECK--(string)
  - MC\_CONFIG--(string)
  - MC\_MODE--(string)

## Configuring the Initial Collection of Items for Actions

When the user initially selects an item and enters an action, one of the first processes invoked is to collect the set of all possible dependents and related items to the initially selected item (this includes all dependents of any Pro/ENGINEER drawings or layouts). Only the default items for the action will initially appear in the list. While this makes the inclusion of dependents and related items quite easy, collecting this superset of items can negatively affect performance, especially when drawings are involved.

To offer an alternative approach, the property `com.ptc.core.collection.collectall` allows you to specify either the immediate collection of the superset or the collection of only those items required by the default rules for the action.

When set to true, the superset of possible dependents and related items is collected. When set to false, only those items required by the default rules for the action will be collected. All other items are collected when the user selects the item and invokes a collection rule.

**Note:** When `com.ptc.core.collection.collectall` is set to false (default), all items in the item list show the default state of the collection command (to add). In some cases this will be correct; in some cases it will not be. When the default case is not correct for a rule, once the user selects the rule button, the button is disabled.

## 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 item type list on the Pro/ENGINEER Wildfire HTML client item selection page, if "Part" or "All" is selected, both WTPart items and custom part items are listed in the page's results table.

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

In addition you must modify the Pro/ENGINEER Wildfire HTML files that use the item selection page, and use the xconfmanager modify or override the type list id entries in com\ptc\windchill\cadx\propfiles\picker.properties.

**Note:** For wt.query.SearchAttributeList, which is the default configured search attribute list, the type id is referred to as the query value.

(See [Customizing the HTML Client Item Selection Page](#) later in this chapter for more details.)

## Replacing WTPart

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

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

3. Restart the method server.

## Supporting WTPart and Custom Part

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

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

## Administering Revision

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

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

**Note:** The revision level synchronization behavior described in the section, [Configuring the Synchronization of Revision Levels During Autoassociate](#) also applies to the revise action when AutoAssociateSetRevisionForWTPart is set to true.

The following sections describe each of these configurations.

### Configuring the Ability to Set a Revision Level

A key-value pair in the cadxhtmlui.ini file (located at <Windchill>/codebase/com/ptc/windchill/cadx/cfg/default /cadxhtmlui.ini), allows you to enable users to set a higher revision level than the next label in the revision scheme.

When set to true (default) the key SetTargetRevision adds a **Set Revision** button to the **Revise** user interface that allows the user to skip one or more levels of revision during a revise operation. In addition, a revision level can be set in the **Create CAD Document** user interface.

### Configuring the Revision of Associated Items

You can configure how passive links are carried forward to new revisions of associated (linked) items. Using the xconf file, modify

<Windchill>/codebase/wt.properties, which contains the following key-value pairing:

Key = wt.vc.struct.copyDescribeLinkOnRevise

Values: false, true (default)

With wt.vc.struct.copyDescribeLinkOnRevise set to the default value, true, the system behaves as follows for the revision of a CAD document or part (WTPart) with a passive association:

- There is only one link, and it is iteration-to-iteration.
- If only the CAD document is revised, the link is not carried forward. The new version of the CAD document does not have an association to the existing versions of the part (because the existing part version might be released, and the content version of the part should not change without a check out or revise).
- If only the part is revised, both the new and the old version of the part have a link to the existing version of the CAD document.
- If both the CAD document and part are revised during the same revision, the new versions have a new passive link between them, and the old versions maintain their existing link.

With wt.vc.struct.copyDescribeLinkOnRevise set to false, the system behaves as follows for the revision of a CAD document or part with a passive association:

- There is only one link and it is an iteration-to-iteration link.
- If only the CAD document is revised, the link is not carried forward. The new version of the CAD document has no association to the existing versions of the part.
- If only the part is revised, both the new and the old version of the part have a link to the existing version of the CAD document.
- If both the CAD document and part are revised during the same revision, then regardless of the settings in the user interface, revise acts as if the CAD document and part are being revised alone.

## Properties for New Versions

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

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

- For Windchill Foundation & PDM, the team and life cycle default to be the same as the original version.
- For Windchill Foundation & PDM, if there is a user interface, the user can override the default team and life cycle assigned to the new version.
- The view of new parts defaults to the same view as the original.

## Configuring the Synchronization of Revision Levels During Autoassociate

A key-value pair in the autoassociate.ini file (located at *<Windchill>/codebase/com/ptc/windchill/cadx/cfg/site/autoassociate.ini*) allows you to set the behavior for the revision of CAD documents and parts during the autoassociate operation.

The key `AutoAssociateSetRevisionForWTPart` can be set to the values `true` or `false`. With this property set to `true`, the revision of a `WTPart` is set to that of the actively associated CAD document during a revise action.

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

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

The following rules apply to both the `autoassociate` and `associate` (revising both part and CAD document) actions when `AutoAssociateSetRevisionForWTPart` is set to `true`.

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

If `AutoAssociateSetRevisionForWTPart` is set to `false`, `autoassociate` continues without trying to set a revision level.

## Enabling Display of Rename History and Location History

The display of Rename history and Location history can be configured by modifying entries in the configAudit.xml file, located at codebase\registry\auditing\configAudit.xml file. You should be aware that enabling auditing can adversely affect performance and consequently choose to audit the most significant events.

To enable display of the Rename History and Location History links on the information page, put the following entries in configAudit.xml and restart the method server:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EventConfiguration SYSTEM "configAudit.dtd">
<EventConfiguration enabled="true">
  <KeyEntry eventKey="*/wt.folder.FolderServiceEvent/POST_CHANGE_FOLDER"
multiObject="true"/>
  <KeyEntry eventKey="*/wt.epm.EPMDocumentManagerEvent/PRE_CHANGE_CAD_NAME"
multiObject="true"/>
  <ConfigEntry class="wt.epm.EPMDocumentMaster" enabled="true">
    <RenameEventEntry eventType="PRE_CHANGE_IDENTITY" enabled="true"
handler="wt.audit.configaudit.RenameEventRecorder"/>
    <KeyedEventEntry
eventKey="*/wt.epm.EPMDocumentManagerEvent/PRE_CHANGE_CAD_NAME" enabled="true"
handler="wt.audit.configaudit.CADNameChangeAuditEventRecorder"/>
  </ConfigEntry>
  <ConfigEntry class="wt.epm.EPMDocument" enabled="true">
    <KeyedEventEntry eventKey="*/wt.folder.FolderServiceEvent/POST_CHANGE_FOLDER"
enabled="true" handler="wt.audit.configaudit.LocationChangeAuditEventRecorder"/>
  </ConfigEntry>
</EventConfiguration>
```

**Note:** The default configAudit.xml file comes with EventConfiguration enabled="false", but all lower level sections enabled. Therefore, setting EventConfiguration enabled="true" enables all auditing for the application as a whole. To selectively audit events, you must customize which auditing events are enabled.

## 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.



## Using and Modifying the AutoAssociatePartFinderCreator Interface

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

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

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

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

The AutoAssociatePartFinderCreator interface supports the following methods:

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

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

Use the following procedure to implement a customized AutoAssociatePartFinderCreator:

1. Derive your customized class as follows:

```
public class CustomFinderCreator implements  
AutoAssociatePartFinderCreator
```

2. Override the following methods:

- public WTPart findOrCreateWTPart(EPMDocument epmDoc,  
EPMWorkspace workspace)

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

- public WTPart createNewWTPart(EPMDocument doc, String partNum ,  
String partName, .....)

This method is invoked for each document selected for auto associate to

create a new part. You can customize the properties of the newly created part. The newly created part is associated to the document by the auto associate action.

**Note:** The following methods are deprecated and not currently used by the action; however, you need to provide a dummy implementation of these methods to compile the class properly.

- public boolean isIsNewPart()
- public void setIsNewPart(boolean a\_IsNewPart)
- public WTPart findWTPart(EPMDocument epmDoc)

Compile the file and place the class in any appropriate location

3. Modify the following entry in the autoassociate.ini file (file location: *<Windchill>\codebase\com\ptc\windchill\cadx\cfg\site\autoassociate.ini*):

PartFinder=<class path>.CustomFinderCreator

4. Restart the method server

## Modifying the CAD Document IBA Value

In *<Windchill>\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 *<Windchill>\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 *<Windchill>*\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 HTML Client Item Selection Page

The HTML client item selection page is used in the Pro/ENGINEER Wildfire HTML client to allow the user to choose items in the Windchill database that are required to complete an action.

To determine the drop down list, search criteria, and result columns for the item 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 item selection page will be configured to support and determine the search criteria, and determine the result columns displayed in the item selection page. (For more SearchAttributeListDelegate details see [Customizing the HTML Search](#), later in this section.)

If this PickerSearchAttributeListDelegate implementation is not sufficient, then you can create and configure your own PickerSearchAttributeList to be used by the item 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 PickerSearchAttribute ListDelegate.

**Note:** If extending PickerSearchAttributeList, you may have to set the filter to avoid NullPointerExceptions. This issue will be addressed in a future release.

To configure a new PickerSearchAttributeListDelegate, use the xconfmanager to add an entry to com/ptc/windchill/cadx/common/picker/picker.properties similar to:

```
wt.services/svc/default/com.ptc.windchill.cadx.common.picker.PickerSearchAttributeListDelegate/<unique delegate id which is also specified for com.ptc.windchill.cadx.common.picker.pickerSearchAttributeList>/java.lang.Object/0=mine.MyPickerSearchAttributeList/duplicate.
```

Using the xconfmanager, change the pickerSearchAttributeList entry in the wt.properties to com.ptc.windchill.cadx.common.picker.pickerSearchAttributeList=<unique delegate id>. If there is no entry in wt.properties, then STANDARD is used as the delegate id.

## Modifying Type Lists

The Pro/ENGINEER Wildfire HTML client item selection page uses configured type lists identified by type list ids, which are specified as the item 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 item selection page these valid type list values are **query** values specified in wt.query.queryResource.

You can remove type ids from the list of type ids specified for a type list id, but you cannot remove an entry or leave the type list empty.

## Customizing the HTML Search

To customize the HTML search to either change the display of the default search objects or to add new classes, see the following file that is distributed as source Windchill\src\wt\query\SearchAttributeList.java. As explained in the javadoc for this class, subclass SearchAttributeList and make the appropriate entries in

service.properties and wt.properties. Following are methods that should be implemented in a custom SearchAttributeList, with examples:

```
public final class MySearchAttributeList extends SearchAttributeList implements
Externalizable {
public void setLocale( Locale locale ) {
// Load in the values for the drop down list for selecting what to search against.
clientLocale = locale;
// **Customize -----
-----
// Add new classes to search to list below.
// Make sure that they are assigned numbers in sequence from 0 to N.
// Set dropDownListCount to N+1.
final int ALL = 0;
final int WTPART = 1;
...
final int MYCLASS = 22
int dropDownListCount = 23;
// -----
-----...
pickList = new String[classCount];
pickList[ALL] =
WtMessage.getLocalizedMessage (RESOURCE,queryResource.ALL,null,clientLocale);
pickList[WTPART] =
WtMessage.getLocalizedMessage (RESOURCE,queryResource.WTPART,null,clientLocale);
...
pickList[MYCLASS] = WtMessage.getLocalizedMessage (RESOURCE,queryResource.
MYCLASS,null,clientLocale);
pickValues = new String[classCount];
pickValues[ALL] = queryResource.ALL;
pickValues[WTPART] = queryResource.WTPART;
...
pickValues[MYCLASS] = queryResource.MYCLASS;
// **Customize You will need a string in here to correspond to each item in
pickList
// The string is a space separated list of what classes to query
// against. If you want to query against multiple classes that have a common
parent that
// has all of the attributes that you are interested in use that one class. If
you want
// to query against multiple classes that don't have a good common parent then
you can
// add them to a list and the search will loop through each class and combine
the results
// at the end. All classes in one list must only search against COMMON
attributes or
// attributes with the same name and of the same class! If you add both a
parent and
// a child class to the list you will get duplicate entries, when the results
are
// combined duplicate entries are not deleted.
queryClass = new String[classCount];
queryClass[ALL] =
"wt.part.WTPart wt.doc.WTDocument wt.change2.WTChangeIssue
wt.change2.WTChangeRequest2 " +
"wt.change2.WTChangeInvestigation wt.change2.WTAnalysisActivity
wt.change2.WTChangeProposal " +
```

```

"wt.change2.WTChangeOrder2 wt.change2.WTChangeActivity2
wt.csm.businessentity.BusinessEntity " +
"wt.effectivity.ConfigurationItem wt.epm.EPMDocument " +
"wt.replication.unit.WTUnit " +
"wt.part.WTPProductConfiguration " +
"wt.part.WTPProductInstance2 "; // Please remember to keep a space at the
end so that conditionally added items work.

...
queryClass[WTPART] = "wt.part.WTPart";
...
queryClass[MYCLASS] = "??.?.MyClass";
// **Customize These are the
// attributes that can be queried against.
inputAttributes = new String[classCount];
inputAttributes[ALL] =
"number name lifeCycleState projectId cabinet creator modifier
modifyTimestamp";
inputAttributes[WTPART] =
"number name view versionIdentifier partType source lifeCycleState projectId
cabinet creator modifier modifyTimestamp";
...
inputAttributes[MYCLASS] =
"name modifyTimestamp";
// **Customize Each individual
// string must match with the string listed above for the inputAttributes. "0"
stands for no
// input processing. If an attribute is an enumerated type use "0" and the
code will generate
// the drop down list. In the first string: projectId is in the fourth
position in inputAttributes
// so the method to generate the drop down list for it is also in the fourth
position in the
// string. The "0"s and methods must match in number with the number of
attributes listed
// under inputAttributes. You may add a fully qualified method from your
customization package
// as long as it is static and returns a vector of strings.
inputProcessing = new String[classCount];
inputProcessing[ALL] =
"0 0 0 wt.query.LocalSearchProcessor.getProjectList
wt.query.LocalSearchProcessor.getCabinetList 0 0 0";
inputProcessing[WTPART] =
"0 0 0 wt.query.LocalSearchProcessor.getViewList 0 0 0 0
wt.query.LocalSearchProcessor.getProjectList
wt.query.LocalSearchProcessor.getCabinetList 0 0 0 0";
...
inputProcessing[MYCLASS] =
"0 0";
// **Customize This is similar in concept to inputAttributes only these are
the attributes
// that will be displayed in the search results.
outputAttributes = new String[classCount];
outputAttributes[ALL] =
"number name versionDisplayIdentifier displayType lifeCycleState projectId
modifyTimestamp";
outputAttributes[WTPART] =
"number name versionDisplayIdentifier projectId lifeCycleState

```

```

modifyTimestamp";
...
outputProcessing[MYCLASS] =
"ObjProps 0";
// **New for 6.0
// **Customize This is similar in concept to outputAttributes only this list
is used
// to indicate which attributes can be sorted, can't be sorted, or an alternate
attribute
// that can be sorted to have the same affect as the display attribute. The
string that is used
// here should be the column descriptor so that it can be used to create the
ClassAttribute for
// the query. The query that is used for search is a simple query that will
not sort on all
// of the display attributes. Changing the 0 to 1 for an unsupported attribute
will
// either cause exceptions or sorts that don't work. Attributes of the
following types are
// just some examples of the attributes that will either throw exceptions or
sort incorrectly:
// EnumeratedType, CabinetReference, DataFormatReference,
LifeCycleTemplateReference, ProjectReference,
// and ViewReference.
sortAttributes = new String[classCount];
sortAttributes[ALL] =
"1 1 versionInfo.identifier.versionId 0 0 01";
sortAttributes[WTPART] =
"1 1 versionInfo.identifier.versionId 0 0 1";
...
sortAttributes[MYCLASS] =
"1 1";
// **New for 6.0
// **Customize This is similar in concept to outputAttributes only this list
is used
// for assigning a unique key to the sort preferences for this search. This
string will
// be persisted and used to retrieve the sort preferences for users. If the
value of one
// of these strings is changed or deleted after the system is in operation it
will create orphaned
// preferences in the system and users will lose the value that they had
persisted for that
// search. New entries can be added when a new search is added so that sort
preferences
// can be saved for that new search. These strings are arbitrary and never
displayed to the user.
sortPref = new String[classCount];
sortPref[ALL] =
"all";
sortPref[WTPART] =
"wtpart";
...
sortPref[MYCLASS] =
"myclass";
}
/**

```

```

*
* <BR><BR><B>Supported API: </B>false
*
* @param locale
* @return MySearchAttributeList
**/
public MySearchAttributeList( Locale locale ) {
    setLocale(locale);
}
/**
*
* <BR><BR><B>Supported API: </B>false
*
* @return MySearchAttributeList
**/
public MySearchAttributeList() {
    return;
}
}

```

wt.query.SearchAttributeList is always the most up-to-date and should be used as a reference.

The remainder of this section describes two new arrays in wt.query.SearchAttributeList: sortAttributes and sortPref.

Due to the data structures used on some classes, not all attributes that can be displayed in search results are sortable in the search results. The sortAttributes array in wt.query.SearchAttributeList is used to designate which attributes are sortable, and if an alternate attribute should be used for sorting. The version attribute is an example of an alternate attribute used for sorting: the attribute used to display is versionDisplayIdentifier, but the attribute used to sort on is versionInfo.identifier.versionId. Base java types, such as String and int, are sortable. Use the examples in wt.query.SearchAttributeList to determine if any custom types are sortable. Otherwise, a simple test will show if the attribute works, has no effect, or throws an exception.

The sortPref array (shown in the preceding code) is used to define a sort preference base name so users can define their sort preferences for that “Search On” item. A default for the sort preferences should be defined at the system level so that the first time the user uses the system, or if a user never defines preferences, the columns will be sorted logically. A default can be defined using wt.load.LoadFromFile or by using the Preference Administrator editor from the System Administrator portal page

If this is a new database, the defaults are loaded as part of running the required section of wt.load.Demo (which runs wt.load.LoadFromFile). The site defaults can easily be added to or modified using the Preference Administrator. If the database was created on a system before Release 6.0, wt.load.LoadFromFile can be used to load the base defaults for the delivered configuration of the HTML search classes. See the “PrefEntry.../wt/query/htmlsearch” entries in Windchill\loadFiles\preferences.txt as examples.



Each user preference has an internal name, which is never seen from the client except in the Preference Administrator. Because the current search uses the `wt.query.SearchAttributeList` to allow users to add new searches, and because there has to be a set of sort preferences for each, a unique sort name is needed for each name in the "Search On" list. Each item in the "Search On" list is not necessarily one object, but can be a list of objects. The `sortPref` array in `wt.query.SearchAttributeList` defines a unique string that forms part of the name of the preference. The preferences for sorting are stored in the `/wt/query/htmlsearch` preference node, and the naming format is as follows:

`<sort preference base name>sortAttrib<#>`

`<sort preference base name>sortDirect<#>`

The `<sort preference base name>` is the unique string from the `sortPref` array in `wt.query.SearchAttributeList`; it has only to be unique within the sort names. The `sortAttrib` is for the attribute name, and the `sortDirect` is to indicate ascending or descending. It is false for ascending and true for descending. The `<#>` is the number of the sort key, 0 = first key, and so on. Following are the preferences that are loaded using `wt.load.LoadFromFile` and `Windchill\loadFiles\preferences.txt` for the All sort:

```
#All
```

```
PrefEntry~allsortAttrib0~number~/wt/query/htmlsearch
```

```
PrefEntry~allsortDirect0~false~/wt/query/htmlsearch
```

```
PrefEntry~allsortAttrib1~versionInfo.identifier.versionId~/wt/query/htmlsearch
```

```
PrefEntry~allsortDirect1~true~/wt/query/htmlsearch
```

In the all-default example, the results are sorted first by the number column and then by the version column, with the number being in ascending order and the version in descending order. Currently, the supported number of sort keys is 3, although theoretically the number of sort keys is limited only by Oracle performance. No testing beyond 3 keys has been done on the system.

## Defining the Rename Report Mail Server

The mail server host name should be defined in the `wt.properties` file under `<Windchill>/codebase` directory 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*

## Managing Secondary Content

The primary content of a CAD document is a CAD model file; however, Windchill allows you to attach other file types as secondary content. In addition, you can specify which file types should be considered outdated, and which should be automatically downloaded with a download of the primary content, as the CAD document moves through the stages (iterations, revisions, life cycle states, and so forth) of development.

### Specifying Whether or Not to Outdate Secondary Content

The file `<Windchill>/codebase/com/ptc/windchill/cadx/cfg/default/contentcat.ini` lists the secondary content categories for your site. For each category of file type, the preference `<secondary_content_category>.markoutofdate=true` means that upon iteration, revision, or state change of the CAD document, any secondary content of the specified category type is marked as outdated (for example, in the Attachments table on the CAD document information page). Setting the value to false means that the category does not become outdated as it is carried forward with the CAD document.

Users can manually override conflicts caused by outdated files (for example, during a Check In attempt) by:

- Aborting Check-in and manually updating the attachment, thus removing the “outdated” flag
- Removing the attachment from the CAD document
- Resetting the status (removing the Outdated flag)
- Overriding the conflict and checking the CAD document in “as is” -- with an Outdated status.

### Setting the preference to automatically download secondary content

The file `<Windchill>/codebase/com/ptc/windchill/cadx/cfg/default/contentcat.ini` lists the secondary content categories for your site. For each category of file type, the preference `<secondary_content_category>.download=true` means that upon iteration, revision, or state change of the CAD document, any secondary content of the specified category type is downloaded automatically when the primary content is downloaded. Setting the value to false specifies that the is not downloaded automatically with the primary content of CAD document.

## Managing Drawing Dependents

The large number of dependents that may be associated to drawings can affect performance if an unnecessarily large number of items are gathered into the workspace during collection activities.

To control this behavior, the property `TraceDrawingOptionalDependents` in the [dependency tracing] section of the `cadxhtmlui.ini` file allows you to specify whether or not optional dependents for drawings should be traced (and collected into the workspace). The possible values for the property and the resultant behavior are as followed:

- `PerConfiguration` -- Whatever rule the user specifies for configuration (that is, Dependents: All, Required or None) is honored by the collection action.
- `Required` -- If the user specifies All for dependents, only the required dependents are traced and collected (optional dependents are avoided). If the user specifies None, no drawings dependents are collected.

The default setting is `TraceDrawingOptionalDependents=PerConfiguration`

**Note:** This setting only applies to drawings that are included by collection; it does not apply to drawings that are "initially selected."

## Controlling the Display of Internal Pro/ENGINEER Relationships

In certain reports accessed through the CAD document details page, Windchill displays a column labeled Dependency Type. This column displays the type of link between the table item and the CAD document reported on, based on the internal relationship of their respective Pro/ENGINEER models. The following reports display in the Dependency Type column:

- Uses
- Where Used
- References
- Referenced By
- Used By

Your site may prefer to remove the display of certain internal Pro/ENGINEER relationships in Wildfire pages. For example, when viewing the References of an assembly drawing you can set a site-wide property in `<Windchill>/codebase/wt.properties` to display only the associated assembly, the Drawing Model (default behavior is to display all dependency types).

To restrict the view of assembly drawing references to the assembly itself, set `com.ptc.windchill.cadx.caddoc.excludeDependencyTypes=<value>` to `-1`.

Additional dependencies can be removed from display by adding other, comma-separated values as described in the following table:

Value	Dependency Type	Description
-2	Internal Pro/E Instance	Dependency linking item & hidden instance
-1	Internal Pro/E	Dependencies created by Pro/ENGINEER that are not visible to the user through the reference viewer. (e.g. displaying a dimension of a component of an assembly that is a model on a drawing.)
0	Internal Pro/E	Any dependencies
1	Layout Declared	Dependency from the layout to a model that has declared it
2	Membership	Component in an assembly
4	Drawing Model or Report	Model defined as a model of the drawing
8	Relation Reference	Relationship created between two items in an assembly
16	Drawing Format	Format on a drawing
32	Generic Model	Family Table Generic
64	Manufacturing Assembly	The assembly used for the .mfg file
128	Merge Part	Dependency from the case where a feature on a part is created due to a merge in an assembly
256	User Defined	User-defined dependency

## Clean-up of the Event Manager

To avoid possible performance issues resulting from an accumulation of a large number of event records in the Event Manager, add the following site-wide property to wt.properties:

```
com.ptc.core.task.purgeTasksOlderThanDays=5
```

Events older than the specified number of days are automatically purged from the Event Manager.

## Administering Table Views

The display of information in many tables is user- and administrator-definable using the **Table View Manager**. Specific views for tables can be created or edited in the **Create Table View** window, which is described in the workspace online help and the manual *Using Pro/ENGINEER Wildfire with Windchill*.

Administrative users have the option of making a table view available to all users by selecting the **Publish to all users** check box on the first step, **Set Name**, of the **Create Table View** window.

## Configuring Table Scrollbar Display

By default, scrollbars controlling the vertical scrolling for tables are positioned on the right side of tables. Occasionally, the combined number and width of table columns may make it necessary to use the horizontal scrollbar to access the vertical scrollbar. A site-wide property allows you to change the location of table scrollbars from the right to the left side to reduce the need for horizontal scrolling.

To position scrollbars on the left side of tables, configure the following property in your site.xconf:

```
<Property name="com.ptc.core.htmlcomp.jstable.positionScrollbarLeftside"
overridable="true" targetFile="codebase/wt.properties" value="true"/>
```

**Note:** No property is necessary for default behavior. This property works for common table components on the Windows platform (only) except for the Event Manager tables.

## Configuring the Number of Workspace Rows Displayed

The property `com.ptc.windchill.uwgm.cadx.ws.sizeToWindow` controls the number of rows displayed in the workspace item list. Set to true (the default setting), the number of rows shown is based on the height of the window.

To display a fixed number of rows (for example, 10 for Windows), set the property to false, as follows:

```
<Property name="com.ptc.windchill.uwgm.cadx.ws.sizeToWindow"
overridable="true" targetFile="codebase/wt.properties" value="false"/>
```

## Configuring Automatic Scrolling in the Workspace

The property `com.ptc.windchill.uwgm.cadx.ws.scrollToTable` controls whether or not the workspace page appears automatically scrolled to the beginning of the item list table or appears scrolled to the top of the page. Set to true (the default setting), the page automatically scrolls to show the workspace item list.

To disable automatic scrolling, set the property to false, as follows:

```
<Property name="com.ptc.windchill.uwgm.cadx.ws.scrollToTable"
overridable="true" targetFile="codebase/wt.properties" value="false"/>
```

# 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.

## Using External File Vaulting

Content files persisted in external vaults are retrieved faster than content files stored in Oracle as binary large objects (BLOBS).

Although use of file vaults can add complexity to backup and recovery operations, vault management can be simplified by using the xconfmanager to set the `wt.property wt.fv.forceContentToVault = true`. This causes all content to vault to the DefaultCacheVault, keeping it out of Oracle BLOBs, without requiring creation of a vaulting rule.

In the event that multiple vaults must be implemented at your site, a vaulting rule applied to the User domain (where EPMDocuments are created) can direct content to vault appropriately.

**Note:** Following a custom checkin, the user is able to see CAD documents to be vaulted only in the default cache folder until an explicit revaulting action (executed through the replication schedule set by the administrator) is executed.

For more information on external vaulting and vaulting rules see the chapter "Administering External File Vaults" in this guide.

## Using Content Replication

Content replication provides the means to copy selected content files from a master server to remotely located replica servers for faster access by users at the remote site, thereby significantly improving access time. The files at the replica site remain retrievable by users at the master site.

For more information, see the chapter "Installing and Administering Content Replication" in this guide.

# 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 chapter "Method Server Maximum Heap Size" in the *Workgroup Manager Performance Best Practices Guide*

## Data Compression

The meta data compression option is intended to improve the upload and download performance of the 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.

### Pro/ENGINEER Settings

In Pro/ENGINEER Wildfire, compression is controlled by a Pro/ENGINEER config.pro setting (dm\_http\_compression\_level) as follows:

dm\_http\_compression\_level *<an integer between 0 and 9 -- 0 for no compression, 9 for maximum compression>*

### Windchill Settings

On the Windchill side, you enable the compression filters provided by the web-servers (for example, mod\_gzip for Apache1.3.x and mod\_deflate for Apache2.0.x).

### Additional SOAP Compression Filter

Additionally, out of the box, the Windchill SOAPCompressionFilter is configured for compressing HTTP response data for special client (for example, Pro/ENGINEER Wildfire) interactions, such as downloading the contents of a model.

Following additional property settings that control data compression behavior are applicable only to the SOAPCompressionFilter.

To use these property settings, add them to the wt.properties file:

- wt.compression.threshold=<size\_in\_bytes> -- Sets a threshold for which HTTP responses are to be compressed. The default value (0) specifies that all responses are compressed.
- wt.compression.off.contentEncodings=<encoding\_types> --Identifies HTTP response encoding types (case insensitive) for which compression is switched off. The default encoding types are (space delimited): identity gzip deflate lws-deflate. Setting the value to asterisk (\*) switches off compression for all encoding types.
- wt.compression.off.contentType=<content\_types> --Identifies HTTP response content types (case insensitive) for which compression is switched

off. The default content types are (space delimited): image/jpeg image/gif application/zip.

**Tip:** While data compression can provide a benefit in a slow network, using compression puts an extra load on CPU resources; therefore, if network speed is not an issue, the use of compression may decrease performance and is not recommended.

## Additional Considerations

If the Windchill compression filter is configured and `dm_http_compression_level` preference is set in the Pro/ENGINEER config file (`config.pro`), this setting will also apply to any interaction between the Pro/ENGINEER embedded browser and the server. That is, a non-zero value of the preference will ensure that not only the meta-data of Pro/ENGINEER models but even the content/UI pages will be sent in the compressed form reducing the overall network traffic.

Also note that the Pro/ENGINEER configuration option `dm_http_compression_level` needs to be set before registering the server through Pro/ENGINEER Wildfire. Any change in the value after the server is registered, will not apply to already registered server(s).

## Maximizing the Oracle Server/Windchill Method Server Connection

Due to the large number of items and CAD documents involved in database transactions, it is highly recommended that the connection between the Oracle server and the Windchill method server machines is both low-latency and high-bandwidth.

**Note:** Bulk HTTP data transfer using Apache on Windows 2000 can be restricted by Apache's default send buffer size. We found that setting property `SendBufferSize=16384` in `httpd.conf` significantly improved throughput over high latency, low bandwidth WANs.

## Choosing to Display Family Item Symbols in Folders Table

Showing or hiding the family table symbols on CAD document type icons in the **Folders** table can be controlled by the following property:

`wt.clients.showFamilyGlyph`

Because there is a significant performance benefit in bypassing the queries that determine whether or not to show the family table symbols, the property defaults to false OOTB. Therefore, family table symbols on CAD document type icons do not appear by default in the **Folders** table.

To show the family table symbols in the **Folders** table, set `wt.clients.showFamilyGlyph` to true using the `xconfmanager`.



## Other Recommendations

### Online Java Performance Guide

You may want to review the online Java Performance Guide to identify server-side Java settings that can boost performance.

**Note:** Be sure to carefully evaluate the options prior to implementation. PTC does not currently support them.

For more information on the online Java Performance Guide, see the chapter "Online Java Performance Guide" in the *Workgroup Manager Performance Best Practices Guide*

### Windchill Folder Structure

It is important to carefully plan the Windchill cabinet/folder structure, and direct Windchill users to keep the number of items (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 item 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

Key	Value(s)	Description
<b>[general] section:</b>		
DefaultApplication=	PROE	
DefaultDateFormat=	ShortDateFormat1 (recommended), or LongDateFormat ShortDateFormat2 DateOnlyFormat1 DateOnlyFormat2	Value of DefaultDateFormat should be one of the formats defined in the [dateformat] section
SetTargetRevision=	true (default) false	If set to true, allows setting of a revision level higher than the next in the series
<b>[newworkspace] section:</b>		
AdditionalValidCharacters_en_US=		Defines valid characters in workspace name. (All alphanumeric characters are valid, so there is no need to define them.)
partCentric=	false (default) true	Specifies part or document centric dependency processing mode  If set to false, non-part centric (CAD document centric) processing is specified.
<b>[editworkspaceoptions] section:</b>		
effectivityConfigSpecForDocsActive=	false (default) true	Indicates whether effectivity configuration specification is applicable to EPMDocument items for a specified workspace.
<b>[dateformat] section:</b>		
LongDateFormat=	3	Date format as yyyy-MM-dd HH:mm:ss z,  For example, 2002-05-15 13:25:58 EST.

ShortDateFormat1=	4	Date format as dd MMM yy HH:mm, For example, 15 Mar 02 13:25.
ShortDateFormat2=	19	Date format as M/dd/yy h:mm a, For example, 3/15/02 1:25 PM.
DateOnlyFormat1=	20	Date format as M/dd/yy, For example, 3/15/02.
DateOnlyFormat2=	22	Date format as M/dd/yyyy For example, 3/15/2002.
<b>[naming/numbering policies] section:</b>		
PROENamingParameter=	<Some String Parameter>	Specifies the Pro/ENGINEER designated parameter from which the CAD document name is copied during initial upload
PROENumberingParameter=	<Some String Parameter>	Specifies the Pro/ENGINEER designated parameter from which the CAD document number is copied during initial upload
AutoAssociateNamingParameter =	<Some String Parameter>	Specifies the Pro/ENGINEER designated parameter from which the WTPart name is copied during auto-associate
AutoAssociateNumberingParameter=	<Some String Parameter>	Specifies the Pro/ENGINEER designated parameter from which the WTPart number is copied during auto-associate
UploadDropNameFileExtension =	true false	When set to true (default is false), specifies dropping the file extension for the upload action when naming new items
UploadDropNumberFileExtension=	true false	When set to true (default is false), specifies dropping the file extension for the upload action when numbering new items
AutoAssociateDropNameFileExtension=	true false	When set to true (default is false), specifies dropping the file extension for the auto-associate action when naming new items

AutoAssociateDropNumberFile Extension=	true false	When set to true (default is false), specifies dropping the file extension for the auto-associate action when numbering new items
<b>[modelcheck] section</b>		
NumErrors=	<integer>	Maximum number of ModelCHECK errors allowed. Set to zero by default
NumHours=	<integer>	Number of hours since the last ModelCHECK run. Set to 24 by default
MCConfig=	<comma separated string>	Type of configuration used followed by actual set of MC Configuration files to be used for validation. Set to "Medium" configuration by default
<b>[caddoc Naming policy] section</b>		
SynchronizeCadCocNameAndNumber=	true false	Applies to the Create CAD Document user interface.  Set to true, specifies that CAD Model Name is synchronized with CAD Doc Number in the Create CAD Document user interface.  Note: This synchronization is not applicable when auto numbering is in effect
<b>[ghost policies] section</b>		
GhostIgnoreDependencyOption =	IgnoreOptionalDependencies (default)  IgnoreOptionalReferenceDependencies  IgnoreProeInternalDependenciesOnly  DoNotAllowIgnore	Specifies the policy for creation of ghost item for upload and check-in
<b>[dependency tracing] section</b>		

TraceDrawingOptionalDependents=	PerConfiguration (default)  Required	Specifies if optional dependents for drawings should be traced  If set to PerConfiguration, whatever the user specifies for configuration (Dependents: All, Required, or None) is honored  If set to Required, if the user specifies All for dependents, only the required dependents are returned. If the user specifies None, no drawings dependents will be returned
---------------------------------	---	---

## Contentcat.ini File

Key	Value(s)	Description
<b>[content_category_prefs] section: Do Not Modify Any Values in This Section</b>		
ANALYSIS_INPUT.download=	true	
ANALYSIS_INPUT.markoutofdate=	false	
ANALYSIS_RESULTS.download=	false	
ANALYSIS_RESULTS.markoutofdate=	false	
DRAWING.download=	true	
DRAWING.markoutofdate=	true	
EXPORT.download=	false	
EXPORT.markoutofdate=	false	
FAMILY_TABLE.download=	true	
FAMILY_TABLE.markoutofdate=	true	
GENERAL.download=	false	
GENERAL.markoutofdate=	false	
I-DEAS_DRAWING_SHEET.download=	true	
I-DEAS_DRAWING_SHEET.markoutofdate=	true	
IDEAS_PACKAGE.download=	true	

Key	Value(s)	Description
<b>[content_category_prefs] section: Do Not Modify Any Values in This Section</b>		
IDEAS_PACKAGE.markoutofdate=	false	
IMPORT.download=	false	
IMPORT.markoutofdate=	false	
INFORMATION.download=	false	
INFORMATION.markoutofdate=	false	
INVENTOR_DESIGN_VIEW_DOCUMENT.download=	true	
INVENTOR_DESIGN_VIEW_DOCUMENT.markoutofdate=	false	
INVENTOR_IPART_INSTANCE.download=	true	
INVENTOR_IPART_INSTANCE.markoutofdate=	false	
IMAGE.download=	false	
IMAGE.markoutofdate=	false	
INSTANCE_ACCELERATOR_FILE.download=	true	
INSTANCE_ACCELERATOR_FILE.markoutofdate=	false	
LOGICAL_REFERENCE.download=	true	
LOGICAL_REFERENCE.markoutofdate=	false	
MANUFACTURING.download=	true	
MANUFACTURING.markoutofdate=	false	
MESH.download=	false	
MESH.markoutofdate=	false	
PACKAGE.download=	true	
PACKAGE.markoutofdate=	true	
PARAMETER_TABLE.download=	true	

Key	Value(s)	Description
<b>[content_category_prefs] section: Do Not Modify Any Values in This Section</b>		
PARAMETER_TABLE.markoutofdate=	false	
PROE_UGC.download=	true	
PROE_UGC.markoutofdate=	true	
PROE_UGC_SECTION.download=	true	
PROE_UGC_SECTION.markoutofdate=	true	
PROE_UGC_SECTION_TOC.download=	true	
PROE_UGC_SECTION_TOC.markoutofdate=	true	
TOOLPATH.download=	false	
TOOLPATH.markoutofdate=	false	
VIEWABLE.download=	false	
VIEWABLE.markoutofdate=	true	

## Newdocument.ini File:

Key	Value(s)	Description
<b>[general] section:</b>		
okToDownloadAlreadyCheckedOut=	true false	Allows download of an already checked out item.
DefaultDocFolder=	/<<Name of the folder>>	The destination folder where a newly created CAD document may be moved upon checkin.
isModelNameUnique=	true	Uniqueness constraint automatically set to true by Pro/ENGINEER.  <b>Note:</b> This value should not be changed.
<b>[proe] section:</b>		
proe.files.component.ext=	.prt	Valid Pro/E model file name extension (no specific extension for "other")
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	
CADComponentDocNumber=	00008_comp_1.prt	Defines numbering convention for document.  [Replace existing values with examples of your chosen convention.]
CADAssemblyDocNumber=	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=	00008_mrk.mrk	
ReportDocNumber=	00008_rep.rep	
SketchDocNumber=	0008_sec.sec	
OtherDocNumber=	00008_other.oth	Defines numbering convention for “other” type of document

## Site Directory INI Files

The following tables list key-value pairs for files in the Site directory:

### Autoassociate.ini File

Key	Value(s)	Description
<b>[general] section:</b>		
SearchForPartAttribute=	PARTNUMBER	<p>The value of this preference setting is an IBA of the CAD Document. The value of the IBA of the CAD Document gives the number of the part to be searched.</p> <p>The value of IBA PARTNUMBER is read programmatically to get the number of part to be searched in database. This preference is handled by the search delegate class.</p>
SearchPartMasterClass=	wt.part.WTPartMaster	<p>Enables search for customized parts. This preference specifies 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 is not searched.</p> <p>The value wt.part.WTPartMaster indicates that the search for part is done on WTPart class.</p>
AutoAssociateSetRevisionForWTPart=	false (default) true	If set to true, during autoassociate the system attempts to set the WTPart revision level to be the same as the level of the CAD document.

## Newdocument.ini File

Key	Value(s)	Description
<b>[general] section:</b>		
isModelNameUnique=	true	Uniqueness constraint automatically set to true by Pro/ENGINEER.  <b>Note:</b> This value should not be changed.

# A

## Windchill Runtime Environment

Topic	Page
Web Infrastructure.....	A-2
Java Platform Support .....	A-2
Three-Tier Architecture.....	A-3
Client Software Components.....	A-4
Server Software Components.....	A-6
Database Components .....	A-18
Full Text Retrieval Indexing Components .....	A-21

## 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.

For information on the Web capabilities available to you through the Windchill architecture, see the following guides:

*Windchill Info\*Engine Administration and Implementation Guide*

*Info\*Engine User's Guide*

*Windchill Adapter Guide*

## 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.

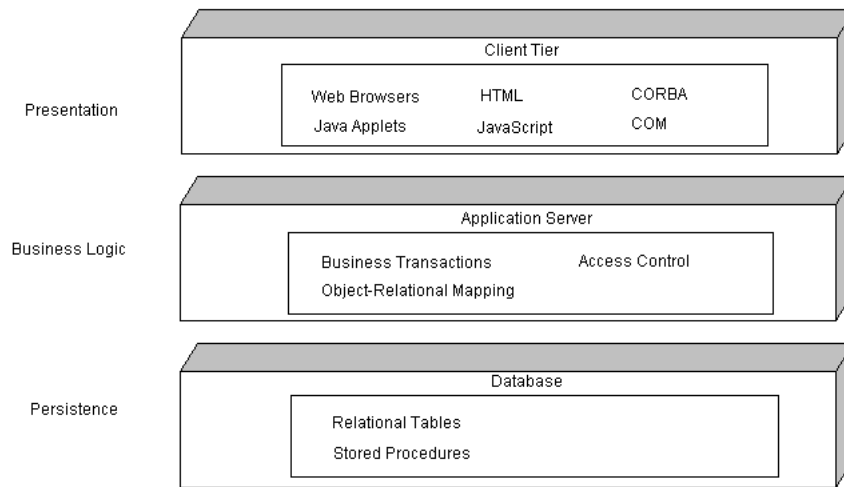
*Internationalization* is the process of designing and developing an application that can be adapted to the culture and language of a locale other than the one for which it was originally developed. Java facilitates this task by providing classes that convert dates and numbers to formats conforming to local conventions, and by 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.

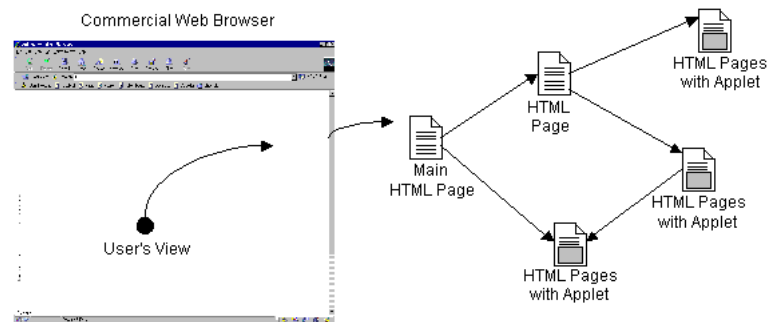
The ability to display HTML pages, although adequate for simple applications, does not provide enough functionality for all aspects of complex information authoring applications. Therefore, Windchill requires a browser capable of hosting Java applets based on the Java runtime and base classes. Two popular examples are Netscape Communicator and 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.



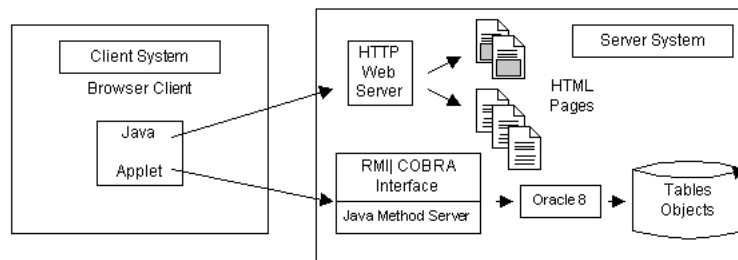
The Windchill system responds with an HTML page. This page may contain JavaScript or JScript to coordinate window or frame usage within the browser.

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

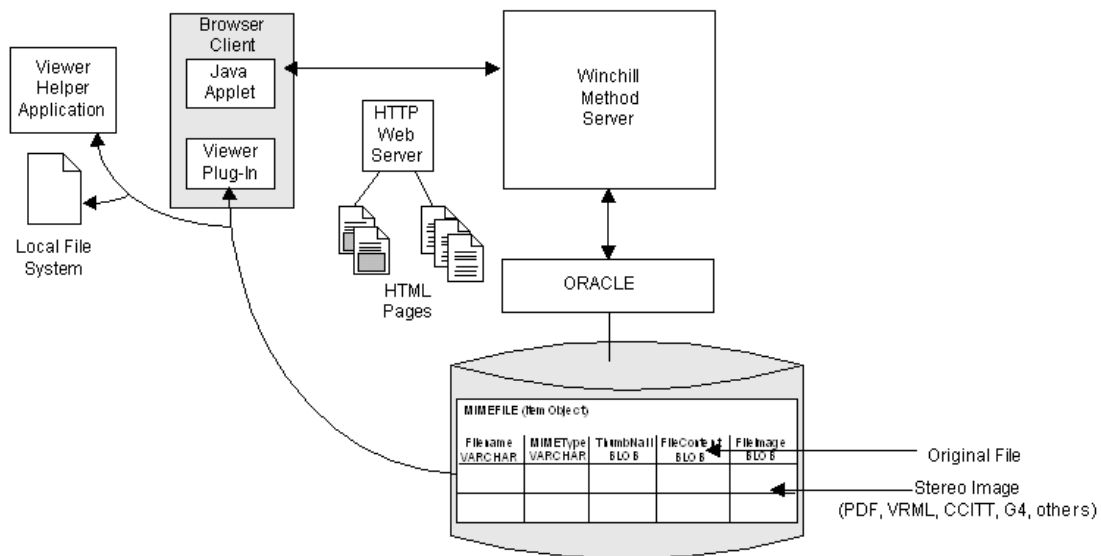
Files are transferred from the client to the server using a chunked RMI upload. The file is split into manageable pieces and then sent to the server where it is reconstructed and inserted into persistent storage. This capability is only accessible to applet clients and is available as a standard bean 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.

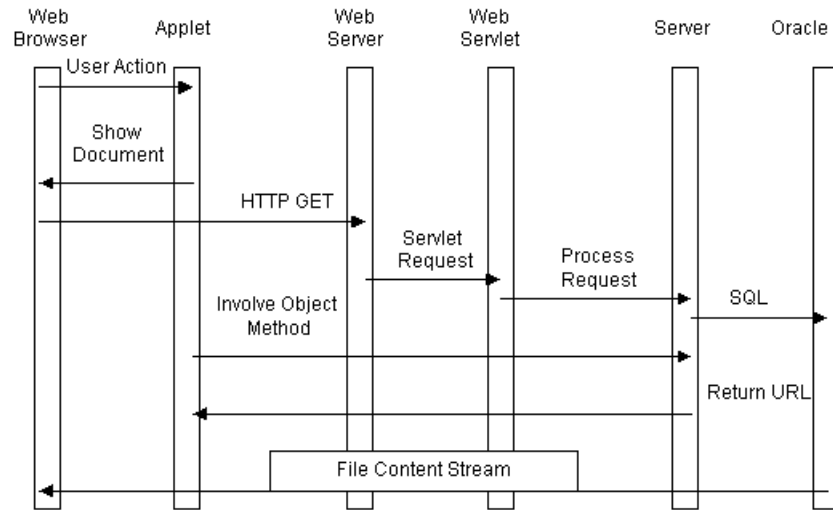
## File Transfer Using HTTP

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.



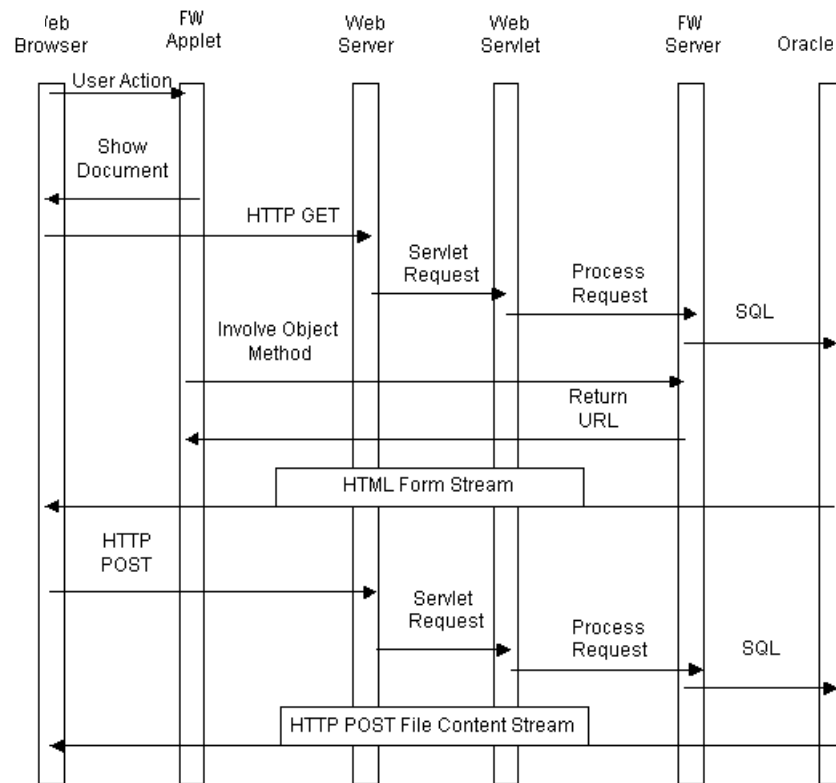
In the Windchill method server, HTTP responses are generated using a streaming interface, allowing the responses to be arbitrarily large. As shown below, this is accomplished by invoking the method to generate the response from within the RMI reply marshaling so the response can be written directly to 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 a 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.

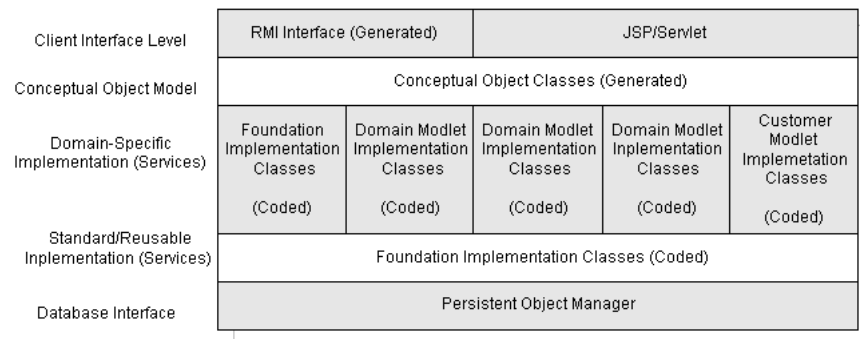
## System Management

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 architectural 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 a database 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.

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 database server to store data objects. In order to maintain the integrity of the associations among stored objects, users and administrators should avoid using vendor tools 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.

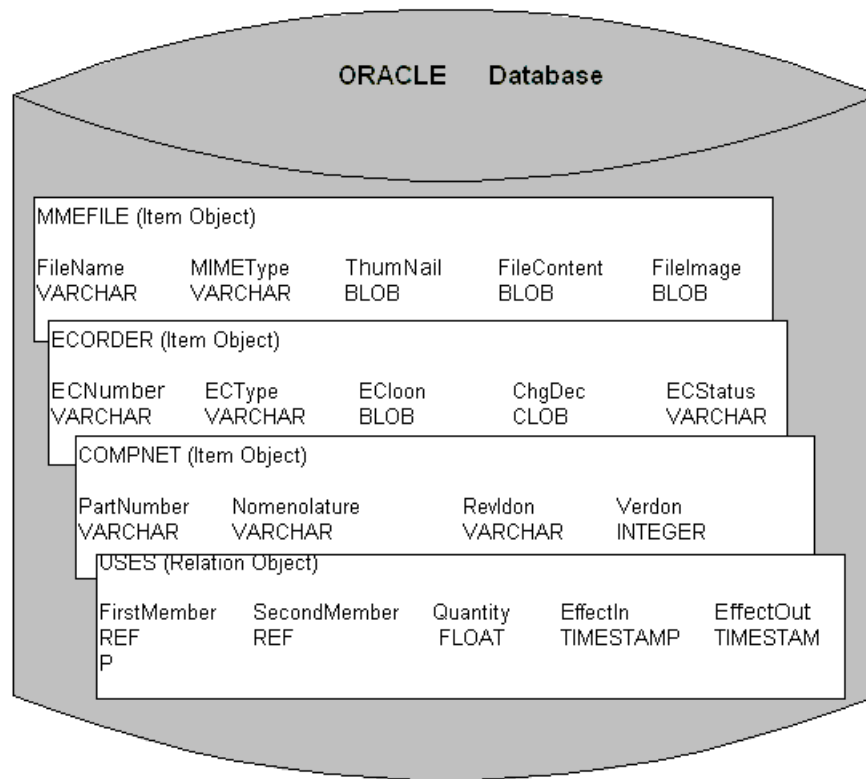
## Single Logical Database

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

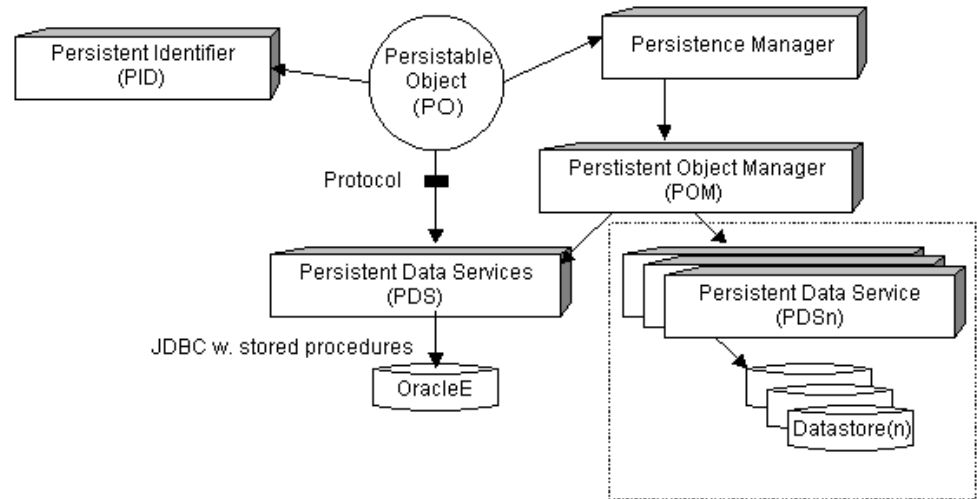
proprietary data formats. The following figure illustrates ORDBMS management of structured and unstructured attributes.



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 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 database 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 *Windchill Installation and Configuration Guide - RetrievalWare*. 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.

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 IndexDelegate objects.

## Indexing Policies

*Indexing policies* determine what objects require indexing. They provide administrative control over indexing by associating indexable objects to IndexDelegate objects. Indexing policies are similar in concept to access control policies.

Essentially, indexing policies are an association between indexable objects and a set of libraries 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*.

## IndexDelegate Objects

For each Windchill system publishing to a RetrievalWare index, there is an associated instance of an IndexDelegate object. The IndexDelegate object acts as an adapter between the indexable objects within the Windchill system and the given RetrievalWare index. The IndexDelegate 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.

IndexDelegate 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.

IndexDelegate objects perform their work in the background and execute as the Administrator.

## 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 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.



# B

## Windchill Considerations for Security Infrastructures

Topic	Page
Overview .....	B-2
Protocols .....	B-2
Authentication .....	B-3
URL Generation .....	B-4
RMI .....	B-7
Configuration Considerations.....	B-10

## 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:

Client	Example	Communicate Protocols	Comments
Browser with pure HTML user interface	Local search page; properties page; Dynamic Client Architecture	HTTP or HTTPS	
Java clients, in the form of Java applets in HTML pages	Windchill Explorer; Product Structure Explorer	Java RMI and HTTP or HTTPS	Java applet or application HTTP requests are performed via the <code>java.net.URLConnection</code> class.
Stand-alone Java applications	Workgroup Manager for Pro/E; Workgroup Manager for CADDs	Java RMI and HTTP or HTTPS	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.

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 database servers using JDBC calls.
- 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 is 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.

## 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 `<BASE>` 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 URL 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.

**Note:** The host name of the computer must conform to the required standard Internet format that specifies the name can be a text string up to 63 characters drawn from only the alphabet (A-Z and a-z), digits (0-9), hyphen (-), and period (.). The period is only used as a domain name separator. The first character of a host name can be either a letter or a digit.

Using unsupported characters in the host name (such as an underscore or a comma) or supplying a host name that is longer than 200 characters causes problems in your Windchill system.

**Note:** When creating a host name alias, always use lower case characters. If the host name has no upper case characters, its IP address can be read from a local cache. Using the local cache is the fastest method of getting the IP address.

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.

Contact your system administrator to learn about the best way to mitigate problems with.

## 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 `RMIConnectionFactory`, 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.WrappedRMHttpToPortSocketFactory`).
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.WrappedRMHttpToCGISocketFactory` 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, which allows 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.

## 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. Using HTTPS for Windchill 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. No additional Windchill configuration is needed.

## 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 host name 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, standalone 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. For example, if the host machine is `proxy.acme.com` and the port is 8080, then include the following:

```
-Dhttp.proxyHost=proxy.acme.com -Dhttp.proxyPort=8080
```

## Server-Side Reverse Proxy Servers

Setting up your environment to use a reverse proxy server involves the use of two host machines:

- On one host, set up a Web server to use as the proxy Web server. For example, install Apache on the server1 host named `server1.mycompany.com`.
- On the second host, set up Windchill. For example, install Windchill on the server2 host named `server2.mycompany.com`.

There are unique tasks to complete on each host:

- On the Windchill server host (server2), update the following:
  - wt.properties as directed in [Updating wt.properties on the Windchill Host](#).

- web.xml as directed in [Updating web.xml on the Windchill Host](#).
- On the proxy server host (server1), update the proxy Web server httpd.conf file as directed in [Updating httpd.conf on the Proxy Server Host](#).

Additionally, you can set up the reverse proxy server for secure access using HTTPS as described in [Using HTTPS with a Reverse Proxy Server](#).

## Updating wt.properties on the Windchill Host

**Note:** When updating the wt.properties file, use the xconfmanager utility from within a windchill shell. For information on using this utility, see [Using the xconfmanager Utility](#).

Using a reverse proxy server 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 correctly. To accomplish this, set the following properties in wt.properties:

```
wt.rmi.clientSocketFactory=wt.boot.WTRMIMasterSocketFactory
wt.rmi.serverSocketFactory=wt.util.WrappedRMISocketFactory
wt.rmi.javarmicgi=servlet/JavaRMIServlet
```

To make the location of the default repository available to DCA in a reverse proxy setup, set the following property in wt.properties:

```
com.ptc.core.ca.co.client.doer.task.default.repository=<Windchill_host_name>
```

To name the SOAP service that DCA uses for interactions in a reverse proxy setup, set the following property in wt.properties:

```
wt.federation.rpc.endpoint=http://<Windchill_host_name>:<task_dispatcher_minPort>
/<web_app_name>/servlet/RPC
```

where <task\_dispatcher\_minPort> is the port number set in the following wt.properties property:

```
wt.adapter.simpleTaskDispatcher.minPort
```

The Windchill host name should resolve to the reverse proxy server for clients that are required to access through the reverse proxy. To generate outgoing Windchill URLs referencing the reverse proxy server, you must set the following properties in the wt.properties file. These properties map the Windchill server codebase to the reverse proxy codebase:

```
wt.server.codebase=http://<proxy_server_host_name>[:<port>]/<web_app_name>
wt.httpgw.mapCodebase=http://<Windchill_host_name>[:<port>]/<web_app_name>
```

## Reverse Proxy wt.properties Example

Assume the following statements are true:

- The fully-qualified reverse proxy host name is server1.mycompany.com.

- The fully-qualified Windchill host name is server2.mycompany.com.
- The Web servers on both hosts use the default port.
- The value in the wt.adapter.simpleTaskDispatcher.minPort property (<task\_dispatcher\_minPort>) is the default value of 18080.
- The Web application name (<web\_app\_name>) defined when Info\*Engine was installed on the Windchill server host is Windchill.

Then, the wt.properties properties to set using the xconfmanager utility are as follows:

```
wt.rmi.clientSocketFactory=wt.boot.WTRMIMasterSocketFactory
wt.rmi.serverSocketFactory=wt.util.WrappedRMISocketFactory
wt.rmi.javarmicgi=servlet/JavaRMIServlet
com.ptc.core.ca.co.client.doer.task.default.repository=server2.mycompany.com
wt.server.codebase=http://server1.mycompany.com/Windchill
wt.httpgw.mapCodebase=http://server2.mycompany.com/Windchill
wt.federation.rpc.endpoint=http://server2.mycompany.com:18080/Windchill/servlet/RPC
```

## Updating web.xml on the Windchill Host

Add the following XML to the end of the web.xml file that is in the codebase/WEB-INF directory. The value to use for minPort in the XML is the value of the wt.manager.port property in the wt.properties file; the value to use for maxPort is the value of the wt.method.maxPort property. In the following, the minPort is set to 5000 and the maxPort is set to 5009:

```
<servlet>
  <servlet-name>JavaRMIServlet</servlet-name>
  <description>RMI over HTTP proxy</description>
  <servlet-class>wt.tools.javarmi.JavaRMIServlet</servlet-class>
  <init-param>
    <param-name>minPort</param-name>
    <param-value>5000</param-value>
  </init-param>
  <init-param>
    <param-name>maxPort</param-name>
    <param-value>5009</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>JavaRMIServlet</servlet-name>
  <url-pattern>/servlet/JavaRMIServlet/*</url-pattern>
</servlet-mapping>
```

## Updating httpd.conf on the Proxy Server Host

Remove the comment characters from the following lines in the httpd.conf file:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Add following lines at the end of the httpd.conf file:

```
ProxyPass /<web_app_name>/ http://<Windchill_host_name>/<web_app_name>/
ProxyPassReverse /<web_app_name>/ http://<Windchill_host_name>/<web_app_name>/
```

Where the values for *<web\_app\_name>* and *<Windchill\_host\_name>* are the same values you entered in the [Updating wt.properties on the Windchill Host](#) section.

## Using HTTPS with a Reverse Proxy Server

Typical use of a reverse proxy server requires all incoming traffic to be HTTPS. 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. The only change to the steps provided in previous sections is to change the URL in the wt.server.codebase property to use HTTPS.

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 can 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.



# C

## Import and Export Policies, Mapping Rules, and Conflict Messages

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.

Topic	Page
XSL-based Policy Files .....	C-2
Mapping Rules .....	C-4
Mapping Through Special Rules .....	C-4
Mapping Through XSL Transformation .....	C-11
Java Mapping with the METHOD Element .....	C-12
Hierarchical Instance Based Attribute Definitions, Exporting, and Importing	C-12
Conflict Messages .....	C-16
Reforming Custom Modeled Attributes .....	C-24

## 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 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 item
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\registry\ixb\dtds\standardX05.dtd\coreobjects.dtd'.

```
-->

<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** check box 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. If you are not setting properties through a graphical user interface or in a mapping file, you can 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 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
Data
  import.reposGuidPrefix=77746
  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>
```

```

    <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>

</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
Data

import.reposGuidPrefix=77746

logLevel=5

debug.enable=true

mappingRules.log.enable=false

mappingRules.debug.dir=C:\\TUNER_RESU

  <COPY_AS>
    <tag>number</tag>

```

```

<value>*</value>
<newValue>444-@@</newValue>
</COPY_AS>

<IGNORE_MASTER>
<path>content</path>
<tag>filename</tag>
<value>EngineReq</value>
</IGNORE_MASTER>

```

## 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 item as an attempt to duplicate an existing item.

## 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"

```
<COPY_AS>
  <tag>view</tag>
  <value>Source_View</value>
  <newValue>Local_View</newValue>
</COPY_AS>
```

### Mapping any Object's View to View "LOCAL\_VIEW"

```
<COPY_AS>
  <tag>view</tag>
  <value>*</value>
  <newValue>Local_View</newValue>
</COPY_AS>
```

### Mapping an Object's Number Attribute "2222" to Number "LOCAL\_2222"

```
<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.

```
<COPY_AS>
  <tag>number</tag>
  <value>*</value>
  <newValue>From_Site_AAA_*</newValue>
</COPY_AS>
```

### Mapping an Object's Version "A" to Version "B"

```
<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 item 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"?>
```

```

<userSettings>
  <properties>
    xsl.filename=C:\\script1.xsl
  </properties>
</userSettings>

```

## 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:

```

static public String mapElement (String path, String tag, String
oldValue, IxbElement oldXmlFile) throws WTEException;

```

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:

```

<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

Beginning in Windchill 7.0, it is suggested that you do not create hierarchical IBA definitions unless the following line is present in the wt.properties file:

```

wt.iba.definition.hierachicaldefinition.enabled=true

```

Setting the preceding property's value true allows the import of hierarchical IBA definitions.

By default, since Windchill 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.

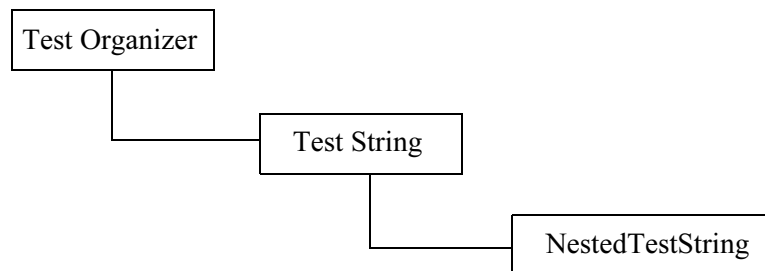
The creation of hierarchical IBA definitions without having the property set true is likely to have occurred in releases prior to Release 7.0 because no recommendation to set the property true existed for those releases. Nested Attribute Organizers were allowed in Release 7.0 as they were in prior releases, 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 than with Windchill, because Windchill 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:



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:

```
<COPY_AS>
  <tag>path</tag>
  <value>TestOrganizer/TestString</value>
```

```

        <newValue>TestOrganizer</newValue>

    </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:

```

    <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`:

```

<COPY_AS>

    <path>ibaDefinitions/StringDefinition</path>

    <tag>path</tag>

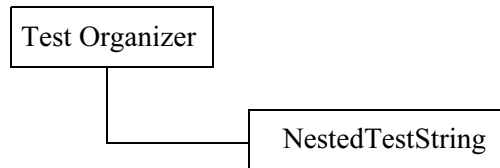
    <value>TestOrganizer/TestString</value>

    <newValue>TestOrganizer</newValue>

</COPY_AS>

```

The altered mapping rule would create the following `StringDefinition` in the database if it did not already exist:



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:

```

<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>
</userSettings>
```

```
#logLevel=4

</properties>

</userSettings>
```

## 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 items being imported exist already in the Windchill database, and the item properties of the imported and existing items do not match. In this explanation, the term Windchill item refers to Parts, Documents, and EPMDocuments.

The following matrix lists the properties of Windchill items 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 item, it is a costly approach for importing a large number of items.

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 item and the data definitions which exist in the target Windchill environment.
- *Dependency Conflicts* -- References in the imported item (for example, through part structure) to other business items that do not exist in the target system.
- *Metadata conflicts* -- Mismatches between the metadata of items (for example, name/number pair) in the target system with the metadata of imported item 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 that occurred in Release 7.0 that could produce conflicts involves the RatioDefinition and RatioValue. These types of data, if included in an export from Windchill 6.2.6 or earlier, result in an overridable import conflict in Windchill 7.0 and later. If you override the conflict, the data is imported as FloatDefinition and FloatValue.



## Administrative Conflicts of Common Business Objects

Potential conflict	Behavior	Resolution or Message
IBA name mismatch	User notification	"Definition of Instance Based Attribute <type> cannot be found. See the <i>Windchill Administrator's Guide</i> for further information."
IBA datatype mismatch	User notification	"Object <type> already exists in database, but has different type of Instance Based Attribute: existing type is <type>, new type is <type>."
IBA units of expression mismatch	User notification	"Object <type> already exists in database, but has different value for Instance Based Attribute: existing value is <type>, new value is <type>."
Denominator mismatch for Ratio values	User notification	Object <type> already exists in database, but has different type of Instance Based Attribute <type>: existing type is <type>., new type is <type>.
Precision mismatch for Float, Ratio, and Unit values	User notification	Object <type> already exists in database, but has different type of Instance Based Attribute <type>: existing type is <type>, new type is <type>.
Existing IBAHolder has fewer IBAs	User notification	The existing Object <type> in database, but it does not have the value for Instance Based Attribute <type>. The value is <type>.
Existing IBAHolder has more IBAs	User notification	The existing Object <type> in database, but it has an extra value for Instance Based Attribute <type>. The value is <type>.
Type Identifier mismatch	User notification	Object existed with a different type. Existed type: <type>; expected type: <type>.
View Definition does not exist	User notification	"Definition of View <type> cannot be found. See the <i>Windchill Administrator's Guide</i> for further information."
Life Cycle Definition does not exist	User notification	"Definition of Life Cycle <type> cannot be found. See the <i>Windchill Administrator's Guide</i> for further information."
Life Cycle State does not exist in template	User notification	"Life Cycle State <type> cannot be found in the Life Cycle Template <type>. See the <i>Windchill Administrator's Guide</i> for further information."

Potential conflict	Behavior	Resolution or Message
Template containing Life Cycle State cannot be found	User notification	"Life Cycle State <type> cannot be found because the Life Cycle Template <type>, to which the State belongs, does not exist. See the Windchill Administrator's Guide for further information."
Domain containing team does not exist	User notification	"Team <type> cannot be found because the Administrative Domain <type>, where team resides, does not exist."
Team does not exist	User notification	"Team <type> cannot be found in the Administrative Domain <type>. See the Windchill Administrator's Guide for further information."
Location (Cabinet) does not exist	Create missing Cabinet if override conflicts = true, user notification otherwise	<p>"Cabinet &lt;type&gt; cannot be found. See the <i>Windchill Administrator's Guide</i> for further information."</p> <p><b>Note:</b> In PDMLink, there is only one Default cabinet and the items are not visible in the user interface if they are created in a cabinet other than Default. For import into PDMLink, the conflict should be ignored and the cabinet replaced by Default</p>
Location (Folder) does not exist	Create folder or user notification, (as checked)	"Folder <type> cannot be found. See the <i>Windchill Administrator's Guide</i> for further information."

## Administrative Conflicts of Administrative Objects

This section discusses conflicts for some common administrative items such as IBA Definition, Attribute Organizer, Quantity of Measure, Measurement System, and Soft Type Definition.

Potential conflict	Overridable	Behavior	Resolution or Message
Description mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	yes	User notification	Attributes are different for "Description". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	yes	User notification	Attributes are different for "Display Name". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Hierarchy Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	yes	User notification	Attributes are different for "Hierarchy Display Name". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Attribute Organizer does not exist	yes	User notification	Attribute Organization <type> cannot be found. See the Windchill Business Administrator's Guide for further information.
Creating Hierarchical IBA Definitions not allowed	no	User notification	Hierarchical Definition <type> is not allowed.
IBA Definition mismatch	no	User notification	Attribute definition <type> is defined locally as <type>, but is imported as <type>. See the Windchill System Administrator's Guide for further information.
IBA Definition or Attribute Organizer does not exist	yes	User notification	Definition of Instance Based Attribute <type> cannot be found. See the Windchill System Administrator's Guide for further information.

Potential conflict	Overridable	Behavior	Resolution or Message
Quantity of Measure does not exist	yes	User notification	Quantity of measure <type> does not exist.
Unit Definition exists with different base unit	no	User notification	The Quantity of measure <type> for Unit Definition <type> exists with different display units or overridden display units.
Measurement System mismatch for base symbol values	no	User notification	Measurement system <type> exists with different base symbol value.
Measurement System does not exist	yes	User notification	Measurement system <type> does not exist.
Type Definition does not exist	yes	User notification	Type Definition cannot be found: <type>
Attribute UserAttributable mismatch	no	User notification	Incompatible attribute "userAttributable" for: <type>, expected: <type>, found: <type>
Attribute Instantiable mismatch	no	User notification	Incompatible attribute "instantiable" for: <type>, expected: <type>, found: <type>
Attribute Deleted mismatch	no	User notification	Incompatible attribute "deleted" for: <type>, expected: <type>, found: <type>
Icon mismatch for Soft Type	yes	User notification	The icon <type> already exists. Overriding this conflict will rename the icon to a different name.
Existing Soft Type has fewer IBAs than in XML file	no	User notification	IBA value (attribute type: <type>, path: <type>, value: <type>) is expected with respect to import for: <type>
Existing Soft Type has extra IBA relative to XML file	no	User notification	Extra IBA value (attribute type: <type>, path: <type>, value: <type>) is found with respect to import for: <type>
Existing Soft Type has fewer Constraints than in XML file	no	User notification	Type constraint (enforcementRuleClassname: <type>, bindingRuleClassName: <type>, enforcementRuleData: <type>, IBA definition path: : <type>) is expected with respect to import for: <type>

Potential conflict	Overridable	Behavior	Resolution or Message
Existing Soft Type has extra Constraints relative to XML file	no	User notification	Extra type constraint (enforcementRuleClassname: <type>, bindingRuleClassName: <type>, enforcementRuleData: <type>, IBA definition path: <type>) is found with respect to import for: <type>
Existing Soft Type belongs to a different Logical Identifier Group relative to XML	no	User notification	The identified group of logical identifier is different: existing: <type>; expected: <type>.
Existing Soft Type has different Logical Identifier relative to XML file	no	User notification	The logical identifier is different: existing: <type>; expected: <type>.

## Dependency Conflicts

Potential conflict	Behavior	Resolution or Message
Referenced Document does not exist	No notification through conflict notification, but notification through general Windchill error message.	"Referenced document not found."
DescribedBy Document does not exist	No notification through conflict notification, but notification through general Windchill error message.	"DescribedBy document not found."
Used Part (Part Structure) does not exist	No notification through conflict notification, but notification through general Windchill error message.	"Part used in part structure not found."

## Metadata Conflicts

### Name and Number Conflict

Potential conflict	Behavior	Resolution or Message
Imported Number matches while imported Name does not match	User notification. Conflict is overridable.	"Warning: Name mismatch for part number <part number>"

### 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 item in the XML file doesn't exist in the database, in which case import fails -- a non-overridable conflict.
- The life cycle of the item 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 item is the life cycle of the previous iteration in the database.
  - Import as new version -- The life cycle of the newly created item is the life cycle from the XML file
  - Import as checked out -- The life cycle of the newly created item is the life cycle of the previous iteration in the database.
  - Modify non-versioned attributes -- The life cycle of the newly created item is the life cycle from the XML file.
  - Update checked out object in place-- The life cycle of the newly created item 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 item in the XML file does not exist in the database, in which case the team of the newly created item is the team of the previous iteration in the database.
- The team of the item 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 item is the team from the XML file.
  - Import as new version -- The team of the newly created item is the team from the XML file.
  - Import as checked out -- The team of the newly created item 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 item that is being checked out.
  - Modify non-versioned attributes -- The team of the newly created item is the team from the XML file.
  - Update checked out object in place-- The team of the newly created item is the team of the checked out object in the database. This behavior is chosen because the team package does not provide an API method to reassign TeamTemplate for an item 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 does not 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 item is the domain of the existing object in the database.

## Folder

There are 2 cases in which a folder conflict occurs:

- The folder of the item in the XML file does not exist in the database, in which case the new folder is created.

- The folder of the item 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 new version -- The folder of the newly created item is the folder from the XML file.
  - All other actions -- The folder of the newly created item is the folder of the existing object in the database.

## Context

If there is no context mapping file, the item will be imported to the context from where the import process is launched.

If there is context mapping file, the item will be imported according to the mapping file. If the mapping file maps the item to a context that does not 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 item 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 item with custom modeled attributes is exported from system A and then imported into system B where the item 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 ensure that import will be successful in a such a case.

Using such a rule 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, standardX05.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 version="1.0" encoding="UTF-8"?>
<userSettings>
  <mappingRules>
    <IGNORE>
      <tag> mySubTypeAttr1</tag>
      <value>*</value>
    </IGNORE>
    <CHANGE_TAG>
      <tag>SubTypeOfWTPart</tag>
      <newTag>WTPart</newTag>
      <newDtd>standardX05.dtd</newDtd>
    </CHANGE_TAG>
  </mappingRules>
</userSettings>
```

```
</userSettings>
```

## XSL Example

```
<?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>

            <tag> mySubTypeAttr1</tag>

            <value>*</value>

          </IGNORE>

        </mappingRules>

        <mappingRules>

          <CHANGE_TAG>

            <tag>SubTypeOfWTPart</tag>

            <newTag>WTPart</newTag>

            <newDtd>standardX05.dtd</newDtd>

          </CHANGE_TAG>

        </mappingRules>

      </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:

```
<IGNORE>

  <tag>tagName</tag>

  <path>pathOfTheTag</path>

  <value>tagValue</value>

</IGNORE>
```

In the preceding syntax, the following line is optional:

```
<path>pathOfTheTag</path>
```

In the preceding syntax, you can use the wild card \* in the following line:

```
<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 item 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 items such as IBA Definition, Attribute Organizer, Quantity of Measure, Measurement System, and Soft Type Definition.

Potential conflict	Behavior	Resolution or Message
Description mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	User notification	Attributes are different for "Description". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	User notification	Attributes are different for "Display Name". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Hierarchy Display Name mismatch for IBA Definition, Attribute Organizer, and Soft Type Definition	User notification	Attributes are different for "Hierarchy Display Name". The value of attribute <type> is different from the value as found in database. Expected: <type>, found: <type>
Attribute Organizer does not exist	User notification	Attribute Organization <type> cannot be found. See the Windchill Business Administrator's Guide for further information.
Creating Hierarchical IBA Definitions not allowed	User notification	Hierarchical Definition <type> is not allowed.
IBA Definition mismatch	User notification	Attribute definition <type> is defined locally as <type>, but is imported as <type>. See the Windchill System Administrator's Guide for further information.
IBA Definition or Attribute Organizer does not exist	User notification	Definition of Instance Based Attribute <type> cannot be found. See the Windchill System Administrator's Guide for further information.
Quantity of Measure does not exist	User notification	Quantity of measure <type> does not exist.
Unit Definition exists with different base unit	User notification	The Quantity of measure <type> for Unit Definition <type> exists with different display units or overridden display units.
Measurement System mismatch for base symbol values	User notification	Measurement system <type> exists with different base symbol value.

Measurement System does not exist	User notification	Measurement system <type> does not exist.
Type Definition does not exist	User notification	Type Definition cannot be found: <type>
Attribute UserAttributable mismatch	User notification	Incompatible attribute "userAttributable" for: <type>, expected: <type>, found: <type>
Attribute Instantiable mismatch	User notification	Incompatible attribute "instantiable" for: <type>, expected: <type>, found: <type>
Attribute Deleted mismatch	User notification	Incompatible attribute "deleted" for: <type>, expected: <type>, found: <type>
Icon mismatch for Soft Type	User notification	The icon <type> already exists. Override this conflict will rename the icon to a different name.
Existing Soft Type has fewer IBA than in XML file	User notification	IBA value (attribute type: <type>, path: <type>, value: <type>) is expected with respect to import for: <type>
Existing Soft Type has extra IBA relative to XML file	User notification	Extra IBA value (attribute type: <type>, path: <type>, value: <type>) is found with respect to import for: <type>
Existing Soft Type has fewer Constraints than in XML file	User notification	Type constraint (enforcementRuleClassname: <type>, bindingRuleClassName: <type>, enforcementRuleData: <type>, IBA definition path: <type>) is expected with respect to import for: <type>
Existing Soft Type has extra Constraints relative to XML file	User notification	Extra type constraint (enforcementRuleClassname: <type>, bindingRuleClassName: <type>, enforcementRuleData: <type>, IBA definition path: <type>) is found with respect to import for: <type>



# D

## Customizing Online Help

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
WebHelp Overview .....	D-2
Customizing Topic Content.....	D-3
Customizing Navigation Pane Content .....	D-6
Customizing Topic Appearance .....	D-10

## 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 a 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>/codebase/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:

Customization	DHTML
Edit, add, and delete topic content.	Yes
Edit existing table of contents entries/links and search results/links.	Yes
Delete table of contents entries/links and search results/links.	Yes
Add table of contents entries and links.	Yes
Add search results and links.	No
Change background color of topics and make other basic topic style changes <sup>1</sup> .	Yes
Add navigation tools in the left frame.	No
Add graphics and text and modify color at the top of the navigation pane (above the tabs).	No
Change icons on Contents tab.	No



Customization	DHTML
Change background color and color of links on navigation pane.	No

1. For important information about linked style sheets, see [Customizing Topic Appearance](#).

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:

```
<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>
<style type="text/css">
<!--
img_whs1 {border: none; width: 23px; height: 16px; float: none; border-style: none;}
-->
</style>
<script type="text/javascript" language="JavaScript" title="WebHelpInlineScript">
<!--
function reDo() {
    if (innerWidth != origWidth || innerHeight != origHeight)
        location.reload();
}
if ((parseInt(navigator.appVersion) == 4) && (navigator.appName == "Netscape")) {
    origWidth = innerWidth;
    origHeight = innerHeight;
    onresize = reDo;
}
onerror = null;
//-->
</script>
<style type="text/css">
<!--
div.WebHelpPopupMenu {position:absolute; left:0px; top:0px; z-index:4;
visibility:hidden;}
-->
</style>
<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>

```

## 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:

```

<script type="text/javascript" language="javascript1.2">
<!--
if (window.gbWhTopic)
{
    if (window.addTocInfo)
    {
        addTocInfo("TOC Heading\nTOC sub-Heading\nNew Topic Title");
        addButton("show",BTN_TEXT,"Show","","","","",0,0,"","","");
    }
    if (window.writeBtnStyle)
        writeBtnStyle();

    if (window.writeIntopicBar)
        writeIntopicBar(1);

    if (window.setRelStartPage)
    {

```

```

        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:

```
<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:

```
<tocdata>
<item name="NAME" url="URL" />
<book name="NAME" >
  <item name="NAME" url="URL" />
  <item name="NAME" url="URL" />
</book>
<book name="NAME" >
  <item name="NAME" url="URL" />
```

```

        <item name="NAME" url="URL" />
    </book>
</book>
</tocdata>

```

**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.

## 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:

```

<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" />
    <item name="Current Product" url="AdminProdToTemplate.html" />
</book>
</tocdata>

```

### 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:

```

<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" />
    <item name="Current Product" url="AdminProdToTemplate.html" />
</book>
</tocdata>

```

### 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.

## Adding an Entry or Book

To add a TOC entry or book, insert new entry in an appropriate whxdata/whtdata##.xml file. For example:

```
<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>
  <item 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:

```
<fts>
<chunkinfo url="whfwddata0.xml" first="200" last="made"/>
<chunkinfo url="whfwddata0.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/whfwddata##.xml (from the <chunkinfo> elements above). See the example below (this is a portion of the whfwddata0.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="accord"> 12, </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 `<tchunkinfo>` elements above. The numbers between the `<key>` opening and closing tags should be organized by relevance, with the 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/whfwddata##.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/whftdata##.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 `<tchunkinfo>` element with the same url name as the `whxdata/whftdata##.xml` file where the `<topic>` reference is present. Note the start value. The topic number is the start value of the correct `<tchunkinfo>` element plus the count number you noted above.

In the `whftdata0.xml` file shown above, the "Creating a Library" `<topic>` element is topic number three (topic number three counting from the top, with `<tchunkinfo>` 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.

Style	Description
P.Topic-Text-Bulleted	Used for bulleted lists
P.Topic-Text-Subbulleted	Used for second level bulleted lists
P.Topic-Text-Numbered	Used for numbered lists
P.Topic-Text-SubNumbered	Used for second level numbered lists
P.Table-Heading	Used for table heading rows
P.Table-Text	Used for text in the body of a table
P.Syntax	Used for monospace text (for example, code)
P.Syntax-indent	Used for indented monospace text

**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.)



# E


## Customizing Online Tutorials

This appendix explains how to download the tutorials for hosting on a Windchill server, add tutorial topics to the existing set of tutorials in Windchill PDMLink or Windchill ProjectLink, and to edit the tutorial index.

<b>Topic</b>	<b>Page</b>
Online Tutorial Infrastructure .....	E-2
Downloading the Tutorials .....	E-2
Adding a New Tutorial.....	E-3
Editing Existing Tutorials .....	E-8
Editing the Tutorial Index .....	E-9

## Online Tutorial Infrastructure

The basic tutorial infrastructure includes the following components:

- The tutorials themselves -- Currently, these can be HTML files strung together to form a tutorial.
- \*Index\_<locale>.xml files --These files contain the information about the tutorials and where they should be shown in the index. Each file defines a hierarchy and where the actual tutorials being referenced exist. These files are located in codebase/registry/tutorials.
- TutorialBuilder -- A Java class provided by PTC that you can use to regenerate the tutorial index structure (menuObjects.js) when you add or modify existing tutorial entries. This class is located in codebase/com/ptc/windchill/enterprise/tutorials/server/.
- menuObjects.js file -- This file contains the data used for the tutorial index that is displayed when a user clicks the Learn link  in the header. This file is generated by the TutorialBuilder. This file is located in codebase/wt/helpfiles/help\_<locale>/tutorials/.
- index.html -This page displays the tutorials index using menuObjects.js. This file is located in codebase/wt/helpfiles/help\_<locale>/tutorials/.

**Note:** Modifying any files that are provided by PTC is considered a site modification. To ensure that you do not lose your changes during a maintenance update, follow the recommended maintenance best practices as described in the *Windchill Customizer's Guide*. An introduction to these best practices can be found in this guide; see [Windchill Software Maintenance and Best Practices](#).

## Downloading the Tutorials

Use the following steps to download the tutorials from ptc.com and host them on a Windchill server:

1. Download the appropriate tutorialHandoff\_rt\_<locale>.zip language zip from the following:

`http://www.ptc.com/community/windchill/8/tutorials/`

For example, get the tutorialHandoff\_rt\_ja.zip for Japanese. See Creating Tutorials for a table of all language extensions.

2. Extract the files into your installation directory <Windchill>. Make sure to check "use folder names" when extracting.
3. Set the following property using the xconfmanager:

```
com.ptc.windchill.enterprise.tutorials.home=  
http://<ServerName>/<webAppAlias>/
```

For example, open a command prompt at *<Windchill>/bin/* and run the following command:

```
xconfmanager -s com.ptc.windchill.enterprise.tutorials.home=  
http://someServer.com/Windchill/ -t codebase/wt.properties  
-p
```

4. Optionally, the Apache Web Server can be configured to handle "page not found" errors by redirecting the user to a friendly error message. Add the following entries to *<Apache>/conf/httpd.conf* to configure Apache to handle any missing tutorial topics by redirecting to *Tutorial\_Navigation\_Message.html*:

```
<Directory "<Windchill>/codebase/wt/helpfiles/">  
    ErrorDocument 404 http://<ServerName>/<web_app_name>/  
wt/helpfiles/help_en/tutorials/Tutorial_Navigation_Message.html  
</Directory>  
<Directory "<Windchill>/codebase/wt/helpfiles/help_<locale>">  
    ErrorDocument 404 http://<ServerName>/<webAppAlias>/wt/helpfiles/  
help_<locale>/tutorials/Tutorial_Navigation_Message.html  
</Directory>
```

For example:



```
<Directory "D:/ptc/windchill.8.0/">  
    ErrorDocument 404 http://MyServer.com/Windchill/wt/helpfiles/help_en/tutorials/  
Tutorial_Navigation_Message.html  
</Directory>  
<Directory "D:/ptc/windchill.8.0/">  
    ErrorDocument 404 http://MyServer.com/Windchill/wt/helpfiles/help_ja/tutorials/  
Tutorial_Navigation_Message.html  
</Directory>
```

More information on the `ErrorDocument` directive can be found on the Apache Web site:

<http://httpd.apache.org/docs-2.0/mod/core.html#errordocument>

## Adding a New Tutorial

Use the following steps to add a new tutorial to the set of existing tutorials:

1. Create the tutorial as described in the following section.
2. Add the tutorial to the tutorial index that appears when a user clicks the Learn link, , in the header. This is done by editing the appropriate *\*Index\_<locale>.xml* file and executing the `TutorialBuilder` Java class to regenerate `menuObjects.js` so that the new tutorial is included as described in [Regenerating the Index Using the TutorialBuilder Java Class](#).
3. Optionally, enable a link in the Windchill solution for the Learn icon  that goes to the new tutorial as described in [Linking to a New Tutorial through the User Interface of a Windchill Solution](#).

## Creating Tutorials

You can use any tool that generates HTML files to create your new tutorial. There must be one HTML file that can be referenced in the menu structure to start the tutorial. New tutorial files should be located under the tutorials directory corresponding to the locale and Windchill solution for which they were created.

For example, create the following directory:

```
<Windchill>/codebase/wt/helpfiles/help_<locale>/tutorials/site/
```

In the above file paths, *<Windchill>* is the installation directory for the solution and *<locale>* is the locale extension for the language in which the tutorial is written. The following table lists the locale extensions:

Locale	Extension Value
Simplified Chinese	_zh_CN
Traditional Chinese	_zh_TW
French	_fr
German	_de
Italian	_it
Japanese	_ja
Korean	_ko
Spanish	_es

## Adding a New Tutorial to the Tutorial Menu Presented in a Windchill Solution

The \*Index\_<locale>.xml files define the tutorials index. They are located in the following directory:

```
<Windchill>/codebase/registry/tutorials/*
```

The files in this directory include the interopIndex\_<locale>.xml, pdmIndex\_<locale>.xml and pjlIndex\_<locale>.xml files that correspond to the "Windchill Operations", "Products and Product Data Management" and "Projects and Project Collaboration" sections respectively.

For customization, it is recommended that the files in <Windchill>/codebase/registry/tutorials/ remain unchanged and customized copies of those files be put into a separate directory. Set the following property in wt.properties using the xconfmanager as follows:

```
xconfmanager -s com.ptc.windchill.enterprise.tutorials.index=  
"index_location" -t codebase/wt.properties -p
```

For example:

```
xconfmanager -s com.ptc.windchill.enterprise.tutorials.index=  
"/registry/customTutorials" -t codebase/wt.properties -p
```

The TutorialBuilder looks for the index files in `<Windchill>/codebase<index location>`.

To add a new tutorial to a section of the index, edit the appropriate xml file in this new custom directory. For example, to add a new English "Products and Product Data Management" tutorial, edit the

`<Windchill>/codebase/registry/customTutorials/pdmIndex_en.xml` file.

After you edit any `*Index_<locale>.xml` files, the index must be regenerated using TutorialBuilder (see [Regenerating the Index Using the TutorialBuilder Java Class](#) section).

### `*Index_<locale>.xml` Content

The existing `*Index_<locale>.xml` files use the following tags to define the menu structure for a Windchill solution:

Tag Name	Description
MainTopic	This is a topic that is typically a container for other tutorials and usually does not have any specific tutorial specified for it.
SubTopic	This is a topic that is a child of a main topic and it can have actual tutorials defined for it, or it can have children topics defined for it. Typically, it does not have both.
FinalTopic	This is the last level and has to have a tutorial defined for it. It cannot have children topics associated with it.
InteropMainTopic	This is used, one time per menuObjects.xml file, to define what the very top level topic should be when both systems are installed.

The Label tag is used to define the link label displayed in the index.

The Resource tag is used to define the link location and the file containing the tutorial.

The tutorialIndex.dtd specifies the basics for the structure of the document.

## Label and Resource Tag Usage

File paths in the `<Resource>` tags are relative to the `<Windchill>/codebase/wt/helpfiles/help_<locale>/tutorials/` directory. For example, a `<Resource>` of the following:

```
interop/R7INOV/index.htm
```

Becomes:

```
<Windchill>/codebase/wt/helpfiles/help_<locale>/tutorials/
interop/R7INOV/index.htm.
```

In the existing `*Index_<locale>.xml` files, the `<?loc-begin key="value"?>` and `<?loc-end?>` tags are used for PTC localization purposes within the Label tags. These tags are not needed when you add new labels or resources for new tutorials in the Label and Resource tags. For example, the following tags in `interopIndex_en.xml` provide the **Windchill Overview** label in the English Index and the

`<Windchill>/codebase/wt/helpfiles/help_<locale>/tutorials/interop/R7INOV/index.htm` file path as the resource for the **Windchill Overview**:

```
<Label>Windchill Overview</Label>
<Resource>interop/R7INOV/index.htm</Resource>
```

## Example \*Index\_<locale>.xml File Additions

The following example additions to the Windchill ProjectLink `pjlIndex_en.xml` file (noted in bold) set up index entries for two new tutorials that describe how to create company parts in Windchill ProjectLink:

```
<!DOCTYPE Tutorials SYSTEM "pjlTutorialDtd.dtd">
<Tutorials>
  <InteropMainTopic>
    <Label>
      <?loc-begin key="overview.main"?>Projects and Project
Collaboration<?loc-end?>
    </Label>
  </InteropMainTopic>
  :
  <MainTopic>
    <Label>Creating a Company Part</Label>
    <SubTopic>
      <Label>Working with Part Basics</Label>
      <Resource> pdm/R7PDMPartBasics/index.html</Resource>
    </SubTopic>
    <SubTopic>
      <Label>Doing Advanced Part Activities</Label>
      <Resource> pdm/R7PDMPartAdv/index.html</Resource>
    </SubTopic>
  </MainTopic>
</Tutorials>
```



The example above results in an index similar to the following:

Projects and Project Collaboration	
<b>Overview</b> Product Introduction Layout and Navigation Creating a Project	<b>Creating a Company Part</b> Working with Part Basics Doing Advanced Part Activities
<b>Collaborating</b> Discussing Projects Discussing Documents Subscribing Routing Documents Approval Routing Notify Routing	<b>Managing CAD Models</b> Pro/ENGINEER Wildfire Other CAD Tools Creating Parts Managing a BOM
	<b>Microsoft Project Editing</b> Importing a Plan

## Regenerating the Index Using the TutorialBuilder Java Class

Use the TutorialBuilder class to regenerate the menuObjects.js file needed for the index.

From a windchill shell, enter the following command where *<Windchill>* is the installation directory for the solution:

```
windchill com.ptc.windchill.enterprise.tutorials.server.TutorialBuilder <Windchill>
```

Executing the TutorialBuilder class completes the following steps:

1. Loops through every help\_<locale> directory looking for the corresponding interopIndex\_<locale>.xml, pjIndex\_<locale>.xml and pdmIndex\_<locale>.xml files.
2. Processes the XML files, validating the existence of each <Resource>. If no file exists for a <Resource>, then the topic is not added to the index (the <Label> is skipped). For topics that are removed like this, warnings are output to the command line. To get more detailed messages from the TutorialBuilder set this property:

```
com.ptc.windchill.enterprise.tutorials.verbose=true
```

3. Appends topics to menuObjects.js. As TutorialBuilder completes the processing of each XML file, it adds the results to the end of the menuObjects.js file. The order of the Labels in the XML files determine the order that the labels appear in the index.
4. Outputs a menuObjects.js file for each locale. The menuObjects.js file is located in the <Windchill>/codebase/wt/helpfile/help\_<locale>/tutorials directory.

## Linking to a New Tutorial through the User Interface of a Windchill Solution


There are two methods for linking tutorials to user interfaces: one for template processor pages and one for JSP pages.

## Linking from a Template Processor Page

Windchill PDMLink tutorials are defined in a number of locations that are appropriate to the user interface they represent. For example, the following entry defines the tutorial for Create Part:

```
wt.services/rsc/default/wt.templateutil.processor.TutorialTemplate/CreatePartHelp/  
java.lang.Object/0=wt.helpfiles.help.tutorials.pdm.R7PDMPtBr.index
```


The tutorial entries follow the conventions of other services type property entries in Windchill PDMLink.

To enable the  icon in a specific user interface, the following script call can be used in an HTML file to generate the link:

```
<SCRIPT language=Windchill>useProcessorService method=getTutorialButton  
HelpContext=CreatePartHelp  
service=com.ptc.core.HTMLtemplateutil.server.processors.UtilProcessorService  
</SCRIPT>
```

## Linking from a JSP Page

Use the `pltutorial.properties` file that is located in the `<Windchill>\codebase`

directory. Within this file, each of the tutorials that correspond to a  icon in the user interface are defined.

For example, entry for the **Part Create** page is as follows:

```
part.part_create=wt/helpfiles/help_en/tutorials/pjl/R7CPCreatePrt/index.htm
```

The first part of the entry is the object, the second is the action.

In most common JSP pages, the code to enable the tutorial icon is already provided; however, for a standalone JSP, including the following JSP ensures its inclusion:

```
com\ptc\netmarkets\util\jsp\tutorial.jsp
```

## Editing Existing Tutorials

If you choose to edit existing tutorial files, be aware of the following items:

- Editing existing tutorial files is considered a site modification. To ensure that you do not lose your changes during a maintenance update, follow the recommended maintenance best practices as described in the *Windchill Customizer's Guide*. An introduction to these best practices can be found in this guide; see [Windchill Software Maintenance and Best Practices](#).
- If you download any updated tutorials from `ptc.com`, the download is also considered a site modification. Be sure to back up the downloaded files as described in the maintenance best practices in the *Windchill Customizer's Guide*.

- If you change the file name of the tutorial start file, you must also change the corresponding Resource tag in the \*Index\_<locale>.xml file.
- If the tutorial was linked from the Windchill solution, ensure that the link is still appropriate for the content changes you made to the tutorial.

## Editing the Tutorial Index

You can edit the content of the \*Index\_<locale>.xml files that make up the tutorial index. For example, you can do the following:

- Change the order that items appear in the index by changing the order in which the MainTopic and SubTopic tags are defined in the \*Index\_<locale>.xml file.
- Change the text that appears in the index by changing the text in the corresponding Label tag value.
- Modify the tutorial that displays as a result of clicking an index link by changing the corresponding Resource tag value.
- Change the index structure by adding or removing MainTopic and SubTopic tags.
- Remove any tutorial topics that are not relevant for your site.

For help with these tasks, see the information provided in the [Adding a New Tutorial to the Tutorial Menu Presented in a Windchill Solution](#) section of this appendix.



# Index

## A

- Adding
  - folders, 3-12
  - hosts, 3-11
- Architecture
  - three-tier, A-3
  - Web-based, A-2
- attribute mapping, 9-21
  - download behavior, 9-22
  - explicit, 9-21
  - implicit, 9-21
  - upload behavior, 9-22
- Attribute publishing in Pro/ENGINEER, 9-25
- Authentication, 1-48–1-49
  - troubleshooting, 1-49–1-50
  - user, A-6, B-3

## B

- Background queues. See Queues
- Backup
  - and replica sites, 4-23
  - process, 1-42
- bandwidth, 9-50
- 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

## C

- Caches
  - central cache vault, 3-4
  - local, 4-16
    - mirroring in, 4-18
    - moving data from, to master site, 4-19
    - vaults for, 4-18
- CAD document naming and numbering, 9-12
- CAD document soft type, 9-20
- Canceling revaulting schedule items, 3-22
- Central Cache Vault, 3-4

- Changing file location in external vaults, 3-24
- Codebase
  - administering, 2-8
  - server property, B-5
- codebase directory, 1-32
- Collections
  - bulk loading, A-23
- compression, 9-49
- config.pro options, 9-3
- Configuration properties, 1-2, B-7
- Configuring, 1-43
  - bootstrap client, 2-3
  - content handling, 7-3
  - firewalls, B-10
  - Pro/ENGINEER build rule, 9-25
  - Pro/ENGINEER initial collection, 9-29
  - queues, 8-9
  - Windchill environment, 1-43
- Conflict messages
  - administrative objects, C-19
  - common administrative objects, C-28
  - common business objects, C-17
  - dependency, C-21
  - metadata, C-22
- Content Cache Server, 4-15
- Content holders, 7-1
  - data formats, 7-4
  - overview, 7-2
- Content replication, 4-1–4-23, 9-48
  - compression, 4-14
  - importing certificates, 4-6
  - master sites, 4-5
  - overview, 4-2
  - overview of vaults and folders, 4-3
  - performance, 4-15
  - queues, 4-11
  - security, 4-6
  - summary, 3-2
  - Windchill Visualization Service, 4-13
- Context in import and export, 6-4
- Context mapping files, C-12
- Convera, A-22
- Converting multiple vaults to single vault, 3-24
- Creating

- external file vaults with FvLoader, 5-1, 5-2
- file vault policies, 3-15–3-18
- mounts, 3-13
- revaulting schedule items, 3-20
- user preferences, 1-38
- vault rules with FvLoader, 5-1, 5-2
- vaulting rules, 3-16
- Custom modeled attributes, reforming, C-24
- Customizing Pro/ENGINEER, 9-1
- Customizing the Naming Service, 9-16

## D

- data compression, 9-49
- database
  - backing up, 1-42
- Database components, A-18
- date format, 9-12
- Default document name preference, 1-38
- Default value, setting, 1-18
- Delegates
  - creating, 1-42
  - preference hierarchy, 1-41
  - scope, 1-37
- Deleting
  - folders, 3-13
  - revaulting schedule items, 3-22
  - vaulting rules, 3-18
- Dependency conflicts, C-21
- Desktop Integration, 1-32
- Downloading
  - Internet Explorer, 1-34
  - JAR files, 2-8

## E

- Editing
  - property files, 1-2
  - user preferences, 1-38
- Enabling As Stored Configurations, 9-27
- Enterprise search, 1-32
- epm.wconf file, 9-25
- Export, 6-1
  - Access control, 6-49
  - EPMDocuments, 6-36
  - EPMDocuments as Checked Out, 6-37
  - graphical user interface, 6-37
  - hierarchical IBA definitions, C-12
  - mapping rules, C-4
  - object from its properties page, 6-40
  - summary, 3-2
  - supported objects, 6-8

- XSL-based policy files, C-2
- Exporting an object from its properties page, 6-40
- external file vaulting, 9-48
- External Storage Administrator window, 3-10
- External Storage Scheduling window, 3-19

## F

- family symbols, 9-50
- File Size Limit for Content Replication, 4-13
- File Vault Administrator, 3-4, 3-10–3-15
- File vaults, 9-48
  - adding, 3-12
  - creating policies for, 3-15–3-18
  - creating with FvLoader, 5-1, 5-2
  - deleting, 3-12
  - deleting folder, 3-13
  - external, 3-1
    - overview, 3-2
  - external in replication, 4-2
  - maintaining, 3-14
  - properties for, 3-6
  - replica, 4-2
  - rules for, 3-15–3-16
    - creating, 3-16
    - deleting, 3-18
  - summary, 3-2
  - updating, 3-12
  - See also Revaulting
- Folders
  - adding, 3-12
  - deleting, 3-13
  - updating, 3-12
- Forcing Content to Vault, 3-22
- full-scale replica site, 4-4
- FvLoader, 5-1, 5-2

## H

- heap size, 9-48
- Hierarchical IBA definitions, C-12
- Hierarchy
  - preference, 1-40
  - scope, 1-41
- Hosts
  - adding, 3-11
  - choosing names, B-6
  - moving for server with external vaults, 3-25
  - updating, 3-11
- HTTP, 9-51
  - gateway, A-7
  - server, A-6

HTTPS, 9-51

## I

ie.properties file, 1-47

Import, 6-1

- Access control, 6-49

- Conflict messages, C-16

- EPMDocuments, 6-36

- EPMDocuments as Checked Out, 6-37

- graphical user interface, 6-42

- hierarchical IBA definitions, C-12

- mapping rules, C-4

- overridable conflicts, C-22

- properties, 6-48

- summary, 3-2

- supported objects, 6-8

- XSL-based policy files, C-2

Indexing, A-21

- file content, A-21

- policies, A-22

- publishing, A-22

INI files

- about, 9-9

- date format, 9-12

- overriding, 9-11

Instance Based Attribute definitions, C-12

internal organization, 1-29

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, 9-51

Java platform support, A-2

Java property files. See Property files

Java RMI servlet, B-10

Java Server Pages, 1-22

Java system property settings

- bootstrap client, 2-6

JSPs. See Java Server Pages

## L

latency, 9-50

lightweight replica site, 4-4

Load balancing, 1-45

Local Upload, 4-15

Log files, 1-9

- and replica sites, 4-23

- e-mail, 1-10

- maintaining, 1-43

- method server, 1-10

- queues, 8-12

- usage, A-18

- viewing, 1-7

## M

Managing Incomplete Dependents, 9-19

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

- local cache, 4-19

Maximum File Size for Content Replication, 4-13

Meeting Center, 1-23

menuObjects.xml files, E-2

metadata compression, 9-49

Metadata conflicts, C-22

METHOD Element, C-12

Method servers

- component, A-15

- configuring queues, 1-44

- configuring single, 1-44

- external vaults, 3-3

- home page, 1-22

- log file, 1-10

- log files, 1-46

- multiple

  - load balancing for, 1-45

Microsoft Office

- edit Windchill objects, 1-32

ModelCHECK validation, 9-19

Mounts

- creating, 3-13

- updating, 3-13

moveItemsBetweenReplicaVaults, 4-9

Moving server with external vaults to new host, 3-25

multiple servers, 9-48

## N

Null value, setting, 1-18

## O

- Object Relational Database Management System (OR-DBMS), A-18
- Oracle database
  - replication, 4-2
- Organization ID, 1-30
- Organization ID Type, 1-29
- Organizations
  - administering, 1-27
  - containers, 1-27
  - internal, 1-29
  - restricted, 1-28
- Overridable conflicts, C-22
- Overview
  - vaults and folders in content replication, 4-3

## P

- parameter mapping, 9-21
  - explicit, 9-21
  - implicit, 9-21
- Part naming and numbering, 9-12
- Performance Tuning, 9-48
- Policies for Import and Export, C-1
- Policy-Managed Naming and Numbering, 9-13
- Preferences
  - creating, 1-38
  - default document name, 1-37
  - default transition processes, 1-38
  - defining scopes, 1-39
  - delegates, 1-37, 1-41
  - Desktop Integration, 1-33
  - editing, 1-38
  - hierarchy, 1-39–1-40
  - hierarchy, defined, 1-39
  - macros, 1-40
  - managing, 1-37
  - removing, 1-39
  - scopes, 1-37
  - search types, 1-38
  - structure, defined, 1-39
  - thumbnail, 1-38
  - time zone, 1-38
- Pro/ENGINEER customizations, 9-1
- Pro/ENGINEER internal relationships, 9-45
- Property files
  - See also wt.properties file
  - editing, 1-2
  - listing values, 1-19
  - modifying, 1-7, 1-8
- Proxy servers

- client-side, B-11
- Meeting Center, 1-26
- server-side, B-11

## PTC

- documentation, xxxii
- technical support, xxxi

## Q

- Queue Manager, 8-2
- Queues
  - background, 8-1
  - configuring, 8-9
  - content replication, 4-11
  - DeliveryStatusOnStartup, 4-11
  - entry states, 8-2
  - groups, 8-9
  - log files, 8-12
  - maintaining, 8-14
  - managing, 1-7
  - maximum values, 4-11
  - out-of-the-box, 8-3
  - PurgeScheduleHistoryQueue, 4-11

## R

- Reference Link, 9-20
- Reforming custom modeled attributes, C-24
- Removing user preferences, 1-39
- Replica sites, 4-15
- Restricted organization, 1-28
- RetrievalWare
  - backup, 1-43
  - index loader, A-23
  - indexing, A-22
- Revaulting, 3-18–3-22
  - History window, 3-20
  - schedule items, 3-19, 3-20
- Revision, 9-31
- Revision level, autoassociate, 9-33
- revision level, setting, 9-31
- Revision of Associated Items, 9-31
- RMI, 9-51, B-7
- RMI-over-HTTP, B-8
- Runtime environment, A-1
- Runtime services chapter, 1-1

## S

- Scopes, preferences, 1-37
- Search types preference, 1-38
- secondary content, 9-44



- secret.text, 1-47
- secret.text2, 1-47
- Security
  - Infrastructures, B-1
  - JAR files, 2-9
- Server, 9-44
- Server codebase property, B-5
- Server hostname property, B-7
- Server Managers, 1-22
- site.xconf, 1-3, 1-11
- Soft Type, 9-20
- Soft Typing, 9-20
- Soft Typing CAD Documents, 9-29
- Special rules for mapping, C-4
- Starting Windchill, 1-7
- Stopping Windchill, 1-7
- Storage Administrator window, 3-10
- Supported objects for import export, 6-8
- System Administrator's Guide
  - about, xxix
  - audience, xxix
  - conventions, xxxii
  - organization, xxx
  - overview, xxix
  - related documentation, xxix
- System configuration
  - Pro/ENGINEER, 9-48
  - Windchill, 1-43
- System Configurator, 1-2, 1-7
- System properties. See Property files

## T

- Technical support, xxxi
- Threshold detection, 1-46
- Thumbnail preference, 1-38
- Time zone preference, 1-38
- Tutorials, E-2
- Type conflicts between Pro/ENGINEER parameters and Windchill IBAs, 9-23

## U

- Unrestricted organizations group, 1-29
- Updating
  - folders, 3-12
  - hosts, 3-11
  - mounts, 3-13
  - revaulting schedule items, 3-22
- URL generation, B-4
- Usage links, 9-26
- Usage Report utility, 1-34

- User preferences. See Preferences
- Users
  - authenticating, 1-48–1-49, A-6
  - preferences, 1-37
- Uses Link, 9-20

## V

- Validating XCONF files, 1-20
- Vault Configuration window, 3-10
- Vaulting Rule window, 3-17
- Vaulting rules. See File vaults, rules for vaults, 9-48
- Vaults. See File vaults
- versions, properties for new, 9-32
- Vetoing operations
  - Pro/ENGINEER, 9-26
- Viewing
  - revaulting results, 3-20
  - revaulting schedule items, 3-22
  - sites in tree view, 3-4

## W

- Web browsers, A-4
- WebEx, 1-23
- Windchill adapter, 1-47
- Windchill clients
  - components of, A-4
- Windchill environment
  - configuring, 1-43
- Windchill Foundation & PDM
  - supported objects for import/export, 6-29
- Windchill home page
  - Meeting Center, 1-23
  - overview, 1-22
- Windchill PDMLink
  - supported objects for import/export, 6-13
- Windchill ProjectLink
  - supported objects for import/export, 6-9
- Windchill Server page, 1-7
- Windchill servers
  - components of, 1-43, A-6
- Windchill Service Pack, 1-53
- windchill shell, 1-11
- Windchill Visualization Service and content replication, 4-13
- Windows XP clients, 1-34
- Workgroup Manager CAD Document Master, 9-20
- Workspace date format, 9-12
- wt.fv.replicationFileSizeThreshold, 4-13
- wt.org.InternalOrganization, 1-31

- wt.org.OrganizationOwned.displayOrganization, 1-31
- wt.prefs.delegates.DefaultSystemDelegate, 1-40
- wt.prefs.delegates.DelegateOrder, 1-41
- wt.prefs.delegates.UserDelegate, 1-40
- wt.prefs.WTPreferences class, 1-40
- wt.properties file
  - content replication, 4-7
  - file vaulting, 3-6
  - import, 6-48
  - log files, 1-43
  - modifying, 1-8
  - preference delegates hierarchy, 1-41
  - primary, 1-2
  - queues, 8-9
- wt.queue.queueGroup, 8-9, 8-11
- WTOrganization, 1-27
- wtSearch.xml file, 1-32

## **X**

- XCONF files
  - overview, 1-3
  - validating, 1-20
- xconf.dtd, 1-4
- xconfmanager, 1-11
  - command syntax, 1-12
  - help, 1-16
  - introduction, 1-2
  - using, 1-11
  - validating XCONF files, 1-20
- XSL transformation, C-11