



Windchill® System Administrator's Guide

Windchill 9.0

Pro/INTRALINK® 9.0™

Arbortext® Content Manager™

Windchill® PDMLink™

Windchill® ProjectLink™

September 2008

Copyright © 2007 Parametric Technology Corporation. All Rights Reserved.

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

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

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

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

For Important Copyright, Trademark, Patent, and Licensing Information: For Windchill products, select **About Windchill** at the bottom of the product page. For InterComm products, on the Help main page, click the link for Copyright 2007. For other products, select **Help > About** on the main menu for the product.

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. 02202007

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

Contents

Change Record	xiii
About This Guide.....	xxi
Intended Audience.....	xxi
Overview.....	xxi
Chapter Contents	xxii
Related Documentation	xxiii
Technical Support.....	xxiv
Documentation for PTC Products.....	xxiv
Comments	xxv
Runtime Services	1-1
Windchill Configuration Properties	1-2
The site.xconf File Format and Contents.....	1-4
The xconf.dtd File	1-4
Using Java Management Extensions (JMX).....	1-7
Client Types	1-8
Setting Up Your JMX Environment	1-12
Establishing Connections	1-13
Remote JMX Monitoring	1-14
Using JMX in a Cluster Environment	1-18
Accessing Windchill MBeans from JConsole.....	1-19
Using MBeans to Modify Property Values	1-20
Using MBeans to Start and Stop Your Windchill System	1-22
Using MBeans to Work with Log Files	1-23
Using MBeans to View and E-mail Log Files	1-24
Windchill MBeans	1-26
Using the xconfmanager Utility.....	1-34
xconfmanager Command Syntax	1-35
Viewing xconfmanager Help	1-40
Setting Property Values and Propagating Your Changes.....	1-40
Listing Property Information	1-43
Validating XCONF Files	1-44
Other xconfmanager Options.....	1-44

Setting Up Meetings.....	1-45
Accessing Meetings	1-45
Configuring E-mail Notifications for Your Site	1-46
WebEx Meeting Center Setup.....	1-46
WebEx User Name and E-mail Address	1-48
Proxy Server for Connection to WebEx Meeting Center	1-48
Disabling WebEx Meetings	1-49
Configuring Password Management.....	1-49
Enabling the Password Change User Interface	1-50
Setting Up a Password Policy	1-50
Deciding on a Password Expiration Notification Process.....	1-51
Customizing Password Change and Apache Login Help Topics	1-51
Setting Windchill Desktop Integration Preferences	1-52
Making Sure the Browser Is Set Correctly	1-52
Adding a Windchill Server to the Local Intranet Zone.	1-53
Setting Windchill Desktop Integration Preferences	1-53
Running the Windchill ProjectLink Usage Report Utility	1-53
Creating Usage Reports.....	1-53
Sample HTML Report.....	1-55
Ensuring Proper Backup and Recovery	1-55
Managing Logging	1-56
Out-of-the-box wt.property Log File Settings.....	1-57
Configuring Which Log Events Are Stored.....	1-59
Monitoring Temporary Files	1-60
Configuring Your Windchill Environment	1-60
Configuring a Single Method Server	1-61
Advanced Configurations	1-61
Changing Authentication Text between Servlet and Windchill Adapter	1-61
Administering the Authentication Process	1-62
Troubleshooting User Authentication	1-63
Double Authentication Can Be Eliminated When Using Internet Explorer	1-63
Authentication Configuration Tools	1-63
Windchill Scheduler	1-64
Windchill Scheduler Automatic Removal of History Items.....	1-64
Other Properties	1-65
Changing the Revision of Objects.....	1-65
Changing Views Assigned to Parts.....	1-66
Setting Up Windchill Index Search to Index CAD Parts.....	1-67
Best Practices for Windchill Administrative Tasks	1-68

Daily Administrative Tasks	1-68
Weekly Administrative Tasks	1-69
Monthly or Quarterly Administrative Tasks	1-71
Yearly Administrative Tasks	1-73
Windchill Maintenance Best Practices	1-73
Installation Process for the Windchill Service Pack	1-74
Best Practices for Managing Windchill Installation and Maintenance	1-76
About the windchill Command	1-77
About the windchill shell	1-80
Bootstrap Client and JAR Files	2-1
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
External File Vaults	3-1
About Storing and Moving Data in Windchill	3-2
About External File Vaults	3-2
Vaults with Automatic Folder Creation	3-4
Root Folders	3-5
Central Cache Vault	3-5
Windchill Properties for File Vaulting	3-7
Specifying the File Threshold Value	3-9
Creating a Notification Rule	3-10
Out-of-the-box Background Queues for External Vaulting	3-11
Windchill External Storage Administrator	3-12
About Mounts	3-14
Updating Root Folder Mounts	3-15
FolderToRootFolderConverter Tool	3-16
Maintaining Your Vaults	3-17
Creating Vaulting Rules	3-18
Examining Existing Rules	3-19

New Rules	3-20
Sending Content to a Vault	3-22
Managing Revaulting	3-22
Examining Existing Revaulting Schedule Objects.....	3-23
Viewing the Results of Revaulting.....	3-24
Forcing Content to Vault.....	3-25
Revaulting and the wt.fv.useVaultsForAllContent Property.....	3-27
Changing the Location of Files in External Vaults	3-28
Migration	3-29
Diagnostic and Repair Tools	3-29
Verifying Content Files	3-29
Detecting and Resolving Multiple Primary Content Files	3-32
Replication	4-1
About Replication	4-2
Windchill File Server.....	4-2
Data Security.....	4-5
Scheduling.....	4-6
Vaults and Folders	4-6
Two Types of Replication Sites	4-7
Modifying the Local Site	4-7
Replication Security	4-7
Importing Certificates into Sites.....	4-8
About Security and Using wt.intersvrcom.URLAuthenticator.useUrlCache	4-8
Editing Replication Properties	4-8
Master and Replica Properties	4-9
Basic Properties	4-11
Out-of-the-box Background Queues for Replication	4-12
Maximum Queue Values	4-13
Replication and Windchill Visualization Service.....	4-14
Configuring Properties.....	4-14
To Replicate WVS Viewables.....	4-14
Replication and Compression	4-14
Improving 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

Predictive Replication	4-23
Ad Hoc Replication	4-26
Site Proximity	4-26
User Initiated Replication	4-26
Parallel Replication	4-27
Deleting Registered File Servers	4-29
Updating File Servers	4-30
Upgrade Considerations	4-31
Enhancing the File Server User Experience	4-31
Utility to Assist Backups	4-33
FvLoader and ReplicaLoader	5-1
About FvLoader and ReplicaLoader	5-2
Tasks	5-2
Commands	5-3
Configuring External File Vaults or Rules	5-4
Supporting Replica Site Vaulting	5-5
Removing Rules	5-7
Listing Domains	5-7
The -listContainers Argument Obtains Data	5-8
The -listDomains Argument Presents Data	5-8
Listing Rules	5-9
Windchill Export and Import.....	6-1
About Export and Import.....	6-2
Context Considerations	6-4
Export Container Availability	6-4
Import Container Availability	6-6
Controlling the Destinations of Imported Objects with Context Mapping Files	6-7
Supported Objects List	6-8
Supported Windchill ProjectLink Objects	6-9
Supported Windchill PDMLink Objects	6-13
Configuration Specification Settings.....	6-26
EPMDocuments.....	6-27
Attribute Limitations	6-27
EPMDocuments as Checked Out	6-28
Export User Interface.....	6-28
Exporting from the Export Window	6-29
Exporting an Object from Its Properties Page	6-32
Import User Interface.....	6-34
Additional Export and Import Actions	6-38

Additional Lock Export Action.....	6-38
Additional Import Actions.....	6-38
Properties.....	6-39
Windchill Export Properties.....	6-40
Windchill Import Properties.....	6-41
Access Control.....	6-42
Export Access Control Rules.....	6-43
Import Access Control Rules.....	6-43
Content Holders and Content Objects.....	7-1
About Content Holders.....	7-2
Content Handling Configuration.....	7-3
Adding and Updating Data Formats.....	7-4
Background Queues.....	8-1
About Background Queues.....	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.....	8-5
MarkForDeleteQueue.....	8-5
NotificationQueue.....	8-5
NotificationScheduleQueue.....	8-6
PagingScheduleQueue.....	8-6
PartsLinkQueue.....	8-6
ProjectScheduleQueue.....	8-6
PublisherQueues.....	8-7
PurgeOrphanedEffAuditsQueue.....	8-7
StatisticsScheduleQueue.....	8-8
WfPropagationQueue.....	8-8
WfScheduleQueue.....	8-9
WfUserWorkQueue.....	8-9
Configuring Background Queues and Related Properties.....	8-10
Configuring a Method Server for Background Queues.....	8-10
Configuring Multiple Background Method Servers.....	8-10
Background Queue Properties.....	8-11
Background Queue Log Properties.....	8-13

Other Background Queue-Specific Properties	8-13
Regular Queue Maintenance.....	8-14
Maintaining Queues.....	8-14
Publishing Agent.....	9-1
Introduction to CAD Agent.....	9-2
Before You Begin	9-2
Guidelines for Configuring a Worker "Common File System"	9-3
Configuring the CAD Agent	9-4
CAD Worker on the Same Machine	9-4
Configuring a CAD Worker on a Different Windows Machine	9-9
Creating a Common Windows to Windows File System.....	9-10
Adding CAD Worker to a Different Windows Machine.....	9-10
Running Worker Daemon as a Windows Service	9-12
Running the Worker Daemon as a Program.....	9-13
CAD Worker on a Different UNIX Machine.....	9-14
Creating a Common File System	9-14
Adding a UNIX CAD Worker to the CAD Agent.....	9-15
Configuring a Non-English UNIX Worker to Run FTP Daemon	9-16
Defining Publish Queue and Worker Availability Settings	9-16
Testing the CAD Agent Configuration	9-17
Test 1: Starting the CAD Worker	9-18
Test 2: Create a Viewable on Demand.....	9-18
Test 3: Creating a Representation from Local Data	9-19
Using WVS with Multiple Background Servers.....	9-20
Configuring the CAD Agent in a Windchill Cluster.....	9-21
Generating Viewables and Thumbnails in Clustered Environments.....	9-21
ProductView Collaboration in a Clustered Environment	9-21
Making Thumbnails Viewable for Microsoft Word or Excel Files	9-23
File Synchronization	10-1
Before You Begin	10-2
Using a File Synchronization-Capable Worker to Download Files	10-2
Configuring the CAD Worker	10-4
Configuring the ProductView Adapter for HTTPS.....	10-6
Publishing Wildfire Assemblies as Positioning Assemblies.....	10-8
Arbortext Publishing Engine (APE) Worker and Publishing	11-1
Configuring the APE Worker	11-2
Before You Begin.....	11-2
Configuring the APE Worker.....	11-3

Configuring the APE Worker Monitor	11-5
Configuring the CAD Agent	11-7
Configuring Publishing for Arbortext Authored Dynamic Documents	11-7
Overview	11-7
Configuring the WVS Publisher for the Arbortext Authoring Application	11-8
Publishing Arbortext Authored Dynamic Documents	11-9
Windchill Runtime Environment.....	A-1
Web Infrastructure	A-1
Java Platform Support	A-1
Three-Tier Architecture	A-3
Client Software Components	A-3
Web Browser.....	A-3
HTML Pages	A-4
Java Applets.....	A-4
Server Software Components.....	A-5
HTTP Server	A-5
HTTP Gateway.....	A-6
Server Manager.....	A-11
Method Server.....	A-14
Log Files.....	A-18
Database Components	A-18
Object Relational Database Management System (ORDBMS)	A-18
Full Text Retrieval Indexing Components	A-20
File Content Indexing	A-20
Publishing.....	A-21
Indexable Objects.....	A-21
Indexing Policies	A-21
IndexDelegate Objects	A-22
Index Loader	A-22
Windchill Considerations for Security Infrastructures	B-1
About Security Infrastructures.....	B-1
Protocols	B-1
Authentication	B-3
URL Generation	B-4
Server Codebase Property.....	B-4
Relative and Absolute URLs	B-5
Choosing Host Names	B-5
RMI	B-6
Server hostname Property	B-6

Configuration Properties	B-7
Using HTTPS Protocol.....	B-7
Import and Export Policies, Mapping Rules, and Conflict Messages	C-1
XSL-based Policy Files.....	C-1
Policy File Examples.....	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-12
Java Mapping with the METHOD Element.....	C-12
Hierarchical Instance Based Attribute Definitions, Exporting, and Importing	C-13
When to Use Mapping Files for Hierarchical IBAs	C-13
How to Write a Mapping File for Hierarchical IBAs	C-14
Conflict Messages	C-17
Importing RatioDefinition and RatioValue	C-17
Administrative Conflicts of Common Business Objects	C-18
Administrative Conflicts of Administrative Objects.....	C-20
Dependency Conflicts.....	C-22
Metadata Conflicts	C-23
Reforming Custom Modeled Attributes.....	C-26
Example of Two Formats of Mapping Files	C-26
Ignoring an Attribute	C-28
Changing a Tag to a Different Name	C-28
Administrative Conflicts of Common Administrative Objects	C-29
Customizing Online Help	D-1
About WebHelp	D-2
Customization Summary.....	D-2
Customizing Topic Content	D-3
Adding a New Topic.....	D-3

Modifying an Existing Topic.....	D-6
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-11
Online Tutorials	E-1
Online Tutorial Infrastructure	E-1
Downloading the Tutorials	E-2
Adding a New Tutorial.....	E-3
Creating Tutorials.....	E-3
Adding a New Tutorial to the Tutorial Menu Presented in a Windchill Solution	E-4
Regenerating the Index Using the TutorialBuilder Java Class	E-6
Editing Existing Tutorials.....	E-7
Editing the Tutorial Index	E-7
Troubleshooting the Publishing Agent.....	F-1
Troubleshooting the CAD Agent	F-2
Running the CAD Agent in debug mode	F-2
Manually Starting Workers	F-3
Starting Workers from the CAD Agent	F-6
Using the CADAgent Monitor	F-7
Manually Publishing a CAD Document	F-13
Using the Publish Monitor	F-13

Index

Change Record

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

Table 1 Changes for 9.0 M050

Chapter	Description
Runtime Services	In the Changing the Revision of Objects section, added "java" to the command being executed. Added the Changing Views Assigned to Parts section.
Bootstrap Client and JAR Files	Removed swmaint.xml script reference.
External File Vaults	Added the Creating a Notification Rule section. Updated the Creating Vaulting Rules section.
Replication	Throughout, removed documentation describing the use of a full-scale replica site, which is no longer recommended functionality. Updated the Windchill File Server section, adding new sub-sections on installing directly from a maintenance release and updating manually. Updated the Ad Hoc Replication section. Updated the procedure in the Removing Unreferenced Files from a Cache Vault on a Replica Site section.
Windchill Export and Import	Updated the About Export and Import section to clarify that the Export/Import Manager is not available in Arbortext Content Manager.
Background Queues	Updated description of NotificationQueue .

Table 2 Changes for 9.0 M020

Chapter	Description
Runtime Services	<p>Reorganized and updated the JMX information in the following sections:</p> <ul style="list-style-type: none">• Using Java Management Extensions (JMX)• Windchill MBeans <p>Added the Changing the Revision of Objects section.</p> <p>Added the Best Practices for Windchill Administrative Tasks section.</p>
External File Vaults	<p>Added new section: Diagnostic and Repair Tools.</p>
Replication	<p>Updated the Replication Security section with new information about multiple use of a single URL with the following property:</p> <p>wt.intersvrcom.URLAuthenticator.useUrlCache</p>
Background Queues	<p>Updated the PurgeOrphanedEffAuditsQueue section.</p> <p>Updated the Indexing Queue and Bulk Indexing Queue section for Windchill Index Search.</p>
Windchill Runtime Environment	<p>Updated the Full Text Retrieval Indexing Components section for Windchill Index Search.</p>

Table 3 Changes for 9.0

Chapter	Description
Runtime Services	<p>Moved preferences and organization information to the <i>Windchill Business Administrator's Guide</i>.</p> <p>Moved some Windchill configuration information to the <i>Windchill Advanced Deployment Guide</i>.</p> <p>Added new sections:</p> <ul style="list-style-type: none">• Using Java Management Extensions (JMX)• Remote JMX Monitoring• Configuring Password Management
Background Queues	<p>Updated references for managing queues to the Queue Manager utility.</p> <p>Updated Configuring Background Queues and Related Properties section to include log4j logging information and moved the queue grouping information to the <i>Windchill Advanced Deployment Guide</i>.</p>
External File Vaults	<p>Updated mount information. Added new information about Automatic Folder Creation and root folders.</p> <p>Removed logging properties, which are replaced by log4j. Information about log4j is now located in the Runtime Services chapter.</p>

Chapter	Description
Replication	<p>Added the following new features:</p> <ul style="list-style-type: none"> • Windchill File Server, which includes information about the File Server Manager Utility for managing the installation with the PTC Solutions Installer (PSI) • Predictive Replication • Ad Hoc Replication • User Initiated Replication • Parallel Replication
Windchill Export and Import	Added new functionality regarding the export of top level objects in jar of jars.
FvLoader and ReplicaLoader	Added ReplicaLoader for the replica site. FvLoader is now only for the master site.
Publishing Agent	Updated the concept of "publishing" within Windchill. Formerly limited to configuring the CAD Agent, the chapter is now also pertinent to publishing non-CAD data, such as Dynamic Documents and WTDdocuments.
File Synchronization	Expanded the information on file synchronization to include all of the Windchill 9.0 file synchronization-capable workers: These workers include Pro/ENGINEER, Unigraphics, SolidWorks, Autodesk Inventor, Arbortext.
Windchill Considerations for Security Infrastructures	Updated the RMI section to move tunneling RMI over HTTP to the <i>Windchill Advanced Deployment Guide</i> .

Table 4 Changes for Windchill 8.0 M020

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

Table 5 Changes for Windchill 8.0 M010

Chapter	Description
Chapter 1, Runtime Services	Updated this section for 8.0 M010.

Table 6 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	Description
Chapter 1, Runtime Services	<p>In the Using the xconfmanager Utility section, documented additional xconfmanager parameters for adding and removing values from multi-valued properties and for validating XCONF files.</p> <p>Updated the Other properties related to Windchill Scheduler operations are the following section to reference the <i>Windchill Customizer's Guide</i> for specific information about making changes to Windchill code.</p>
Chapter 2, Bootstrap Client and JAR Files	Added Determining Client JAR Contents section.
Chapter 3, External File Vaults	Added warning for creating and mounting folders in the About Mounts section.
Chapter 4, Replication	<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 Removing Unreferenced Files from a Cache Vault on a Replica Site section.</p>
Chapter 5, FvLoader and ReplicaLoader	Added the VE,vaultName argument in the Configuring External File Vaults or Rules section.
Chapter 6, Windchill Export and Import	Revised the Controlling the Destinations of Imported Objects with Context Mapping Files section to improve readability and provide specific information on context mapping files.
Chapter 7, Content Holders and Content Objects	Moved information contained in the Adding and Updating Data Formats section to the <i>Windchill Customizer's Guide</i> .

Chapter	Description
Chapter 8, Background Queues	Added new queues: CtScheduleQueue, Indexing Queue, Bulk Indexing Queue, and PartsLink Queue to the Out-of-the-box Background Queues section.
Appendix E, Online Tutorials	Updated chapter to document new approach for tutorials.

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:

	System Administrator
Responsibilities	Keeping the system running. Interfacing with other systems. Administering Windchill applications.
Skills	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.

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.

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.

Chapter Contents

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

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

Chapter 1, [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, [Bootstrap Client and JAR Files](#), 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, [External File Vaults](#), describes the creation and maintenance of external file vaults.

Chapter 4, [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, [FvLoader and ReplicaLoader](#), describes the FvLoader and ReplicaLoader utilities. FvLoader, a shortened version of File Vault Object Loader, works with the master site; ReplicaLoader works with replica sites.

Chapter 6, [Windchill Export and Import](#), describes files and configuration properties for moving content and metadata to and from Windchill sites.

Chapter 7, [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, [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, [Publishing Agent](#), describes how to use the Publishing Agent - commonly referred to as the CAD Agent - to manage the relationship between one or more workers and Windchill Visualization Service.

Chapter 10, [File Synchronization](#), provides the instructions to configure file synchronization-capable workers to assist the Visualization server in publishing a viewable.

Chapter 11, [Arbortext Publishing Engine \(APE\) Worker and Publishing](#), provides instructions for configuring the Arbortext Publishing Engine (APE) Worker,

configuring the WVS publisher for the Arbortext Authoring Application, and defining and loading publishing rules.

Appendix A, [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.

Appendix E, [Online Tutorials](#), describes how to customize Windchill online tutorials.

Appendix F, [Troubleshooting the Publishing Agent](#), provides information to help analyze and resolve issues you may encounter when using the Visualization Services CAD Agent.

Related Documentation

The following documentation may also be helpful:

- *Windchill Business Administrator's Guide*
- *Windchill Installation and Configuration Guide - Advanced*
- *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*
- *Windchill Advanced Deployment Guide*
- *Oracle 10g Tuner for Windchill Solutions Installation and Configuration Guide*
- *FAST InStream Operations 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 or the product documentation.

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

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

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

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

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

Documentation for PTC Products

You can access PTC documentation using the following resources:

- **Windchill Help Page** — Click **Help** in the header of any Windchill page to open the **Windchill Help** page, which provides you with a portal to all Windchill documentation, including:
 - A complete set of current online help topics for your products
 - Product tutorials available on the PTC Web site
 - Windchill manuals for users, administrators, and programmers

In addition, you can click **Search All Help Sources** to access the Windchill Help Center, an online knowledgebase that includes a universal index of all Windchill documentation. You can search all of the documentation at once, or use the advanced search capability to customize your search. Once you have located a topic you want to reference later, you can bookmark that topic for quick access and even save your own comments about the topic.

- **Product CD** — All relevant PTC documentation is included on the CD set.
- **Reference Documents Web Site** — All books are available from the Reference Documents link of the PTC Web site at the following URL:

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

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

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

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

1

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 Java Management Extensions (JMX)	1-7
Windchill MBeans.....	1-26
Using the xconfmanager Utility	1-34
Setting Up Meetings.....	1-45
Configuring Password Management	1-49
Setting Windchill Desktop Integration Preferences	1-52
Running the Windchill ProjectLink Usage Report Utility	1-53
Ensuring Proper Backup and Recovery.....	1-55
Managing Logging	1-56
Monitoring Temporary Files	1-60
Configuring Your Windchill Environment	1-60
Changing Authentication Text between Servlet and Windchill Adapter	1-61
Administering the Authentication Process	1-62
Troubleshooting User Authentication	1-63
Windchill Scheduler	1-64
Changing the Revision of Objects.....	1-65
Changing Views Assigned to Parts	1-66
Setting Up Windchill Index Search to Index CAD Parts	1-67
Best Practices for Windchill Administrative Tasks	1-68
Windchill Maintenance Best Practices.....	1-73
About the windchill Command	1-77
About the windchill shell	1-80

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.

The `xconfmanager` is a command line utility that you can run to add, remove, or modify properties in any Windchill property file. By using the `xconfmanager` you modify the properties and then propagate the changes to the property values.

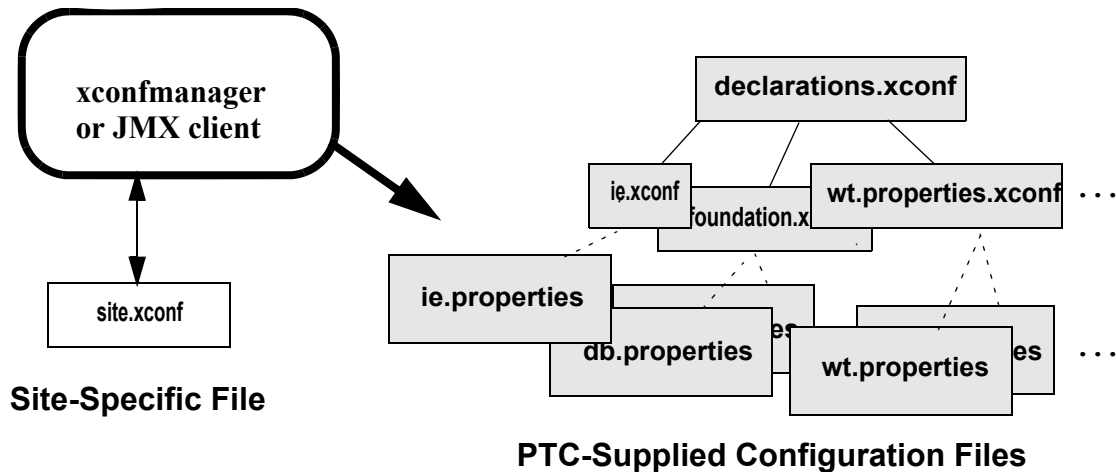
JMX provides MBeans which allow you to view and modify properties, change log settings, configure notifications, and so forth. The Properties MBean provides a read only view of all properties which can be modified. The PropertiesManager MBean allows you to view and change the values for properties. For additional information about using JMX to view Windchill properties, see [Using Java Management Extensions \(JMX\)](#).

The changes made through a JMX client and `xconfmanager` 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 the `xconfmanager`.

Note: By using the `xconfmanager` utility or a JMX client, your `site.xconf` file always contains 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 xconfmanager utility 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 a JMX client or the xconfmanager utility to update properties, Windchill always propagates the changes to the corresponding property files.

When using a JMX client, you must apply your changes using the applyPendingChanges operation available through the PropertiesManager MBean. For additional information about using JMX to view Windchill properties, see [Using Java Management Extensions \(JMX\)](#).

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 utility, see [Using the xconfmanager Utility](#).

Whenever you change a property setting using 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 backup 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 JMX client 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 adds 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 adds 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 a JMX client or the xconfmanager to modify the contents of the site.xconf file, some administrators may chose to modify the site.xconf file without using the Windchill tools. If you do manually modify the site.xconf file, be sure to format elements according to the xconf.dtd, which is documented in the next section. To propagate your changes to the affected property files, you must run the xconfmanager with the -p option and, to use the updated property files, you must restart your Windchill solution.

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 a JMX client, see [Using Java Management Extensions \(JMX\)](#).

The xconf.dtd File

Windchill uses the xconf.dtd to validate all elements in all XCONF files that it uses, including the site.xconf file. To ensure that this validation takes places for all XCONF files, no matter where they are located in the codebase and without

access to the internet, the xconf.dtd is supplied using a JAR file and is not readily available through the Windchill directory structure.

The contents of the DTD file is as follows:

```
<!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 UnsetProperty EMPTY -->
<!-- ATTLIST UnsetProperty
    %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 Java Management Extensions (JMX)

Java Management Extensions (JMX) provide a foundation for the management and monitoring of Java applications, such as Windchill, by using Java Management Beans (MBeans). Windchill has its own set of MBeans that are used to manage and monitor the application. Additionally, there are other standard Java MBeans that monitor more general areas of application performance, which are not specific to Windchill.

In JMX, a managed bean is a special type of Java bean used to encapsulate management or monitoring functionality. Each MBean represents a resource, which in terms of JMX, can represent an object, aspect, subsystem, or component of an application or system.

Each MBean consists of the following:

- Attributes -- Allow data to be exposed for read and write purposes.
- Operations -- Allow various actions to be performed against the MBean.
- Notifications -- Convey status change information.
- MBeanInfo -- Provides the detailed metadata on the available attributes, operations, and notifications for each MBean.

JMX allows the system administrator to monitor and manage aspects of the Windchill application that are only known by the Java Virtual Machine (JVM). For example, you can use JMX to do the following:

- Control how notifications are sent
- Edit properties
- E-mail and view log files
- Start and stop the server manager
- Manage and monitor critical areas such as:
 - Garbage collection
 - Memory usage
 - CPU usage
 - Threading

The following sections describe the JMX clients that you can use and provide some details on the MBeans you can use for monitoring and managing Windchill.

JMX also allows you to monitor Windchill performance to see which threads are alive or how much memory is currently being used. More information on performance management and tuning can be found in the *Windchill Performance Tuning Guide*.

Client Types

You can monitor the Windchill MBeans using any JMX client. The sections that follow provide information on some potential clients.

Many of the usage examples provided in this chapter are for JConsole, which is the standard client bundled with the Java Software Development Kit (SDK).

JConsole

JConsole is bundled with the Java Software Development Kit (SDK) starting with Java 5. JConsole uses a tree to display the MBeans, allowing you to easily navigate to the MBean of interest.

JConsole is a good JMX management console; however, it is an RMI-based Java application, which is not the best client for all use cases. Also, the Java 5 JConsole current functionality is not as adequate as other JMX consoles. For example, you cannot see a long description of an attribute in JConsole.

After JConsole is started, it provides a list of Java processes that are running on the same machine for the current user ID that is configured to allow local JMX connections.

Note: Out-of-the-box, Windchill only allows local JMX connections, essentially those belonging to the same user ID and user account.

To allow JConsole (and other JMX management consoles) to connect remotely, additional configuration is required. For information on remote management, see [Remote JMX Monitoring](#).

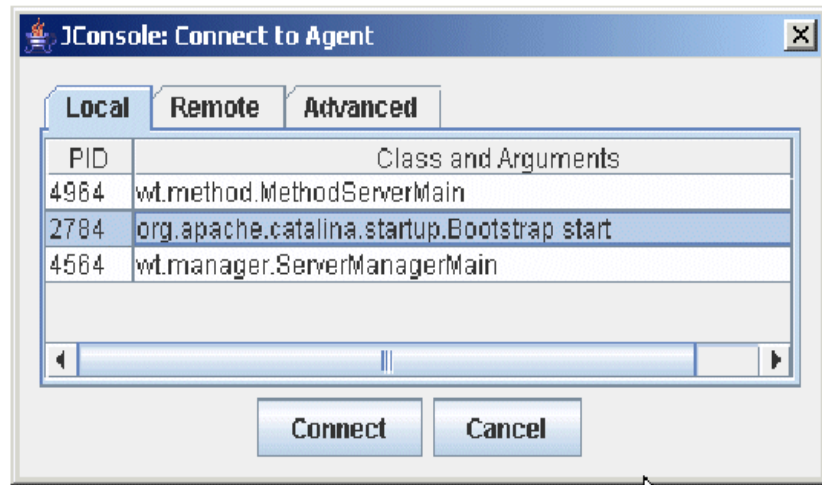
Getting Started with JConsole

For detailed instructions on using JConsole to manage and monitor applications, use the following URL:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html>

On a Windows system, you can access JConsole from the bin directory where Java 5 is installed. JConsole is started in the same manner as Tomcat or Apache; click the jconsole.exe file located in the bin directory.

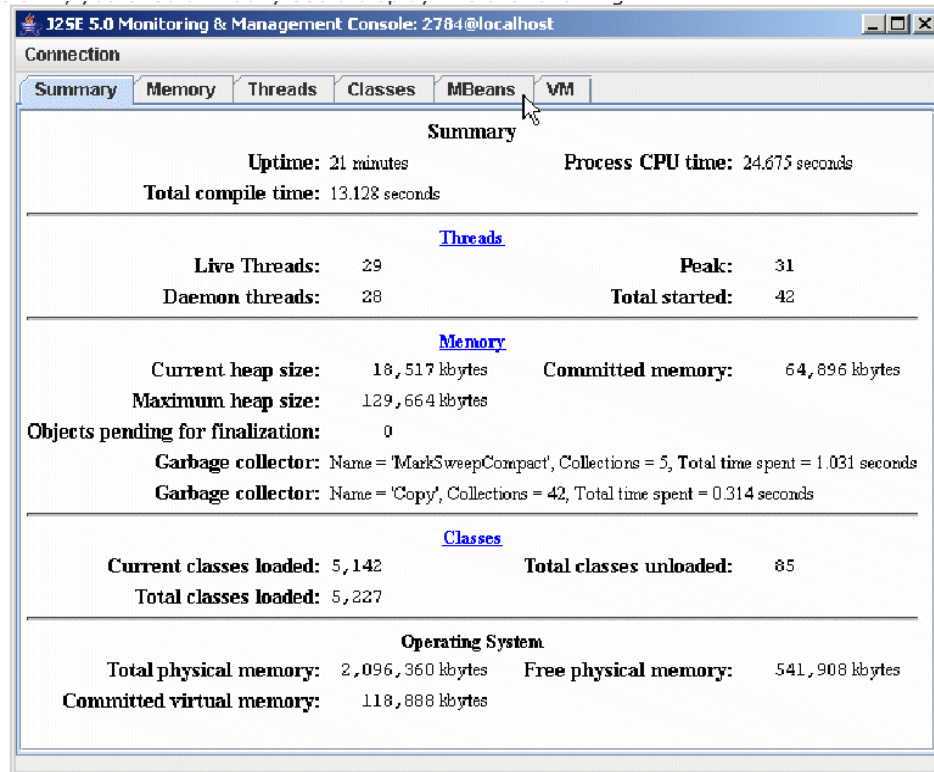
After JConsole is started, it provides a list of Java processes that are running locally under the current user ID and that are configured to allow JMX connections. The window presented to you is similar to the following:



Each line displayed in the table indicates the process ID and the main Java class that is being run. In this example, the server manager and method server are easily identifiable by their class names. The other process (starting with org.apache.catalina), which is shown selected is for Tomcat.

Select the process you want to manage and then click **Connect**.

A display similar to the following opens:



The main window provides summary information and includes details on:

- Memory -- provides details on how memory is being used.
- Threads -- lists all the threads in the JVM.
- Classes -- shows the number of classes unloaded and loaded.
- MBeans -- provides the interface from which you can do activities such as set logging options, properties, and e-mail options, and monitor JVM activities.
- VM -- gives all detail on Java parameters, operating system data, and so on. From this tab, you can verify that any parameters you specified through the xconfmanager utility were correctly applied and that the machine is not running out of memory.

After you have mastered the basic navigation techniques for JConsole, you can use the examples provided later in this chapter to learn about how to perform some basic Windchill-specific tasks from JConsole.

jManage

jManage provides an HTML client interface to JMX which allows you to bookmark the aspects you find important. Therefore, you have a way to return to specific pages at a later time. It also allows long descriptions to display for both attributes and MBeans. These descriptions provide further detail on what the individual attributes and MBeans represent respectively.

jManage can be downloaded online from the following URL:

<http://www.jmanage.org>

Install jManage using the instructions available on the download site.

With jManage, you are required to connect directly to individual MBeans by specifying their address.

Getting Started with jManage

After jManage is installed and running, you can configure it to monitor the Windchill server processes using the jManage HTML user interface by doing the following:

1. Select **Add New Application** > **jsr160 application**.
2. Fill in the following fields:
 - In the **Name** field, enter **Windchill**.
 - In the URL field, enter the following:

```
service:jmx:rmi:///fullserverhostname/jndi/rmi:///fullserverhostname:jmxPort/jmxrmi"
```

Replace *fullserverhostname* and *jmxPort* with appropriate values for the Windchill server manager.

- In the name and password fields, enter your name and password, if necessary.
3. Click **Save**.

Other JMX Clients

The underlying JMX communication technology relied upon by the Windchill JMX software is referred to as "JSR 160" or sometimes "Java 5 (JMX Remoting)". This is a standard part of the latest Java standards and so all JMX consoles should support this or will be moving to support it in the near future. Any JMX console that supports this technology can connect to Windchill server processes using the full JMX URL.

There are numerous JMX clients available for you to use. You can find out more information by doing a search for JMX clients using your preferred search engine.

Individual clients may differ in the way you perform the tasks, but offer the same functionality. For instance, JConsole allows you to navigate to the individual

MBeans via a tree, whereas jManage requires that you connect directly to individual MBeans by specifying their address. Review the documentation of the various available clients to determine which of them best suits your needs.

Setting Up Your JMX Environment

The basic JMX features include the ability to automatically send e-mail notifications for defined events. To use this feature, you must:

- Ensure that your mail server is setup to send the e-mail notifications. Check with your site IT department to verify the mail host is setup for the environment where Windchill is running. By default, Windchill sends all electronic mails anonymously to the SMTP host defined in the wt.mail.mailhost property.
- Determine if the default threshold values for sending notifications are appropriate for your site.
- Establish the group of users who should be notified. By default, the JMX-Administrators list is used for this purpose.

Checking the Notification Threshold Values and Error Levels

Each JVM has a set of threshold values and error levels that are set by default for notification activities that can occur. For example an e-mail notification is sent whenever the server manager starts. The setup for notifications is handled by the NotificationHandlers MBeans that are available in each JVM. Each MBean identifies who receives the e-mail in the EmailRecipientList Name attribute. By default, the value is JMX-Administrators.

For additional information, see [Accessing Windchill MBeans from JConsole](#) and [NotificationHandlers](#).

Populating the JMX-Administrators List

To populate the JMX-Administrators list for the servlet engine, server manager and method servers on a specific system, you can use the following ant script. From a windchill shell, navigate to Windchill codebase directory and enter the following command:

```
ant -f modifyMBeanConfig.xml setEmailList -DemailAddresses=<comma-delimited list of e-mail addresses>
```

The e-mail addresses specified in the script command are added to each JMX-Administrators list.

Additionally, you can manage the JMX-Administrators lists individually through the MBean interface. From JConsole, navigate to the **Emailer > EmailLists > JMX-Administrators** MBean from an individual connection. Select the **JMX-Administrators** MBean and then click the **Operations** tab. Use the corresponding methods to add and remove addresses. Refresh the attributes to view your additions and deletions.

For additional information about e-mail lists, see [Emailer](#).

Establishing Connections

When you connect your JMX client to Windchill, you connect to a specific servlet engine, server manager or method server JVM process. From each connection, you are able to manage a different set of actions. To get an overall picture of your entire Windchill system, connect to each JVM process in your environment.

Note: The JConsole graphing and charting capabilities only apply to the JVM to which you are connected.

When you are using JConsole on the same machine and as the same user under which a process is running, the process appears in the list of local JMX-monitorable processes upon startup. The server manager and method servers process names are recognizable. For the Tomcat servlet engine process, select the name that starts with org.apache.catalina.

The following sections provide some additional information about your connections.

Servlet Engine Connections

The servlet engine connection provides information about the interaction between the end user client machines and the Windchill servers. Connect to this process to monitor and manage this interaction. Consider monitoring the servlet requests through the ServletRequests MBean that is in this connection.

For additional information about the ServletRequests MBean, see [ServletRequests \(Monitors\)](#).

For remote access to the servlet engine, the port number that is required is set when you configure the servlet engine for remote access. See [Configuring Unauthenticated JMX Connections to Tomcat](#).

Server Manager Connections

The server manager connection provides you with high-level information such as total memory consumption, local and remote server calls, heap memory usage, live threads, logging options for the server manager, and e-mail notifications.

Since method servers have a more unpredictable life cycle than server managers and use dynamically selected port numbers for JMX communication, server managers proxy their method servers' MBeans by default. Thus, by connecting to a server manager, you can manage and monitor its method servers as well. However, when you require detailed information about a method server, you should connect directly to the specific method server.

Note: JConsole does not pick up on the proxied MBean's monitoring memory, and so on, from method server JVMs when targeting the server manager. Also, the

proxy arrangement involves an extra inter-process hop and has somewhat lower performance and higher overhead.

Consider monitoring the server manager through the ServerManager monitoring MBean that is in this connection. The MethodServers monitoring MBean allows you to monitor method servers from the server manager.

For additional information about the ServerManager MBean, see [ServerManager \(Monitors\)](#).

For additional information about the MethodServers MBean, see [MethodServers \(Monitors\)](#).

For remote access to the server manager, the port number that is required is set when you set the properties for remote access. See [Configuring Unauthenticated JMX Connections to Server Managers and Method Servers](#).

Method Server Connections

A method server connection provides details about that specific method server, such as the specific method server memory usage, method server logging options, and e-mail notifications.

Consider monitoring the method server through the MethodContexts monitoring MBean that is in this connection.

For additional information about the MethodContexts MBean, see [MethodContexts \(Monitors\)](#).

Identifying a Method Server Port Number and JMX URL for Remote Access

When JConsole is run under a different user or on a remote machine, connecting to the method server requires the JMX port number of the full JMX URL. Most other JMX consoles, such as jManage, require the full JMX URL for local monitoring as well.

You can obtain the necessary JMX URLs from MBeans in the server manager that represents and controls the JMX proxy connections to the method servers. These MBeans have ObjectNames with the following pattern:

```
com.ptc:wt.processGroup=MethodServers,methodServer=MethodServer.processId
```

These MBeans appear in JConsole as the following in the JConsole MBean tree:

```
com.ptc > MethodServers > MethodServer.processId
```

The RemoteJmxUrl attribute of each of these MBeans provides the necessary remote JMX connection URL.

Remote JMX Monitoring

The out-of-the-box Windchill configuration results in server side processes that only allow incoming JMX connections from the same machine and user account.

Note: This "local connection" capability is not provided by IBM JVMs. You must configure for remote JMX connections in order to connect to Tomcat or the server manager when an IBM JVM is used. Also, the Windows services normally run under a different user account. Unless this is changed, you cannot use local monitoring to monitor any process that is run as a Windows service.

This configuration is used out-of-the-box so as to default to a secure, locked down configuration.

You might want to check up on server processes from other machines and user accounts. JConsole is one of the few JMX consoles capable of forming the type of connections allowed by the out-of-the-box configuration.

Managing a Remote Machine using JConsole

If you are using JConsole to manage a remote machine, you must specify the host name and port number of the specific servlet engine, server manager, or method server on the machine to which you want to connect. Use the **Remote** tab that is on the JConsole connection window to enter this information.

Additionally, you must configure the remote machine to accept remote management. For details on configuring a machine to accept remote management, refer to the following URL:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#remote>

Configuring Unauthenticated JMX Connections to Tomcat

Caution: This configuration should only be used for testing purposes as it is not secure. For details as to how to change these settings and configure for more secure JMX connections, see the documentation on the Sun Web site at the following URL:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#remote>

To allow unauthenticated JMX connections to Tomcat, do the following:

1. Open a Windchill shell.
2. Navigate to the Tomcat installation directory.
3. In the config properties file, enter or edit the `extraJavaArg` configuration property as follows:

```
extraJavaArgs=-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.ssl.need.client.auth=false
-Dwt.jmx.rmi.port=<open TCP/IP port>
```

where *<open TCP/IP port>* can be the number of any open port you choose. This is the port number you specify when connecting remotely.

4. Execute the following command:

```
ant -f config.xml configure
```

5. If you are running Tomcat as a Windows service, stop the service and execute the following command:

```
ant -f config.xml installService
```

This command reinstalls the service with the new parameters (after which point you can restart the service at any time).

Configuring Unauthenticated JMX Connections to Other Servlet Engines

Allowing unauthenticated JMX connections to servlet engines other than Tomcat is similar to the Tomcat procedure except for the following:

- The means of adding to the Java command line options vary by servlet engine and are not specified using `config.properties` or `extraJavaArgs`. For details on how to add options, see the Java documentation for your servlet engine.
- The following addition to the Java command line options to allow local JMX connections would not be done:

```
-Dcom.sun.management.jmxremote
```

Note: This addition is only used for the out-of-the-box version of Tomcat.

Configuring Unauthenticated JMX Connections to Server Managers and Method Servers

Caution: This configuration should only be used for testing purposes as it is not secure. For details as to how to change these settings and configure for more secure JMX connections, see the documentation on the Sun Web site at the following URL:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#remote>

JMX configuration for the server manager is similar to the Tomcat configuration and other Java applications except that the `wt.properties` entries contain the following JMX settings:

- Those that are common to both the server managers and method servers
- Those that are unique to server managers and to method servers

Thus for maintenance, ease of interaction with Technical Support, and others, PTC recommends that the properties described in this section be used.

Method servers are different in that the server manager starts multiple method servers, starts new method servers when old ones become unresponsive, and so on.

Method servers select the first available port in the range specified by the wt.jmx.rmi.minPort and wt.jmx.rmi.maxPort wt.properties settings.

Note: It is best not to use a single fixed port setting from the wt.jmx.rmi.port Java property.

The information necessary to directly connect to a given method server is exposed using the JMX MBean attributes at the server manager level. Do the following:

1. Use the xconfmanager utility to set the following properties (the example is shown on multiple lines; enter the xconfmanager command on one line):

```
xconfmanager -s wt.manager.cmd.common.jmx.args="-Dcom.sun.management.jmxremote
-Djmx.remote.protocol.provider.pkgs=wt.jmx.remote.protocol
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.ssl.need.client.auth=false"
-s wt.manager.cmd.ServerManager.jmx.args="-Dwt.jmx.rmi.port=<open TCP/IP port>"
-t wt.properties -p
```

where *<open TCP/IP port>* can be the number of any open port you choose. This is the port number you specify when connecting remotely.

Note: Be sure to propagate these changes by including the -p option in the xconfmanager command.

Note: You can add these settings using the JMX PropertiesManager MBean, but a server restart is still required. See [Using MBeans to Modify Property Values](#).

Connecting Remotely when there are Firewalls

Firewalls are generally configured to block RMI. Unfortunately, RMI is the built-in wire protocol for "JSR 160" JMX remoting communication in the JDK. JSR 160 specifies a general architecture for JMX remoting protocols, but no other protocols have been included in the latest supported version of the JDK.

Windchill provides a specialized, non-transparent HTTP(S) proxy for JMX JSR 160 remoting communications called JmxProxyServlet. In terms of network access, this proxy only requires normal HTTP(S) access to the Windchill web application.

To use this JSR 160 savvy JMX console and the JMX RMI connections that you have established as a result of reading earlier sections in this guide, perform the following steps:

1. Copy the following files from codebase/lib to your Java JDK jre/lib/ext directory:

WtHttpClientAddOns.jar
WtJmxClientConn.jar

You can do this by browsing to the following URL in the browser, picking these files, and saving the results to your Java JDK jre/lib/ext directory:

```
http(s)://yourHostAndPort/yourWebAppName/lib
```

2. Provide your JMX console with the following JSR 160 connection URL:

```
service:jmx:ptchttp://yourHostAndPort/yourWebAppName/servlet/JmxProxyServlet?jmxurl  
=service:jmx:rmi://jmxTargetHost/jndi/rmi://jmxTargetHost:jmxTargetPort/jmxrmi
```

where you should replace the following:

- **http** with **https** if using https.
- *yourHostAndPort* with your web server host and port.
- *yourWebAppName* with your Windchill web application name, for example, Windchill_9.
- *jmxTargetHost* and *jmxTargetPort* with the host name and port, respectively, that the proxy servlet should connect to using JMX RMI.

For JConsole, the URL required in step 2 can either be provided on the command line or by using the **Advanced** tab of the JConsole connection window.

A utility exists to ease the process of creating the required URLs. To use the utility, the JARs identified in step 1 do not have to be installed in the JDK jre/lib/ext directory as long as they are in the same directory as each other and have their original names.

On the command line, enter the following commands:

- From within the same directory as WtJmxClientConn.jar

```
java -jar WtJmxClientConn.jar -usage
```

- From anywhere (as long as WtJmxClientConn.jar is in the classpath):

```
java com.ptc.jmx.client.Main -usage
```

Note: You can use the help to enter a command line or graphical utility to construct a JMX service URL and optionally launch JConsole with it. On Windows, you can simply double-click on WtJmxClientConn.jar to launch the graphical version of the utility.

Using JMX in a Cluster Environment

To use Windchill MBeans in a cluster, the following properties must be set:

- The java.rmi.server.hostname property on each of the cluster cache slave nodes and on the cluster cache master node must each be set to the physical host name of the system.

The JMX management components use this property to communicate across nodes in the cluster.

- The `wt.rmi.server.hostname` property on each of the cluster cache slave nodes must be set to a commonly known name for the entire cluster. For example if the master node is named A, you can set the `wt.rmi.server.hostname` property for each slave node to A. This assumes that you have already set the local machine host lookup for each slave node to resolve the common name to the current node. To do this, the following entry could be added in the server host file (UNIX = `/etc/hosts`; Windows = `\WINNT\system32\drivers\etc\lmhosts`):

127.0.0.1 A

Note: Do not set the `wt.rmi.server.hostname` property on the cluster master node.

For additional details on cluster setup, see the *Windchill Advanced Deployment Guide*.

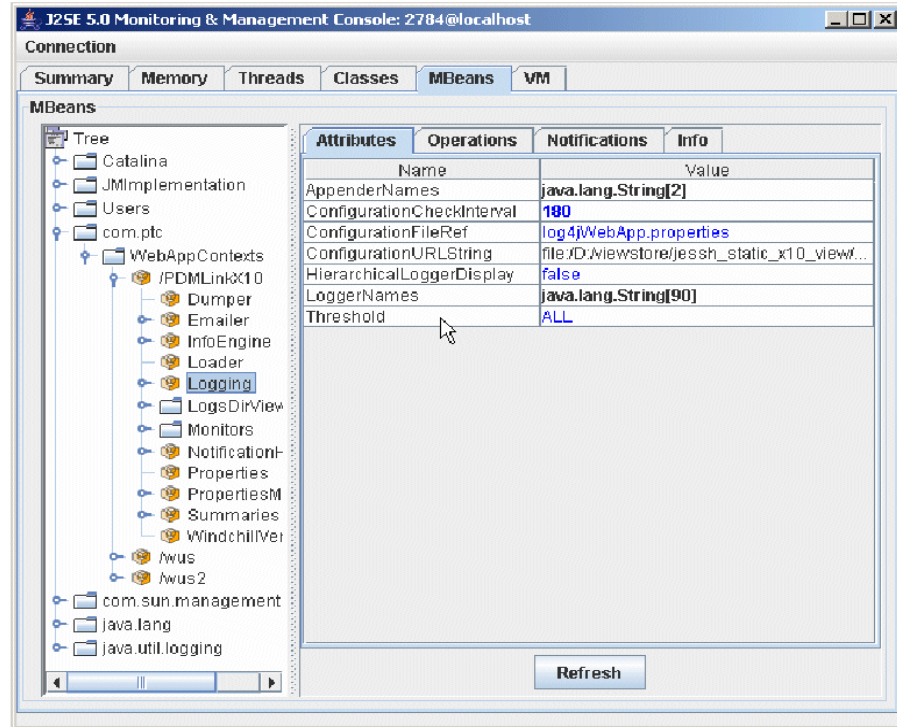
Accessing Windchill MBeans from JConsole

To access the Windchill MBeans, complete the following steps:

1. Access the JConsole as described in [Getting Started with JConsole](#).
2. Select your connection if accessing the local machine, or if accessing a remote machine, select the **Remote** tab and specify the host name and port number for the JVM process on the remote machine.
3. Select the **MBeans** tab.

The **MBean** tab provides a tree panel for navigation and a details panel providing information on the Attributes, Operations, Notifications, and Info for the currently selected MBean. The set of Windchill-specific MBeans is accessible from the `com.ptc` node in the tree. Selecting an MBean from the tree displays the information related to that MBean, allowing you to view and modify the related attributes and execute selected operations.

This example shows the access to the Windchill-specific Logging MBean:



Using MBeans to Modify Property Values

To change the value of a property in a property file, you can use the PropertiesManager MBean. Using this MBean allows you to change values of properties that existing in a Windchill property file.

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

Properties that should be available only to server-side classes are located in separate property files. For example, properties that control access to the database, including the database password, are located outside the codebase directory in the <Windchill>/db directory as db.properties.

The following are common Windchill property files:

- wt.properties

Windchill classes use wt.properties to control the runtime behavior of the Windchill system. These properties are usually read when a class is first loaded and are used to configure status settings within that class.

- db.properties

The Windchill database connection layer uses db.properties to access the database.

Note: 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](#).

To change a property value using JConsole, complete the following steps:

1. Access the **MBeans** tab as described in [Accessing Windchill MBeans from JConsole](#).
2. Do one of the following:
 - If you are in the servlet engine, select **com.ptc > WebAppContexts** from the tree.
 - If you are not in the servlet engine, select **com.ptc** from the tree.

3. Select the PropertiesManager MBean.

The PropertiesManager MBean expands to display all its child nodes.

4. Navigate to and select the property file you want to modify.

5. Select the **Operations** tab.

This tab displays all operations available for this configuration file.

6. Locate the **setProperty** button.

To the right of this button are two fields; one for providing the property name and the other for providing the property value.

Note: You can view the configuration file by clicking **viewFile**. This allows you to view the read only version of the configuration file.

7. Enter the name of the property you want to modify, ensuring it is typed as shown in the configuration file.
8. Enter the new value for the property.

Note: You can review pending changes by clicking **reviewPendingChanges**.

9. After entering the property name and value, click **setProperty**.

Note: You can cancel any pending changes by clicking **clearPendingChanges**. This resets any recently modified properties back to their former values.

10. When you are satisfied with your changes, click **applyPendingChanges** to apply the changes.

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.

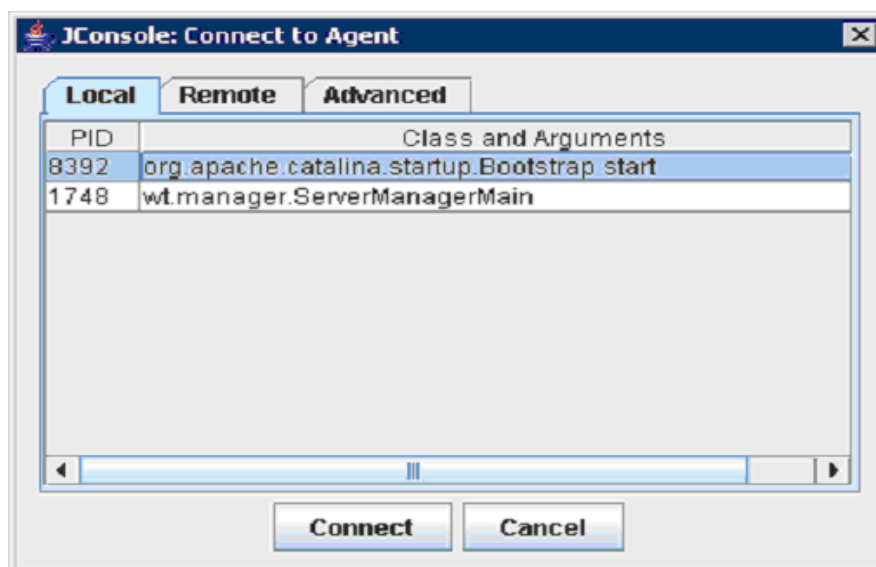
Note: The regenerated property files are used to set system properties when the system is next restarted.

Using MBeans to Start and Stop Your Windchill System

To start and stop your Windchill system, use the `WebAppContexts` MBean from your servlet engine connection.

From JConsole, complete the following steps:

1. Start JConsole and select the connection for your servlet engine.



Note: If you are connecting to a remote machine, select the **Remote** tab and then specify the host name and port number for the servlet engine on that machine.

2. Select the **MBeans** tab from the details panel.
3. Select **com.ptc** from the tree.
4. Select **WebAppContexts** from the tree.
5. Select **Monitors** from the tree.
6. Select **ServerManager** from the tree.
7. Select the **Operations** tab from the details panel.
8. If you want to stop the server manager, click **stopServerManager**.

If the server manager is running, clicking this action stops it.

9. To start the server manager, click **startServerManager**.

If the server manager is stopped, clicking this action starts it.

Using MBeans to Work with Log Files

Logging functions are handled within JMX through the Logging MBean. By accessing the Logging MBean, you can perform a number of operations, including:

- Set the log level for a specific logger.
- Register the loggers.
- View the configuration file for logging.

The Windchill system generates several system log files and any user-customized log files defined in Windchill. The Windchill system log files are enabled when the `wt.logs.enabled` variable in the `wt.properties` file is set to true. Additionally, there is a log file variable associated with each system log file. The individual log files are generated when the log-specific variable and the `wt.logs.enabled` variable are set to true.

Note: Logging changes made using JMX are immediate, but are not saved as part of the configuration and will not be applied on the next restart of the Java Virtual Machine (JVM).

Making logging changes through the Logging MBean works well when making temporary changes for troubleshooting; however, it does not allow you to make permanent changes. To make a change to the log4j configuration that will be performed on the next restart of the process, you must make the change in the corresponding properties file. For more information, see [Managing Logging](#).

The changes will be made within a few minutes. You can execute the reconfigure operation on the Logging MBean to make this change immediately.

From JConsole, complete the following steps to access the Logging MBean:

1. Determine which log files you want access to and selection the appropriate connection.
2. If you are in the servlet engine, select **com.ptc > WebAppContexts** from the tree.
3. If you are not in the servlet engine, select **com.ptc** from the tree.
4. Select the Logging MBean from the tree.
5. Select the **Operations** tab for the Logging MBean to view the available actions that can be performed.

From the **Operations** tab, you can execute a number of operations related to logging. Click **registerLoggers** to create an MBean that will represent each log4j logger and will register it within a Logger folder underneath the Logging MBean in the MBean tree. By default all loggers are registered in a single list by their full names.

To get a hierarchical display of the loggers, change the Logging MBean **HierarchicalLoggerDisplay** attribute value to true. This attribute is located on the **Attributes** tab for the Logging MBean.

Using MBeans to View and E-mail Log Files

The LogDirsViewer MBean allows you to view and e-mail logs.

From JConsole, complete the following steps to e-mail all logs in the *<Windchill>/logs* directory:

1. Start JConsole and select the connection for the server manager.
2. Do one of the following:
 - If you are in the servlet engine, select **com.ptc > WebAppContexts** from the tree.
 - If you are not in the servlet engine, select **com.ptc** from the tree.
3. Select the **LogDirsViewer** MBean.
4. Select **wt.logs.dir**.
5. Select the **Operations** tab.
6. To e-mail all log files:
 - a. Locate the **emailFiles** button.
 - b. Enter the recipient addresses and subject information in the fields next to the button.

Note: By default, a separate e-mail is sent for each log file. To send only one e-mail containing all log files, type true in the third field (labeled as oneMessagePerFile), which is located to the right of the subject field.

- c. Click **emailFiles**.

To view and e-mail a specific log file:

1. Start JConsole and select the desired process.
2. If you are in the servlet engine, select **com.ptc > WebAppContexts** from the tree.
3. If you are not in the servlet engine, select **com.ptc** from the tree.
4. Select the LogDirViewers MBean from the tree.
5. Select wt.logs.dir.
6. Select the **Operations** tab for the Logging MBean to view the available actions that can be performed.
7. Click **refresh**. This creates an MBean to represent each log file stored in *<Windchill>/logs*.
8. Select the log file you want to view from the wt.logs.dir node.
9. Select the **Operations** tab.
10. To view the log file, click **viewFile**.
11. To e-mail the log file:
 - a. Locate the **emailFile** button.
 - b. Enter the recipient address and subject information in the fields next to the button.
 - c. Click **emailFile**.

Windchill MBeans

As mentioned earlier, the Windchill MBeans are accessible from the com.ptc node in the JConsole. Selecting a Windchill MBean from the tree displays the information related to that MBean.

The following sections provide descriptions of some of the more prominent Windchill MBeans. The MBeans included are listed in ascending alphabetical order and the **(Monitors)** designation is included for the MBeans that appear under the **Monitors** node in the JConsole tree display.

DirContexts (Monitors)

The DirContexts MBean monitors JNDI requests made by the system. It is similar in intent to the servlet request and method context monitoring MBeans. The DirContexts MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

Dumper

The Dumper MBean provides:

- Operations to dump stack traces, attributes MBean of all or selected MBeans (as XML), or text file contents to a new window in JConsole.
- Operations to e-mail any of this data to individual e-mail addresses, or a mailing list managed by the Emitter MBean.
- Provides ability to apply XSLT stylesheet to MBean attribute XML.
- Emitter MBean performs actual e-mailing.
- A DefaultXsltStylesheet attribute which is a file, URL, or Java resource path reference to an XSLT style sheet used by default for MBean attribute XML styling. Out-of-the-box this defaults to a simple XSLT style sheet that produces easy-to-read HTML.

Emitter

The Emitter MBean provides:

- The underlying e-mailing facilities used by all other MBeans.

These e-mailing facilities do not use Windchill's persistent queues as they are intended to be usable from any Java process; however, they do e-mail in the background and there is a retry function that is configured to resend a message a configurable number of times at a configurable interval prior to giving up (and logging an error).

- Manages e-mail lists for use by other MBeans in e-mailing to a group of e-mail addresses.

These e-mail lists are described by EmailList MBeans shown nested beneath the Emitter MBean in JConsole.

- Attributes for configuring the appropriate e-mail server, port, and so on, to use.

These are read from wt.properties or Java system properties by default, but can be overridden here.

To configure the EmailList, see [Populating the JMX-Administrators List](#).

GarbageCollection (Monitors)

The GarbageCollection MBean monitors the percentage of overall time spent in garbage collection during intervals of a configurable duration. If this value exceeds a configurable threshold a JMX notification is sent.

IeCalls (Monitors)

The IeCalls MBean provides aggregate statistics about outgoing Info*Engine calls. The IeCalls MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

IeContexts (Monitors)

The IeContexts MBean monitors Info*Engine requests being handled by a method server. It is similar in intent to the servlet request and method context monitoring MBeans. The IeContexts MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

InfoEngine

The InfoEngine MBean is the root MBean for an Info*Engine System under which service beans and other administrative information beans are exposed. The Loader MBean manages this MBean.

Loader

The Loader MBean can detect changes to the configuration file, unload the existing MBeans, and instantiate new MBeans from the file. Changes to MBeans are not automatically detected nor are the MBeans automatically/periodically saved at any time.

Rather, if an administrator wants to save configuration changes they have made so they extend beyond the lifespan of the server JVM, they can use the save operation at their discretion.

Logging

The Logging MBean provides the ability to examine and dynamically change the log4j logging configuration.

The Logging MBean itself can detect when the log4j configuration file has changed and apply the altered configuration. It cannot save a log4j configuration change made via JMX back to the logging configuration file. Instead changes made to the logging configuration via JMX are for the life span of the target JVM only.

To make permanent changes to a log file, refer to [Using MBeans to Work with Log Files](#) for further instructions.

wt.logs.dir

The wt.logs.dir MBean produces MBeans for viewing/tailing and e-mailing log files found in the Windchill logs directory (and its subdirectories). The intent is to provide basic capabilities for legacy (non-log4j) logs.

The wt.logs.dir MBean is currently instantiated in server manager, method server, and servlet engine processes.

Note: The MBeans that represent the logs directory and its subdirectories (including the wt.logs.dir MBean itself) do not poll for changes in the corresponding directories; instead, explicitly use the **refresh** operation on the MBeans to view changes.

Memory (Monitors)

The Memory MBean allows percentage thresholds to be set for various memory pools. It also allows an overall memory usage percentage threshold to be set and periodically checks to see if this has been exceeded. JMX notifications are sent when any of these thresholds are exceeded. It also allows the current percentage usage values to be examined via its attributes.

MethodContexts (Monitors)

The MethodContexts MBean monitors method server method contexts and thus is only instantiated in method servers.

Use this MBean to establish what data is logged in the method server.

Tip: To get the data needed for tracing anything through Windchill, add the `IeContextIds` and `ServletRequestIds` to the `ContextLoggerOutputAttributes`.

Other values may also be helpful. For example, assume you have set the following values for `ContextLoggerOutputAttributes`:

```
StartTime, Id, IeContextIds, ServletRequestId, UserName, ClientHost, TargetClass,
TargetMethod, AccessLogDetail, Redirects, AverageActiveContexts,
ElapsedTotalSeconds, ResultClass
```

If the `ContextLoggerLevel` is set to `INFO`, then the following is a sample of the output in the method server log:

```
INFO : wt.method.MethodContextMonitor.contexts.rmi - 2007-11-22 16:02:33.786,
f9bdna1m;3316;105, -, f9bdna1m;3316;104, peter, 192.168.0.3, wt.httpgw.HTTPServer,
processRequest,
/wt.content.ContentHttp/viewContent/1_wb_tech-10_blue_prt.jpg?u8&HttpOperationItem=
wt.content.ApplicationData%3A996917&ContextHolder=wt.viewmarkup.DerivedImage%3A9969
13&forceDownload=true, 0, 1.9606879606879606, 0.407, wt.httpgw.HTTPResponseImpl
```

In this sample:

- The request ID is: `f9bdna1m;3316;105`

- It is not an Info*Engine request: (-)
- The ServletRequestID is: f9bdna1m;3316;104
- The user is peter
- There are no redirects; there is about 2 average active contexts and the request took .407 seconds

Notice that the delimiter is a comma. The semicolon is considered part of the contextId string

Tip: From the **Operations** tab, you can use the **captureActiveContexts** button to capture running contexts They will then display under MethodContexts MBean.

MethodServer

The MethodServer MBean provides a brief summary of some top-level method server information. This MBean is only available for the method server.

MethodServers (Monitors)

The MethodServers MBean is only instantiated in the server manager and provides information on a server manager's method servers using child MBeans which it manages.

By default these child MBeans proxy all of the MBeans from the corresponding method server JVMs into the server manager JVM (as nested MBeans) to allow an administrator to easily manage and monitor details of a server manager and all of its method servers by just connecting to the server manager. Each of the MBeans representing method servers also exposes a RemoteJmxUrl attribute which provides a JMX URL that can be used to connect a JMX client directly to the given method server.

The MethodServers MBean fires JMX notifications whenever:

- A method server is registered
- A method server is removed
- The JMX connection to a method server is lost.

The MethodServers MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

NotificationHandlers

The NotificationHandlers MBean allows you to create and manage NotificationHandler MBeans. The NotificationHandlers MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

NotificationHandler MBeans allow an administrator to specify logging and/or e-mailing actions that are to take place upon the occurrence of a JMX notification. One or more MBeans can be listened to by filtering on the types of notifications that are of interest. Administrators can specify additional data to query from the

MBean generating the notification or other MBeans when a notification occurs, which is specified to be included in logs and e-mails.

Some items to note:

- MBeans are referenced via the ObjectName syntax - either an exact ObjectName or an ObjectName pattern. See the ObjectName documentation for more information. Also note that JConsole provides the ObjectName of each MBean on the Info tab for that MBean.
- If an e-mail subject is not provided, one will be automatically generated upon receipt of a notification from the notification's message and other data.
- One can specify multiple attributes via "*" or fields within a composite data attribute by using "attributeName.fieldName" syntax.
- The NotificationHandler MBean's log level is one of the few logger level attributes that is actually a persistent attribute (meaning that it is reapplied when the JVM restarts). In most other cases, such attributes are merely shortcuts to allow per-JVM-session tweaks and overrides to log4j log levels. An exception is made here to allow easier dynamic creation and specification of new notification handlers.

ProcessCpuTime

The ProcessCpuTime MBean monitors the percentage of overall CPU time consumed by the target process over the last configurable time interval. If this metric exceeds a configurable threshold, a JMX notification is sent.

Properties

The Properties MBean exposes current WTProperties settings, providing a read only view of those settings. It is instantiated out-of-the-box in all server JVMs.

PropertiesManager

The PropertiesManager MBean exposes the properties files, providing you with a method to modify the property values in the files. It is instantiated out-of-the-box in all server JVMs.

Note: Your changes are written to the site.xconf file and the affected property files are regenerated using your changes. The regenerated property files are used to set system properties when the system is next restarted. For additional information, see [Using MBeans to Modify Property Values](#).

Queues (Monitors)

This MBean is instantiated in the background method server and provides management and monitoring for the queues. It is not managed by the Loader MBean.

Reporting System

The Reporting System MBean periodically pings the Cognos (when it is installed and enabled) and reports on its availability, including the providing of notifications when the system availability changes. It also exposes various administrative operations against it.

This MBean can be deployed in both servlet engine and method server JVMs, but is only deployed in the servlet engine by default. The Reporting System MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

RMICalls (Monitors)

The RMICalls MBean provides statistics on the number of total and "recent" outgoing RMI calls to Windchill method servers calls and the average time consumed during that time span.

ServerManager

The ServerManager MBean provides a brief summary of some top-level server manager statistics. It is available for the server manager.

ServerManager (Monitors)

This MBean appears in the "Monitors" grouping in JConsole and differs from the ServerManager MBean which appears outside this grouping. This MBean runs in the servlet engine and monitors the local server manager via pings. It provides ping statistics, notifications upon server manager status change, and operations to start, stop, and restart the local server manager.

ServletRequests (Monitors)

The ServletRequests MBean monitors servlet requests and thus is only instantiated in the servlet engine.

Using the attributes for this MBean, you can set the logger levels to gather statistics. The following table shows the levels of statistics that can be gathered:

Statistics Grouping	Description
Total	Statistics accumulated since web application startup.
Baseline	Statistics accumulated since last invocation of the <code>resetBaselineStatistics()</code> operation (or since web application startup if this operation is never invoked).

Statistics Grouping	Description
Summary	Statistics accumulated for a time interval specified by the SummaryIntervalSeconds attribute. Summary statistics are logged but not reflected in the MBean attributes.
Recent	Statistics accumulated for a time interval specified by the RecentIntervalSeconds attribute. The corresponding Recent* MBean attributes are updated at the end of each measurement interval.
Request	Data for an individual servlet request.

Tip: You can set up which servlet requests are tracked and which statistics are gathered by setting values for the RequestLoggerOutputAttributes and StatisticLoggerOutputAttributes in this MBean.

Tip: To trace a user session into the method server, you can set RequestLogLevel to INFO and set the Id and SessionID in RequestLoggerOutputAttributes.

Tip: The RequestLoggerUseShortFormat attribute controls whether the Property Name is prepended to the value in the logs. If you put the column headings in Excel, you can set this to false and import the data more easily for analysis.

Tip: From the **Operations** tab, you can use the **captureActiveContexts** button to capture running contexts. They will then display under ServletRequests MBean.

The ServletRequests MBean also provides a MaxAverageActiveRequestsThreshold attribute. When the average (concurrently) active requests over a summary statistics interval exceeds this threshold a JMX notification is sent. For information about notifications, see [Setting Up Your JMX Environment](#).

ServletSessions (Monitors)

The ServletSessions MBean monitors servlet sessions and thus is only instantiated in the servlet engine.

The ServletSessions MBean provides capabilities similar to those of the ServletRequest MBean, but for servlet sessions rather than for servlet requests. One difference is that the ServletSessions MBean does not provide a "capture" operation. Instead, you can set the SessionRegistrationEnabled attribute to true from which point new sessions will be represented using nested Session MBeans.

Summaries

The Summaries MBean is very similar to the NotificationHandlers MBean. It allows you to create and manage Summary MBeans which allow you to log attributes collected from one or more MBeans on a periodic basis. Thus Summary MBeans are much like NotificationHandler MBeans except that the trigger is a timer, not a notification, and that only logging, not e-mailing is a possible result of the trigger.

Out-of-the-box a single summary MBean is instantiated to provide information summarizing basic JVM statistics every 10 minutes.

Support

The Support MBean provides simple, straightforward operations to e-mail logs and JMX MBean data to PTC technical support. It associates this with an existing technical support call number. The Support MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

UserSpecified (Monitors)

The UserSpecified monitors MBean allows the (administrative) user to create their own instances of the various built-in Java subclasses of `javax.management.monitors.Monitor` which are then managed and persisted through the UserSpecified monitors MBean.

The creation of the Monitor instances is done via operations on the UserSpecified monitors MBean whereas further configuration of the Monitor instances is done via attributes and operations on the Monitor instances themselves (which are also MBeans). The UserSpecified monitors MBean is instantiated out-of-the-box in all server JVMs.

VaultSites

The VaultSites MBean monitors the accessibility of Windchill vaulting sites to HTTP(S) requests. It periodically sends short, simple "ping" requests to all sites (including itself) and logs results. Aggregate statistics and last ping results are available via the `SiteStatusInfo` attribute. A JMX notification is generated whenever a site does not respond to a ping.

The VaultSites MBean is deployed into the method server by default; the file vaulting services requires this deployment.

WebAppContexts

The WebAppContexts MBean is only instantiated in the servlet engine and provides a top-level information node for an entire Web application.

WindchillVersion

The WindchillVersion MBean provides a summary of Windchill version information, including the products, temporary patches, and locale support that have been installed. The WindchillVersion MBean's existence, life cycle, and configuration persistence are controlled by the Loader MBean.

WrappedSockets

The WrappedSockets MBean provides visibility into the number of Windchill "WrappedSocket" RMI sockets and number of bytes sent and received over them. It is currently instantiated in both method server and server manager processes.

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. 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	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. 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.
-d or --describe	Lists the values that are currently set and the corresponding XCONF file where each value is set for the specified properties. This option executes after all parameter setting options and the -p option have executed.

Parameter Name	Description
-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>
-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>

Parameter Name	Description
-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>
-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>

Parameter Name	Description
--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><property_file></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>
-t or --targetfile	<p>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.</p> <p>Use this parameter in conjunction with either the -s or --setfromfile parameter.</p>

Parameter Name	Description
-u w or --unix win	<p>Indicates the platform for which the property files are to be generated. Normally, the current platform settings determine the format of the property files.</p> <p>Include this parameter when you want to generate property files for a specific platform that is not the current platform.</p> <p>Specify -u or --unix for UNIX platforms.</p> <p>Specify -w or --win for Windows platforms.</p>
--undefine	Resets the specified properties such that their values will be null (instead of an empty string) when read through a <code>java.util.Properties</code> 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	<p>Validates a specific file as a declarative XCONF file.</p> <p>Returns a non-zero result if file cannot be validated.</p>
--validatefilesasdecl	<p>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.</p> <p>Returns a non-zero result if any of the files cannot be validated.</p> <p><i><declar_list_file></i> is either a full URL or relative file path to the file containing the list of declarative XCONF files you want to validate.</p>
--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><site_list_file></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>

Parameter Name	Description
--validateassite	Validates a specific file as a site-specific XCONF file. Returns a non-zero result if file cannot be validated. <i><site_xconf></i> is either a full URL or relative file path to the site-specific XCONF file you want to validate.

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 `WCtemp` directory that is under the Windchill installation directory [as defined by `$(wt.home)`]:

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

Assuming that the command was executed from the Windows `C:\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 UndefineProperty element for each property that was specified (in this case, only wt.services.service.1160), and then propagates the change to property files that have the specified properties (in this case, only wt.properties).

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).

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 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 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 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`.

Disabling WebEx Meetings

The `wt.meeting.centerURL` property provides the URL for the WebEx server. If this property is absent from the `wt.properties` file, WebEx meetings become disabled. You can remove the `wt.meeting.centerURL` property by undefining it using the `xconfmanager` utility. Undefining the property results in its removal from the `wt.properties` file.

In the following example, the `wt.meeting.centerURL` property is undefined and then the change is propagated:

```
xconfmanager --undefine wt.meeting.centerUrl
xconfmanager -p
```

Note: You must issue the `xconfmanager -p` command to ensure the modification is propagated.

Configuring Password Management

The password management feature allows Windchill users to change their passwords using the Windchill interface. If you are using the Aphelion Directory Service for storing Windchill users and groups, and are using Apache and Tomcat as your Web server and servlet engine, you can easily configure password management.

Note: To be able to change their passwords, Windchill users must be able to write to the directory service where the user information is stored. If users are stored in your Aphelion Directory Service, allowing users to change their passwords is set up by default.

Configuring password management in an environment that uses Aphelion, Apache, and Tomcat consists of completing the following tasks:

- Enabling the user interface that allows Windchill users to change their passwords. See [Enabling the Password Change User Interface](#).
- Setting up a password policy in the Aphelion Directory Service. See [Setting Up a Password Policy](#).
- Deciding how to notify users when their password will expire. See [Deciding on a Password Expiration Notification Process](#).
- Customizing PTC help topics to provide user information on the password change requirements and log in problems. See [Customizing Password Change and Apache Login Help Topics](#).

Read the following sections for information on completing these tasks.

Enabling the Password Change User Interface

By default, the Aphelion Directory Service is set up to allow users to modify their own password.

Using the xconfmanager utility, you can enable password change by setting the wt.org.services.userPasswordChangeEnabled property value to *true*.

Following is an example of how the wt.org.services.userPasswordChangeEnabled property is enabled. Enter the xconfmanager command on one line:

```
xconfmanager -s "wt.org.services.userPasswordChangeEnabled=true"
-t "<Windchill>/codebase/wt.properties" -p
```

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

Note: After changing the value of the wt.org.services.userPasswordChangeEnabled property, stop and re-start Tomcat to ensure the change becomes effective.

If you have set the wt.org.services.userPasswordChangeEnabled property to true, users can change their password by selecting the **Profile** option from the **Home Utilities** page. From the user information page that displays, select **Edit Password** from the **Actions** drop-down list to invoke the **Edit Password** window.

The user information page is also accessible from the **Team** page. Display users on the **Members** table and click the information icon for a user to display the user information page.

Users can click the help icon on the **Edit Password** window to access information on how to change the password from the **Edit Password** window. This help can also contain site information about the password requirements you have established. To add your requirements to this help, customize the passwordmgmt/PasswordPolicy.html help topic. For example, you may require that the password be at least five characters and contain at least one integer. For information about customizing this help topic, see [Customizing Password Change and Apache Login Help Topics](#).

Setting Up a Password Policy

Your site password policy is defined in your LDAP directory service and enforced by that service. Windchill does not place any restrictions on passwords.

By default, the Aphelion Directory Service defines no password policy. However, Aphelion supports a password policy that includes password expiration, password history rules, password syntax and length rules, and more. For information on configuring a password policy in Aphelion, see the *Aphelion Directory 2003.3 Administration Guide*.

The password policy setup is done through the Aphelion Web Tools. These tools are automatically configured in an Express installation. Configuring the Aphelion Web Tools is an option that is selectable when installing your Windchill solution

through the Advanced installation. For details on this option and how to manage Aphelion, see *Windchill Installation and Configuration Guide - Advanced*.

If the web tools were not configured, you can manually configure them using the WebToolsConf\ConfigAphelion.xml script file available from the Aphelion installation directory. Open this file using an XML editor and read the instructions contained in the Examples section of the file to learn how to run the script to configure the Aphelion Web Tools.

Deciding on a Password Expiration Notification Process

Note: Windchill does not send users any notification that their password is about to expire.

You must manage the user notification of password expirations through your site policies outside of Windchill. However, Windchill provides the following Apache error pages that can be customized with site specific information:

- When Apache returns a 500 error (indicating that an internal server error has occurred), the `apachelogin/apacheerror.html` displays. Apache returns this error page when a user is locked out of Aphelion. This page is returned whenever an unexpected condition prevents Apache from completing an HTTP request. Therefore, you should keep the text on the page very general.
- When Apache returns a 401 error (indicating that the login authorization has failed), the `apachelogin/apachelogin.html` displays.

PTC supplies English and Japanese versions of these files.

Customizing Password Change and Apache Login Help Topics

You can customize the content of the password policy and Apache login help topics to ensure that users have access to your site information.

The WebHelp systems that are delivered with Windchill are stored in `<Windchill>/codebase/wt/helpfiles/help_xx/online` directory, where `<Windchill>` is the Windchill installation directory and `xx` is the two-character language suffix (for example, `en` for English).

Help for Password Change

The password change help is located in the passwordmgmt directory. The files in this directory are used to deliver the topic content and are standard Windchill online help files. The content is contained in the following files:

- PasswordEdit.html -- for information about editing a password.
- PasswordPolicy.html -- for general password policy information, including the expiration policy.

You can modify the content of these files. Follow the instructions found in the [Customizing Online Help](#) chapter.

Help for Apache Login

The help for the Apache 500 and 401 errors is located in the apachelogin directory.

There are only two files in this directory that are accessed by Apache and follow the Apache conventions for online help. These files end in an extension that indicates the content language of the file. PTC supplies only English and Japanese versions of the files. The .en extension is for English and .ja is for Japanese.

The following lists the English files:

- apacheerror.html.en -- for internal server error information to display when the Apache error number is 500.
- apachelogin.html.en -- for login authorization information to display when the Apache error number is 401.

The Japanese version of the files end in .ja and are located in the *<Windchill>/codebase/wt/helpfiles/help_ja/online/apachelogin* directory.

You can modify the content of these files in the same manner that you modify the contents of standard Windchill online help files. See the [Modifying an Existing Topic](#) section in the [Customizing Online Help](#) chapter

Setting Windchill Desktop Integration Preferences

Windchill Desktop Integration (DTI) allows you to create and edit Windchill objects in Microsoft Office applications.

Making Sure the Browser Is Set Correctly

Disable all pop-up blockers using an option such as **Allow popups from the following websites**.

Examples of pop-up blockers include Internet Explorer, Google, and other spy ware tools.

Many web browser and third-party applications offer pop-up blocker capabilities to prevent unwanted pop-ups from being opened on a local machine. This can interfere with some Windchill operations that automatically open new windows, preventing you from completing certain functions, such as downloading object content.

Select any **Automatic Prompting for file downloads** setting.

Adding a Windchill Server to the Local Intranet Zone.

If you have a Windchill server in your trusted list, but not in the local intranet zone, you must enable the automatic prompting for file downloads preference. You need to add your Windchill server name in the **Add This Web site to the zone** field.

Setting Windchill Desktop Integration Preferences

In the **Preference Manager**, go to **Attachment > File Download Mechanism** and select **Use Windchill Desktop Integration (DTI) functionality to download files**.

Presently the download defaults to the basic browser functionality.

Note: This step must be implemented or the server will not work correctly.

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. Enter 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 15:03:46.0	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 wadmin 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	EEmail	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).

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 Windchill Index Search files should be backed up on a regular basis. For information on backing up indexing files, see the *FAST InStream Operations Guide*.
- 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 and when backed up must be synchronized with database backups. 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.

Managing Logging

Windchill log messages contain exception tracebacks and other information that can be used for troubleshooting application behavior.

Starting in Windchill 9.0, Apache log4j is now being used as the primary mechanism for managing and issuing log messages. Some legacy logging has been modified to make use of log4j, but a large amount of previously existing Windchill logging capabilities remain as they were in previous releases and are still managed by Windchill property files configuration settings.

In future releases, more legacy Windchill logging capabilities will be migrated to log4j.

To manage logging, you should be familiar with the following:

- The property settings that affect logging that are in the wt.properties file. For information about these properties, see the properties.html file that is located in the <Windchill>/codebase directory and the [Out-of-the-box wt.property Log File Settings](#) section.
- General information about the log4j package.

The following URL takes you to an introduction to log4j:

<http://logging.apache.org/log4j/docs/manual.html>

- The log4j loggers that are used for Windchill logging.

For descriptions of the out-of-the-box Windchill loggers, see the following file:

<Windchill>/codebase/loggers.html

- The property settings that affect the log4j logging. These property settings are in the properties files located in <Windchill>/codebase/WEB-INF:
 - log4jIeServer.properties - Controls standalone Info*Engine log4j logging
 - log4jMethodServer.properties - Controls method server log4j logging
 - log4jServerManager.properties - Controls server manager log4j logging
 - log4jWebApp.properties - Controls log4j logging for the Windchill web applications running in servlet engine.

Additionally, the <Windchill>/codebase/log4j.properties file is used as the default configuration file that log4j uses if another configuration file is not specified.

Changes to the log4j properties files listed previously must be done by editing the files instead of using the xconfmanager utility. If you make any manual changes to these files, be sure to back up the files as described in [Windchill Maintenance Best Practices](#).

- How to make temporary changes to log settings using the Logging MBean. See [Using MBeans to Work with Log Files](#).
- How to view and e-mail log4j log files. See [Using MBeans to View and E-mail Log Files](#).
- How to use the xconfmanager utility to make permanent changes to the log settings for legacy logging and verbosity properties. For information on using the xconfmanager utility, see [Using the xconfmanager Utility](#).

The following sections provide additional information on wt.property log file settings and log event configuration settings.

Out-of-the-box wt.property Log File Settings

This section provides information about some of the common log file settings in the wt.properties file that are used in Windchill.

The wt.logs.dir property sets the default directory used for log files. Out of the box, this property is set to <Windchill>\logs.

Each log file is enabled or disabled using .log.enabled properties. If logging is enabled, there are additional property settings to check. For example, check the .log.append properties, which specify whether a log file is appended to or overwritten when the associated application is started.

Additionally, the log file names and locations are controlled by the .log.file properties that you should check.

For example, the following properties are the out-of-the-box settings for the server manager and method server logging options:

```
wt.manager.log.append=true
wt.manager.log.enabled=$(wt.logs.enabled)
wt.manager.log.file=
$(wt.logs.dir)$(dir.sep)ServerManager-$(DATE(yyMMddHHmm))-$(wt.jvm.id).log
wt.manager.log.maxsize=9
wt.manager.log.tee=true
wt.manager.log4j.config=WEB-INF/log4jServerManager.properties

wt.method.log.append=true
wt.method.log.enabled=$(wt.logs.enabled)
wt.method.log.file=
$(wt.logs.dir)$(dir.sep)MethodServer-$(DATE(yyMMddHHmm))-$(wt.jvm.id).log
wt.method.log.maxsize=9
wt.method.log.tee=true
wt.method.log4j.config=WEB-INF/log4jMethodServer.properties
```

The default log file names include the following:

- The `$(DATE(format))` macro is used to construct date-dependent file names. See the Javadoc for the class `wt.util.WTProperties` for information about the `$(DATE)` macro.
- The use of `wt.jvm.id` ensures that a separate log file is created for each method server and server manager process. The JVM ID is listed toward the top of the startup log when the process starts. This allows you to easily determine which log file has the information for a specific process.

The maximum size of a log file is configurable. For example, the size for the server manager log file is set in the `wt.manager.log.maxsize` property. After the size of a log file reaches the maximum size, the existing file is renamed by adding a numeric suffix and a new log file is started with the existing file name.

The main method server and server manager log files contain all of messages generated from the existing code as well as those messages generated using the log4j loggers. For information on working with just the output from the log4j log files, see the [Using MBeans to Work with Log Files](#) section earlier in this chapter.

For information on viewing and e-mailing log4j log files, see the [Using MBeans to View and E-mail Log Files](#) section earlier in this chapter.

Configuring Which Log Events Are Stored

By default, Windchill stores only ERROR and FATAL log events (except where the configuration has already been extended to output other log events) in Windchill log files. Therefore, you must usually change the configuration to see other log events, such as TRACE, DEBUG, INFO, or WARN.

To turn on a given logging level for a given area, determine if the area has a corresponding logger by viewing the loggers from the **Operations** tab (for details, see [Using MBeans to Work with Log Files](#)):

- If a log4j logger has not been implemented, locate the property that sets the log level in wt.properties and set the level in that property using the xconfmanager utility.
- If a log4j logger has been implemented, you can turn on a given logging level for all loggers or adjust the logging level at a more specific level.

To turn on a given logging level for all loggers, find the log4j.rootLogger property in the appropriate properties file and change the value before the first comma ("error" by default) to the desired level. For example, change:

```
log4j.rootLogger=error, ...
```

to:

```
log4j.rootLogger=debug, ...
```

Turning on logging for all loggers usually results in too much data in the log output; therefore, a better practice is to adjust the logging level at a more specific logger level. To do this, you can append a line of following form to the properties file:

```
log4j.logger.<targetLoggerName>=<desiredLogLevel>
```

where *<targetLoggerName>* is the name of the logger and *<desiredLogLevel>* is the log level.

For example, to set the log level for wt.servlet.ServletRequestMonitor.stats to INFO, add the following property to the log4jWebApp.properties file:

```
log4j.logger.wt.servlet.ServletRequestMonitor.stats=INFO
```

Note: Adding this property also causes the default log level of all wt.servlet.ServletRequestMonitor.stats.* loggers to be set to INFO. For example, the level of the wt.servlet.ServletRequestMonitor.stats.summary logger is also set to INFO unless the level of this logger is otherwise specified.

Changes to log4j configuration files may go unnoticed for as long as a few minutes as the checks for modifications to these files take place on a periodic basis. To force an immediate reload of the log4j configuration file, change the configuration file modification check interval, or make temporary changes to the log4j configuration without changing the configuration files, you must use the JMX MBeans.

Note: Manual changes made to property files may not be preserved when you apply maintenance releases; ensure that any files you manually change are backed up.

Monitoring Temporary Files

Windchill stores temporary files in the *<Windchill>/temp* directory.

When your Windchill system is running as expected, you can remove any files in this directory that are older than twenty four hours.

For example, when specific operations, such as import and export operations, fail temporary files can be stored in this directory.

Additionally, Visualization services can store files in the wrmfTransport and pubtemp directories under the temp directory.

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 *Windchill Installation and Configuration Guide - Advanced*.

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 Java EE servlet container
- A relational database server
- An LDAP server

- The Windchill Index Search 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
```

Advanced Configurations

For information on setting up multiple method servers and configuring background method servers, see the *Windchill Advanced Deployment Guide*.

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, 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 want 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 - Advanced* for more information about specifying anonymous access.

Troubleshooting User Authentication

You can perform troubleshooting with Internet Explorer and other authentication configuration tools.

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
wt.scheduler.purgeHistoryItems	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.
wt.scheduler.purgeHistoryItemsInterval	Specifies the number of days between the cleanups of history items that are thirty days old.
wt.scheduler.purgeHistoryItemsOlderThan	Specifies the age in days of the items that are 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.

Changing the Revision of Objects

As the site administrator, you can use the wt.vc.ChangeFirstRevisionLabelUtility command line utility to change the revision of objects that fit the following criteria:

- The object was created using one of the following types or a subtype of one of the following types:
 - WTPart
 - WTDocument
 - EPMDocument
 - EPMSEPFamilyTable
- The object has not been revised and there are no view versions; there is only one version of the object.

The revision label you specify must be a label that is in the current versioning scheme that is in place for the object type. For information about versioning schemes, see the *Windchill Business Administrator's Guide*

To obtain help instructions for this utility, enter the utility name from a windchill shell:

```
java wt.vc.ChangeFirstRevisionLabelUtility
```

The following example changes the initial revision label of an existing part named part3 to A:

1. Open a windchill shell on the method server.
2. Enter the following command:

```
java wt.vc.ChangeFirstRevisionLabelUtility -name part3  
-newRev A
```

3. Enter the user name and password for the site administer when prompted.

If there is only one part named part3 in the system, then the initial revision label is changed to A. If there are multiple objects with this name, the utility lists all of the objects using their object identifiers. You can copy the object identifier for an object from the list and reenter the command using the -oid argument in stead of -name.

Note: To limit the search to a specific organization context, you can include the -org parameter.

This utility can be used when an initial data set has been loaded and you want to change the initial revision label for the objects that were loaded. For example assume you are using the MilSpec multicharacter series versioning scheme which uses hyphen (-) as the initial revision label. Using this utility, you can replace the hyphen with the letter A (or any other character that is defined in the versioning scheme that is in place).

Changing Views Assigned to Parts

As the site administrator, you can use the wt.vc.views.ReassignSingleViewCleanser command line utility to change existing views that are assigned to parts (type WTPart or subtype of WTPart).

Note: To use this utility to change the view, all iterations for a given part must be assigned to the same view. The utility can not change views when the iterations of a part are assigned to different views. For example if revision A.1 is assigned to the Design view and revision A.2 is assigned to the Manufacturing view, you cannot use this utility to change the view.

The view you specify must be a product structure view that has been established through the View Administrator. For information about views, see the *Windchill Business Administrator's Guide*.

To obtain help instructions for this utility, enter the following command from a windchill shell:

```
java wt.vc.views.ReassignSingleViewCleanser -help
```

For example, assume that parts have been created under the Demo organization context, where some parts are associated with the Design view and some parts are associated with the Manufacturing view.

To change the view association for all parts created under the Demo organization context that are using the Design view to use the Manufacturing view, complete the following steps:

1. Open a windchill shell on the method server.
2. Enter the following command:

```
java wt.vc.views.ReassignSingleViewCleanser -org Demo -update -sourceview Design  
-targetview Manufacturing
```

3. Enter the user name and password for the site administrator when prompted.

The utility returns information about the parts that were located, indicating whether the views were changed or not changed.

Note: The search is limited to a specific organization context by including the -org parameter.

To change parts with a null view to the Design view, use the following command:

```
java wt.vc.views.ReassignSingleViewCleanser -org Demo -update -targetview Design
```

To change parts with a Design view to the null view, use the following command:

```
java wt.vc.views.ReassignSingleViewCleanser -org Demo -update -sourceview Design
```

Setting Up Windchill Index Search to Index CAD Parts

If you want to be able to index notes created on CAD drawings, you need to modify the DataFormat table. Modifying the table allows for the file types to be marked. Update the .prt extension for UTG Part Data Format to true. This enables all .prt files to be indexed.

Best Practices for Windchill Administrative Tasks

The following sections provide information on best practices for Windchill Administration tasks. The tasks are structured according to when they should be done.

Daily Administrative Tasks

Consider doing the following tasks on a daily basis:

- System monitoring
 - For JVM monitoring, consider using jConsole, PrintGCStats, or HPjmeter.
- Oracle monitoring
 - Consider using Oracle Enterprise Manager, perfmon, or others.
- Do heap, CPU, vaulting, and replication monitoring.
- Network monitoring
 - Ensure sufficient bandwidth and minimal latency.
- Be proactive about capturing client/server logs when errors occur.

Configure the Pro/ENGINEER Wildfire batch file to automatically capture and archive client logs.
- External storage and content administration
 - Check for any external storage vaults which have switched to read-only:
 - Review the method server log on the master site for the cause.
 - The switch could be due to normal operations (size threshold reached).
 - The switch could indicate an error reading to or writing from the vault (much more commonly the issue).
 - Check for disabled vaults or folders.
 - Broadcast configuration to replica sites.

Failures indicate the master and replica sites are not communicating properly.
- Defragment your disks, especially when using virtualized environments.
 - Be sure to defragment both the host and guest OS
 - Run defragmentation utilities often. For example on Windows, use Diskkeeper or similar tool in automatic defragmentation mode.

Be careful to select a tool / procedure that minimizes impact to application disk I/O.

- Monitor log files:
 - Check operating system, hardware, and network event/failure logs.
 - Check method server logs for exception stack traces (trouble-shooting may lead to analysis of other log files). Also check for any unexpected messages; in the log4j log files, check for high priority errors. In traditional log files, look for unexpected messages
 - Monitor the Oracle alert_{instance name}.ora file for ORA-XXXX error messages (if you are using the Oracle database). These messages nearly always indicate an issue that needs to be addressed to avoid massive failures.
 - See [Managing Logging](#)

Weekly Administrative Tasks

- Check queues for failed entries, and investigate the cause. In particular, check the following queues:
 - PurgeScheduleHistoryQueue
This queue removes history records for schedule queues.
 - PurgeUnreferencedFvItemQueue
This queue removes business object to content links which are no longer referenced by business objects.
 - replQueue
This is the master schedule queue for replication.
 - RevaultingQueue
This is the master schedule queue for vaulting.
There is one entry for each scheduled bulk re-vaulting job.
 - wt.router.1.1 (wt.router.1.* if pooled))
This is the master execution queue for vaulting.
There is one entry for each item to be vaulted during a scheduled bulk re-vaulting job.

Additionally, see the [Background Queues](#) chapter for information on queue maintenance and the descriptions of some of the queues.
- Perform Workgroup Manager maintenance:
 - Delete unreferenced files and unused workspaces.

- Keep Workgroup Manager clients are up to date with performance settings and recommendations.
- See TANs/TPIs from the Technical Support knowledge base as well as updated product documentation.
- Check log files for Aphelion, Apache, and Tomcat:
 - Aphelion log files are located in:


```
<Aphelion_Home>/var/ldc/PTCLdap/PTCLdap_logs
```

Back up the ldc.log and ldc.stat log files into a compressed archive each week. Alternatively, logging can be disabled by setting loglevel to 0 in the PTCLdap_ldc.conf file.
 - If you are using the Apache Web server, check Apache error logs and mod_jk logs for error messages related to thread issues, missing or outdated jar files (imagexx_RB.xxx not found), and so on. Optionally, enable access logs and track with awstats to get an idea of concurrency, and user acceptance and utilization rates.
 - If you are using Tomcat, check catalina.out for error messages dealing with thread issues or other problems.
- Check disk utilization on all volumes of the servers.

The usage should be recorded over a period of time. Collating these statistics and extrapolating makes it is easier to predict when additional disk capacity will be required. Relevant statistics include sec per transfer, IO operations per sec (IOPs/sec).
- Check server memory and CPU utilizations (during peak business hours).

The usage should be recorded over a period of time. Collating these statistics and extrapolating makes it is easier to predict when additional computing power will be required. Memory usage should not exceed 80%, CPU should not exceed 70%, and system or privileged time should not exceed 15% for extended periods of time (10 minutes or more).
- Check network saturation (during peak business hours).

Check network performance between the Windchill server and the Oracle server. The usage should be recorded over a period of time. Collating these statistics and extrapolating makes it is easier to identify if any actions are required in order to improve overall performance. Relevant statistics are Packets/sec and Bytes/sec.
- Check for increases in the database size.

The capacity and usage of the database (table space utilization) should be assessed regularly so as to ensure that sufficient capacity is available for expected data growth. The usage should be recorded over a period of time.

Collating these statistics and extrapolating makes it easier to predict when additional disk capacity will be required.

- When there are periods of rapid change in usage and you are using an Oracle database, ensure that you are properly using the Oracle 10g Tuner.

See the *Oracle 10g Tuner for Windchill Solutions Installation and Configuration Guide*. You can access this guide from the Reference Documents link on the PTC Web site. See [Documentation for PTC Products](#).

Monthly or Quarterly Administrative Tasks

- Perform the following external storage and content administration tasks:
 - For each vault (if not using the **Force Content to Vault** option), execute a re-vaulting session with verbosity enabled, and investigate any errors reported.
 - For the master and for each replica site, enable verbosity and execute a replication session, and investigate any errors reported.

Vaulting verbosity settings:

- wt.fv.verbose
- wt.fv.log.enabled

Replication verbosity settings:

- wt.fv.replica.verbose
- wt.fv.replica.log.enabled
- wt.fv.master.verbose
- wt.fv.master.log.enabled
- wt.intersrvcom.verbose

- Test FTP transfer time from server to client, and compare to Windchill download. Windchill will probably be slower due to application overhead, but they should be similar.
- Review the Oracle maintenance schedule and update as appropriate.
- Upgrade to latest Java minor version every six months. Check the latest Windchill Software matrix for the latest information. This matrix is available from the Reference Documents link on the PTC Web site. See [Documentation for PTC Products](#).
- Keep up to date on operating system Services Packs (Windows) or Patch Sets (UNIX versions) every six months.

- Keep up to date with latest version of Apache (including upgrades) every three to six months. Doing this means that you will have the latest security patches and modules in use.
- Keep up to date on Windchill temp patches by checking the Technical Support Web site every six months.
- Retain the log files for Windchill for up to 4 weeks. After this time, compress and copy log files to an appropriate directory.

Following this best practice, ensures successful problem resolution. To aid in this practice, PTC recommends that the log files are created with a timestamp included in the name. This can be done by appending the arguments shown below to the `wt.manager.cmd.MethodServer` and `wt.manager.cmd.BackgroundMethodServer` entries in `wt.properties`:

```
wt.method.log.file=$(wt.logs.dir)/MethodServer.$DATE(yyMMddhhmmss).log
```

```
wt.method.log.file=$(wt.logs.dir)/backgroundMethodServer.$DATE(yyMMddhhmmss).log
```

- Run WinDU diagnostic utilities to ensure your database is clean (and therefore, ready when you perform an upgrade).
- Perform Workgroup Manager maintenance:
 - Reinitialize the client cache.
 - Upgrade Pro/ENGINEER Wildfire (as frequently as is feasible).
Review the compatibility matrix to ensure the desktop and server version combination is certified.
- Verify that your system is sized correctly.
 - Compare current server sizing with the sizing guidelines.
Check the Server Hardware Sizing Guidelines documents that are available from the Reference Documents link on the PTC Web site. See [Documentation for PTC Products](#).
- Check and validate cache size settings (WTPrincipalCache, ACLCache, and so on), as described in the *Windchill Performance Tuning Guide*.
- Review usage metrics:
 - Identify which users, groups, and organizations are using specific areas of functionality.
 - Validate usage and performance test scenarios.

Yearly Administrative Tasks

- At least once a year, export and re-import of the Aphelion directory service to clean it up. For details on completing this task, see the *Windchill Installation and Configuration Guide - Advanced*.
- At least once a year, export and re-import the database to ensure that the tables are defragmented and that the indexes are rebuilt.

Windchill Maintenance 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 ProjectLink, Windchill PDMLink, Info*Engine, and Information Modeler. It also delivers updates to a subset of the Workgroup Managers and gateway products. For these products, there are updates for both English and language specific versions. The specific list of products covered is provided through the documentation accompanying the maintenance release.

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

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

On the test system, you can then do a complete installation to validate that the PTC updates, site modifications, and customizations operate together. After

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 none of the special processing regarding site modifications are required.

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 Pack². In particular this avoids re-installing updates to site modified files when you have previously incorporated those changes.

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

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.

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 accompanying documentation 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
-h, --help	Displays help and exits.
-v, --[no]verbose	Explains what is being done when a command is executed. Default is noverbose.
-w, --wthome=DIR	Sets the Windchill home directory. Default is the parent directory containing the windchill script. Note: 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.
--java=JAVA_EXE	The Java executable. Default is the wt.java.cmd variable value specified in the \$WT_HOME/codebase/wt.properties file.
-cp, --classpath=PATH	Java classpath. Default is the wt.java.classpath variable value specified in the \$WT_HOME/codebase/wt.properties file.
--javaargs=JAVAARGS	Java command line arguments.

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 — generates the help report with no arguments</p>
CLASS [CLASS_ARGS]	<p>Run a Windchill class with optional class arguments.</p> <p>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, enter the following command at the command prompt:

```
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

Bootstrap Client and JAR Files

This chapter provides information about 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.

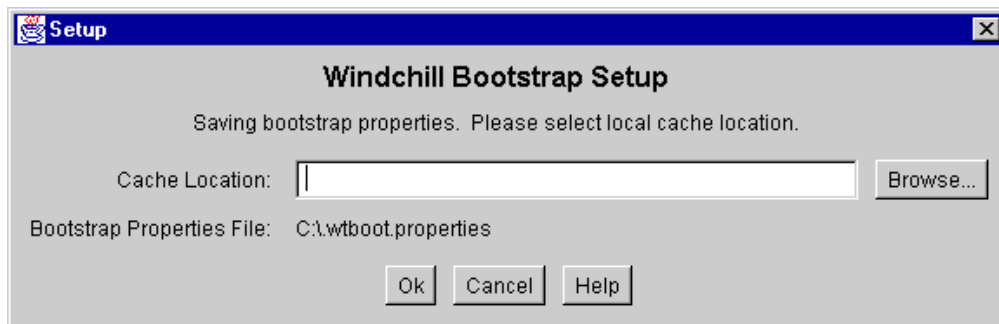
For information about the Java Plug-in, see the *Windchill Info*Engine Installation and Configuration Guide*.

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

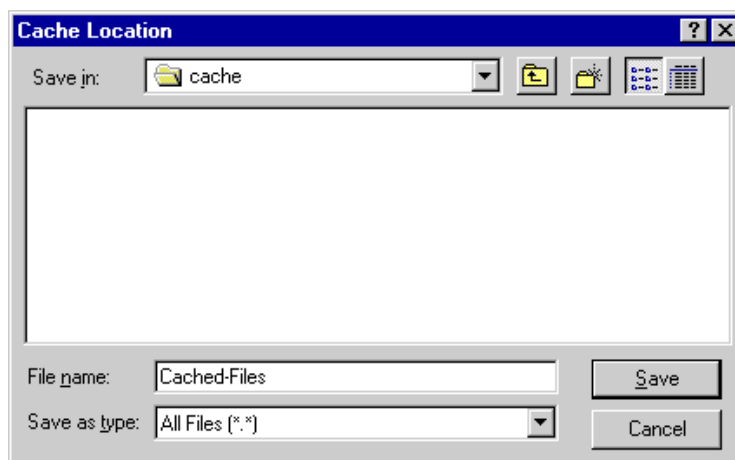
The first time you use the bootstrap feature, you are prompted for the location in which to cache JAR files.



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 that 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, they can be downloaded 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
```

For information on client JAR files, 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

External File Vaults

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, [FvLoader and ReplicaLoader](#).

Topic	Page
About Storing and Moving Data in Windchill	3-2
About External File Vaults	3-2
Central Cache Vault	3-5
Windchill Properties for File Vaulting	3-7
Out-of-the-box Background Queues for External Vaulting	3-11
Windchill External Storage Administrator	3-12
Creating Vaulting Rules	3-18
Managing Revaulting	3-22
Changing the Location of Files in External Vaults	3-28
Migration	3-29
Diagnostic and Repair Tools	3-29

About 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.
- **Replication** -- Windchill 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 replication are of two types:
 - *replica sites* (includes the File Server and full scale sites) to store data for rapid access
 - *master site*, which sends 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 [Replication](#) for detailed information about replication.

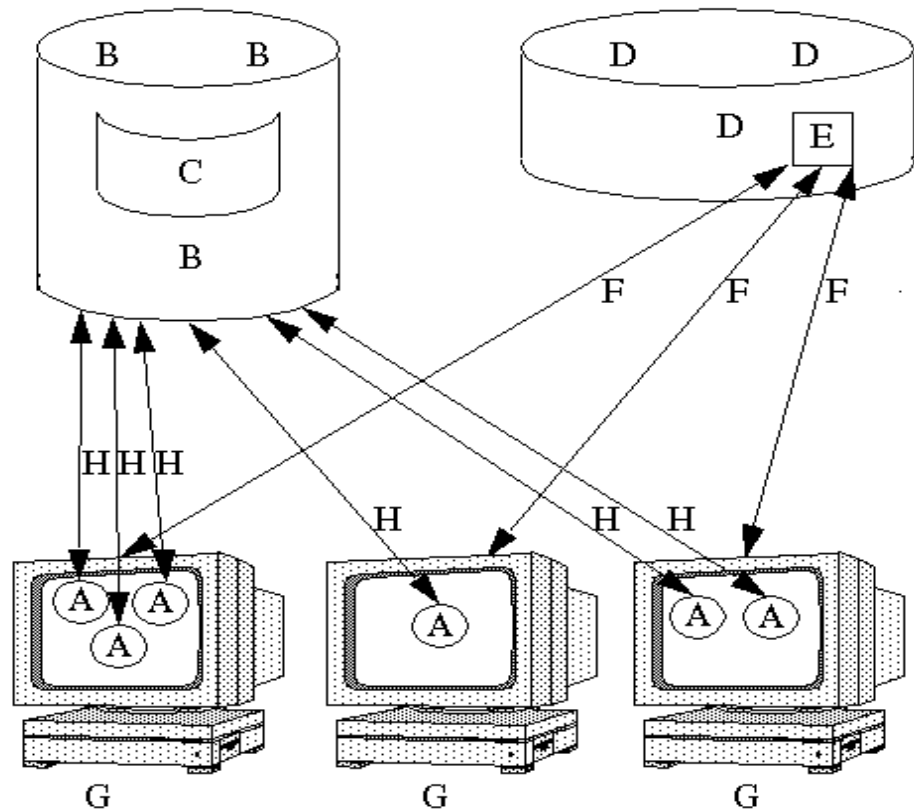
About External File Vaults

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

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

A Windchill *site*, or cluster, is a group of hosts with one URL. The hosts can be accessed individually or as a single unit. File vaults function as elements of sites.

The following graphic summarizes the relationships between the entities that compose a site.



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

- A. Windchill method server -- The Windchill method server that manages the Windchill database processes data and queries that pass between the Windchill database and the external file vaults.
- B. Windchill Database -- The Windchill Database provides an interface for 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 Vault Configuration displays a tree view of the site object, which includes all the hosts, vaults, and folders.

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

- Uploading and downloading content files, which are common Windchill operations, are significantly faster when the files are stored in a vault rather than in the database.
- The storage location of content files is transparent to the user, so no user operations have to be modified.

When uploading very large files (greater than 100 Megabyte) to Windchill, you should not upload to database BLOBs. While it is possible to do so, it can take a significant amount of time (up to several hours for files greater than 0.5 Gigabytes) because of the database speed limit of writing to BLOBs. The upload may seem to hang.

The recommended solution is to configure Windchill so that uploads happen to Windchill file vaults. You can do this one of two ways:

- Use a single vault configuration and set the property `wt.fv.forceContentToVault=true`
- If using multiple vault configuration, then write appropriate rules so that the content is uploaded to appropriate vaults.

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.

Vaults with Automatic Folder Creation

This functionality allows the File Server or System Administrator to create vaults that manage their own folders based on the "number of file" threshold. The system then automatically creates disk folders, as necessary, under a "super" folder called Root Folder.

This decreases the maintenance effort for the Administrators and allows efficient use of the space allocated to the Windchill file vaults.

Root Folders

Root Folders are the "super" folders that automatically manage their own folders.

Note: The first time you register the File Server and no folders exist under the root folder, the first physical folder is automatically created under that root folder.

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 [About Mounts](#).

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 is 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](#).

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. See [Using the xconfmanager Utility](#).

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](#).

A scheduler object created at startup periodically executes to clean up unreferenced database information. Two wt.properties control the timing of the vault purging (default is daily) and the age of unreferenced objects 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. See [Using the xconfmanager Utility](#).

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.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 objects changing their domain and/or life cycle state. The default is true.
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.

Property	Description
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 the content 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	<p>If true, the property wt.fv.fvFileThreshold is effective.</p> <p>If false, wt.fv.fvFileThreshold has no effect.</p>
wt.fv.fvFileThreshold	<p>Value of this property sets the maximum number of files that each folder associated to a vault can hold.</p> <p>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.</p> <p>Refer to Specifying the File Threshold Value for more information.</p>

Property	Description
wt.fv.useVaultsForAll Content	<p>Sets the system to send all content to the vaults.</p> <p>By default, this property is false. If this property is defined as true, all content, irrespective of whether it is domain administered or not is uploaded to a DefaultTargetForSite vault even if there is no vaulting rule.</p> <p>Refer to Sending Content to a Vault for more information.</p>

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 10,000, then the folder becomes read-only at somewhere between 9000 and 11,000 files.

If the file limit is enforced, processing time is 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 is 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, 100,000), it may be beneficial to avoid the performance penalty of automatic enforcement by Windchill, and instead manually monitor folder growth.

Creating a Notification Rule

In the event Windchill is unable to write to a particular folder from any mount point, this folder is automatically set Read-only. This occurs even if the failure is transient in nature, such as an interruption in network access to vaults stored on an NAS or SAN. If all folders for a vault are set read-only, the vault is also set read-only.

It is therefore recommended that administrators create notification rules to alert them to folders and vaults that have become read only. From the Policy Administrator, select the "Site /System" domain and create a Notification Rule for the Administrator user for the OVERFLOW event for the following object types:

wt.fv.FvVault

wt.fv.FvFolder

If the administrator is notified that a vault has been set read-only, they should investigate the cause. If the mount for a folder can be validated, it should be safe to disable the read-only setting for this folder and its corresponding vault.

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 replication to track and clean up information about the old (outdated) schedules for revaulting and content replication. The queue contains only one object 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

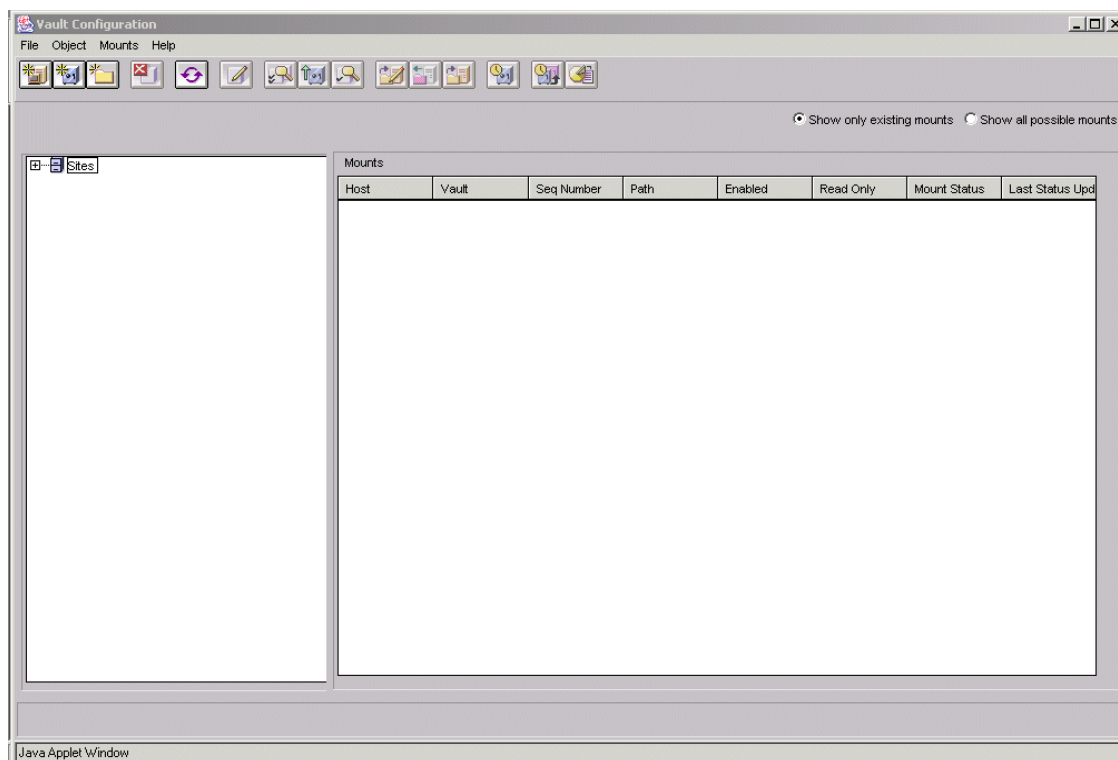
In order to implement file vaulting, you use the **Vault Configuration** window to define the following objects:

- 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.

Note: Site gets created from the **Site Administration** window and not from the **Vault Configuration** window. However, Site does get displayed in the **Vault Configuration** window.

- 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 object, 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, the right panel displays the details about the vaults configured to 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.

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.

- 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, details about mounts for that folder and all the hosts that are configured are displayed. 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 objects.
- Clean up your vaults by removing unreferenced files.
- Validate a single mount or all mounts of an object.

Note: The External Storage Administrator does not verify the validity of the host name you enter.

About Mounts

A mount is the association between a folder or a root 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.

All folders or root 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 or a root folder is mounted to more than one host (for example, in a cluster), all mount paths associated with one folder or root folder must point to the same physical location.

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

- The folder or root folder to be mounted on the specified host system.
- The path to the physical storage location represented by the folder or root folder. If a mount is directed to a nonexistent path, uploads to and downloads from this folder will fail. Each vault folder or root folder **must** have its own unique directory.

When a folder is mounted to a directory, a file with the name starting with "Site_..." is created in the directory. If later, the folder mount is changed, and the you want to use this directory path as a mount for some other folder, you must first delete this file.

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.

Caution: Do not simply unmount the vaults that contain files that are referenced by objects in the Windchill database.

Note: You need to ensure that folders in the vault configuration are validated in order for them to be used. Contents are written only to the folders that are validated and when the corresponding mounts status is VALID.

For example:

If a user creates a folder, mounts it, and does not validate it, content is not written even if the mount path is correct.

In other words, content is written to those vault folders for which all the mounts that have a mount status as VALID.

Updating Root Folder Mounts

When the mount associated with the root folder is updated, the physical directory path, specified as the new mount path, should have the same directory structure below it as the physical directory specified by the old mount path.

For example, consider that the old root folder mount path is the following:

```
D:\folder1
```

with the following subdirectory below it:

```
testvault_RootFolder_1_Folder_1
```

If the user wants to update the mount path of this root folder to the following:

```
D:\folder2
```

then the following sub-directory must exist:

```
testvault_RootFolder_1_Folder_1
```

under the following path:

```
D:\folder2
```

If a root folder mount update is attempted without the required subdirectories created, an error message similar to the following appears:

```
Update of mount for root folder "testvault_RootFolder_1" failed

Update of mount for subfolder
"testvault_RootFolder_1_Folder_1" failed. A physical folder
corresponding to the subfolder mount path
"D:\folder2\testvault_RootFolder_1_Folder_1" does not exist.
```

When you are updating the mount of a root folder, do the following:

1. Mark the root folder to **Read Only** status to prevent additional files from being uploaded to the current mount path.
2. Copy all subdirectories and files recursively from the old mount path of the root folder to the new mount path.

3. Update the root folder mound with the new path specification.

The mounts of the auto folders below the root folder are automatically updated.

4. Update the root folder again the clear the Read Only status.

FolderToRootFolderConverter Tool

This tool replaces existing folders on a master or replica vault with root folders. The root folder uses the mount path of the folder as its mount path. A subfolder is created below the root folder mount path and any files present in the (old, converted) folder are moved to this subfolder.

This tool is run from the windchill shell with the following command:

```
Windchill wt.fv.tools.FolderToRootFolderConverter
```

The username and password for startup are the administrator logon details. You can find a list of possible command line options by running the following command:

```
Windchill wt.fv.tools.FolderToRootFolderConverter -usage
```

To list existing vaults and folders, run the tool with one of the following options:

```
-listVaultsFolders  
-listRemoteVaultsFolders
```

To convert folders to root folders, run the tool with the following option:

```
-vaults
```

For each folder of each vault passed in the following argument, the folder is unmounted and in each folder's place, a new root folder is created with a root mount path corresponding to the original folder's mount path.

Following is a list of valid arguments:

- -user=<adminid> -- User ID of the Administrator user.
- -password=<adminpassword> -- Password of the Administrator user.
- -vaults=<vault1,vault2..> -- Only folders in specified local/remote vaults are converted.
- -listVaultsFolders -- Print vault and folder names and exit.
- -listRemoteVaultsFolders -- Print vault and folder names on remote site and exit.
- -usage -- Print list of valid arguments and exit.
- -debug=<level> Print additional informational/debug messages.

Valid values range from 1 to 3 (1 = least detailed, 3 = most detailed).

Note: PTC recommends that this program be executed on a per vault basis. PTC also recommends that a backup of all disk folders affected should be taken before attempting to run this tool. It needs to be restored in case the tool run ends in an error.

Maintaining Your Vaults

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

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

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

You can use this file to configure your system back-up tool for effective protection of your file vaults. If you are not setting properties through a graphical user interface or in a mapping rules file, you add or edit properties with the xconfmanager utility. See [Using the xconfmanager Utility](#) elsewhere in this guide.

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

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

Creating Vaulting Rules

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 object with which they are associated.

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

You should be aware that vaulting rules created on life cycle state "ALL" do not apply to new life cycle states that are created after the rule creation time. If life cycle states are created after vaulting rules have been created on life cycle state "ALL", those rules do not take into account any objects that are in these new life cycle states. As a result, the objects with these new life cycle states will not be vaulted but instead be stored in the database (as BLOBs).

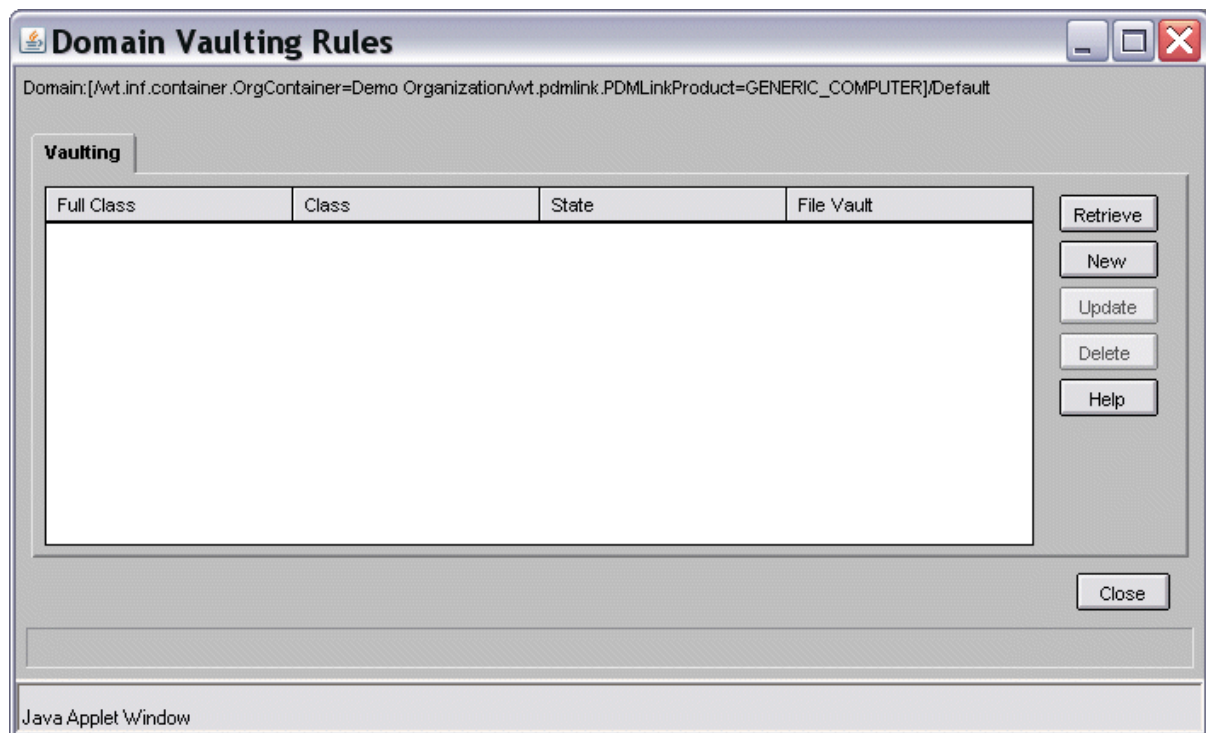
Note: This does not apply if `wt.fv.forceContentToVault=true` is set.

In order to take into account the new life cycle states for vaulting, the rules on life cycle state "ALL" need to be manually created anew in the interface or using the FvLoader. The FvLoader utility can be used to retrieve the list of all the rules, remove and re-create them. See the chapter [FvLoader and ReplicaLoader](#) for detailed information.

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, state, and a file vault. The following example shows the **Domain Vaulting Rules** window for the Parts domain, with the **Vaulting** tab selected.



New Rules

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

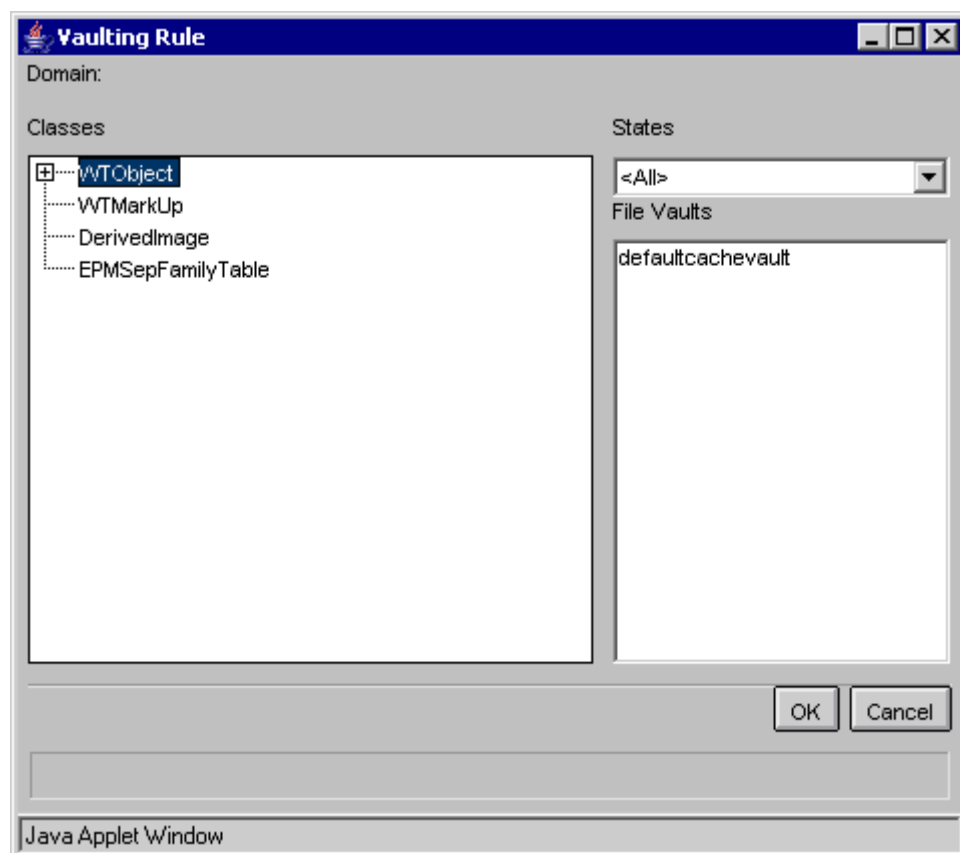
- `wt.viewmarkup.DerivedImage`
- `wt.viewmarkup.WTMarkUp`
- `wt.epm.familytableEPMSepFamily Table`

Note: A vaulting rule can be created on all the objects implementing the following interface that are `DomainAdministered`.

```
wt.content.ContentHolder
```

Also, all the non-`LifeCycleAdministered` are included.

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

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

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

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

Sending Content to a Vault

Vaulting rules are defined to all the ContentHolder classes that are DomainAdministered. ContentHolder classes that are not DomainAdministered cannot have rules.

With the following property, all content that does not have existing rules or cannot have rules can also be vaulted to a special vault called a default target vault:

```
wt.fv.useVaultsForAllContent
```

If this property is set to true, the content always goes to the vault based on the following rules:

- If a vaulting rule is defined for such content, the content goes to the vault based on the rule.
- If a vaulting rule is not defined for such content, the content goes to the default target vault.

Note: The rule may not be present because either a rule cannot be defined for such a ContentHolder class (not domain administered), or a rule can be defined, but has not been defined.

If this property is set to false, the content goes to a vault or a BLOB based on the following conditions:

- If a vaulting rule is defined for such content, the content goes to the vault based on the rule.
- If a vaulting rule is not defined for such content, the content goes to the BLOB.

Note: This property defaults to false.

Managing Revaulting

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

Revaulting is necessary when a vaulting rule is modified to use another file vault or when a vaulting rule is deleted, which is equivalent to designating the object storage to be in a BLOB. Revaulting may also be needed when a change occurs in the domain or life cycle state of an object that holds content files. The revaulting process for such object changes can be done in the background, which is administered by a property, wt.fv.revaultOnChange (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. See [Using the xconfmanager Utility](#) 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

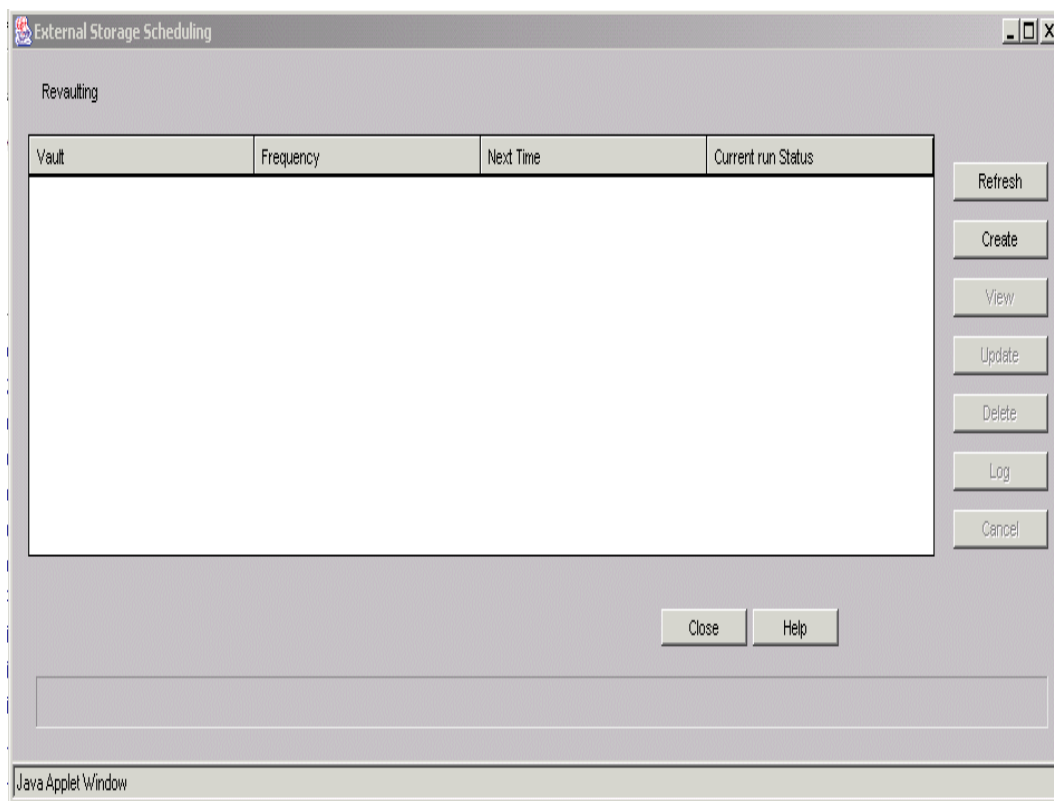
when it is to be done. Creating a schedule object for the vault does this. At the scheduled time, the revaulting process is launched and the contents of the vault are either relocated to another vault, moved from a vault to BLOB storage, or moved from a BLOB to a vault automatically. Use the **Remove Unreferenced Files** menu object 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 Objects

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

The **External Storage Scheduling** window opens.

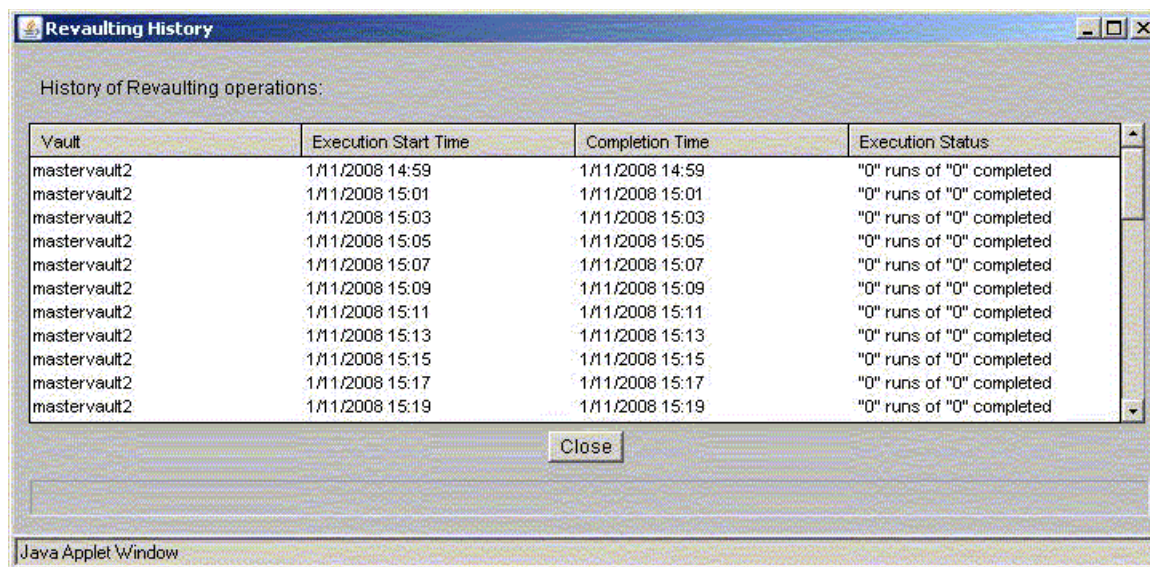


A list of vaults is displayed, for which revaulting has been scheduled. Each vault displays the frequency of revaulting, the time of the next revaulting run, and the status of the current run. A status of **READY** means that the run has been scheduled. Click **Refresh** to update the contents of the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

Viewing the Results of Revaulting

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

The **Revaulting History** window opens.



The screenshot shows a window titled "Revaulting History" with a table of execution logs. The table has four columns: "Vault", "Execution Start Time", "Completion Time", and "Execution Status". All entries in the "Vault" column are "mastervault2". The "Execution Start Time" and "Completion Time" columns show timestamps from 1/11/2008 14:59 to 1/11/2008 15:19. The "Execution Status" column shows "0" runs of "0" completed for each entry. A "Close" button is located below the table. The window is identified as a "Java Applet Window" at the bottom.

Vault	Execution Start Time	Completion Time	Execution Status
mastervault2	1/11/2008 14:59	1/11/2008 14:59	"0" runs of "0" completed
mastervault2	1/11/2008 15:01	1/11/2008 15:01	"0" runs of "0" completed
mastervault2	1/11/2008 15:03	1/11/2008 15:03	"0" runs of "0" completed
mastervault2	1/11/2008 15:05	1/11/2008 15:05	"0" runs of "0" completed
mastervault2	1/11/2008 15:07	1/11/2008 15:07	"0" runs of "0" completed
mastervault2	1/11/2008 15:09	1/11/2008 15:09	"0" runs of "0" completed
mastervault2	1/11/2008 15:11	1/11/2008 15:11	"0" runs of "0" completed
mastervault2	1/11/2008 15:13	1/11/2008 15:13	"0" runs of "0" completed
mastervault2	1/11/2008 15:15	1/11/2008 15:15	"0" runs of "0" completed
mastervault2	1/11/2008 15:17	1/11/2008 15:17	"0" runs of "0" completed
mastervault2	1/11/2008 15:19	1/11/2008 15:19	"0" runs of "0" completed

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.

The logs should display the total number of files and the number of the files revaulted.

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 object 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.

Note: Schedule objects saved to the repository with the **Apply** button cannot be undone.

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. See [Using the xconfmanager Utility](#) 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.

```
wt.fv.forceContentToVault=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.

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.

Revaulting and the `wt.fv.useVaultsForAllContent` Property

The following property is set to true:

```
wt.fv.useVaultsForAllContent
```

When a revaulting is scheduled on a vault and the property is set to true, the following happens:

- All content in this vault, for which the rules have been defined with a different vault as the targets, is moved to its target vaults.
- All content in all other vaults, for which the rules have been defined with this vault as target, is moved to this vault.
- All content in this vault, which have no rules defined, go to the default target vault for the site.
- All content in BLOBs, for which the rules are defined with this vault, is moved to this vault.
- If the vault on which revaulting is scheduled is the default target vault for the site, all content in BLOBs, for which no rules are defined, is moved to this vault.

Note: This only happens once. As with this setting, all uploaded content, for which no rules are defined, is stored in the default target for site. Every site must have a vault that is the default target for the site. If a vault that is a default target for a site already exists and you select a new vault, the new vault becomes the default target for the site rather than the existing one.

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:

Note: If vaulted files are located in auto-created folders, the following steps are different.

1. Log into Windchill as an administrator.
2. Click **System Administration > External Storage Administrator > Revaulting Scheduler**. In the External Storage Scheduling window, either verify that no revaulting is currently underway or cancel all revaulting which is (or is about to be) in progress.
3. Click **External Storage Administrator > Vault Configuration**.
4. In the Vault Configuration window, click on the **Folders** node and select the folder that you want to move.
5. Click **Object > Update**. In the **Update Folder** dialog box, check **Read-Only**, then click **OK**.
6. Copy all the files stored inside /opt/windchill/vaulting/folder-001 to /usr/vaulting/folder-001.
7. Select **Object > Mount** and update the mount location from /opt/windchill/vaulting/folder-001 to /usr/vaulting/folder-001.
8. To test the success of the location change, rename opt/windchill/vaulting/folder-001 and try to access the content of a Windchill object stored in the external vault.
9. In the **Vault Configuration** window, select the folder you have moved and click **Object > Modify**. In the **Update Folder** dialog box, clear the **Read-Only** check box you checked in Step 5.
10. In the **External Storage Scheduling** window, restore any rescheduling operations you canceled in Step 2.

Migration

When the gateway publishes information to Windchill, it transfers the object metadata, not the object files themselves. As part of the gateway mapping process, you specified URL addresses for the source files vaults. When metadata is published to Windchill, the gateway includes this URL value so that a Windchill user can retrieve the object file from within Windchill.

You have the option of configuring your Windchill system to periodically scan the Windchill objects that originated in the source data management system and retrieve the object file from the system using this URL value. While transferring the object files to Windchill requires more network bandwidth than just transferring metadata, this process ensures that the object files are automatically stored within Windchill.

To turn on this content migration feature, you must create a schedule in Windchill that defines when the content is to be transferred. You access this content migration schedule from within the gateway.

See the online help to perform the following tasks for migration:

- Creating a migration schedule
- Modifying a migration schedule
- Deleting a migration schedule

Diagnostic and Repair Tools

This section discusses utilities available for checking and correcting vaulted content files.

Verifying Content Files

Windchill offers a command line utility, WContentVerify, that checks for content files referenced in the database but missing in the vaults, and also verifies actual content file sizes with the sizes in the metadata in the database. Running the `wt.fv.WContentVerify` command from the Windchill shell gives the details of the business object that is associated with missing or incorrectly sized files. The results are output to an XML file.

The command `Windchill wt.fv.WContentVerify -usage` details all the valid arguments and their effects. Running it without the `-email` argument specified (a value of, either `DIRECT_EMAIL` or `EMAIL_GROUP` or both, comma separated) results in no email notifications being sent. All command line arguments are optional, and if no arguments are supplied, the system checks all system content for internal integrity (all vaults and folders are checked for missing or incorrectly sized files).

Note: The `wt.mail.mailhost` property in `wt.properties` file in the codebase directory must point to a valid smtp server.

The arguments and their description are listed in the following table:

Argument	Description
-user=<adminid>	User Id of the Administrator user
-password=<adminpassword>	Password of the Administrator user
-propertyFile=<pathname>	Location of utility's property file
-vaults=<vault1,vault2,...>	Only folders for the specified vaults will be checked. (No spaces allowed.)
-folders=<folder1, folder2,...>	Only specified folders will be checked. (No spaces allowed.)
-replicavaults=<vault1,vault2,...>	Only folders for the specified replica vaults will be checked. (No spaces allowed.)
-replicafolders=<folder1, folder2,...>	Only specified folders will be checked. (No spaces allowed.)
-onlyExistence	Only check and report file existence.
-onlyReportLatest	Report only latest iteration of iterated documents.
-email=[DIRECT_EMAIL,EMAIL_GROUP]	Utility enables mail to specified users. This argument overwrites equivalent properties of the property file.
-listVaultsFolders	Print vault and folder names and exit.
-listRemoteVaultsFolders	Print vault and folder names on remote sites and exit.
-usage	Print list of valid arguments and exit.

Additionally, if WContentVerify throws any exceptions during execution, it is recommended that you run the tool with the -debug option. Running the tool with -debug=1 switches on additional info messages, -debug=2 enables debug messages and -debug=3 adds trace messages as well.

Most of the command line arguments can be specified in a property file and the property file path specified on the command line with the -propertyFile argument. Alternatively, the property file can be saved as WContentVerify.properties in the Windchill codebase directory, the default name and place that this utility searches for.

Note: Arguments specified on the command line override property settings in the property file.

An example property file format is as follows:

```
# Path to the directory that will store utility's output. If not
# specified, will default to $WT_HOME/logs
OUTPUT_STORAGE_PATH=D:\\XML_Output\\

# true/false Enable sending of summary email after a run of the utility
# The wt.properties setting of wt.mail.mailhost is required
EMAIL_GROUP.enabled=true

# Comma separated windchill usernames. Everyone on this list receives
# email notification of a completed utility execution. No spaces allowed.
EMAIL_GROUP.list=testUser1,testUser2

# Enable sending emails to modifiers of the files that have been detected to have
# errors
DIRECT_EMAIL.enabled=true

# Subject of emails sent to modifiers of files that have been detected to have
# errors
DIRECT_EMAIL.mailSubject=Direct Email Report

# Opening line(s) of emails sent to modifiers of files that have been detected to
# have errors
DIRECT_EMAIL.body=First line of Direct Email Report

# Valid windchill username that will be set as the originator of the email
# notification
DIRECT_EMAIL.replyTo=testUser3

# Valid values are html or text. Determines whether the modifiers receive a text
# or html email
DIRECT_EMAIL.format=html

# The maximum number of errors permissible for direct email to be sent.
# If the total number of errors is greater than this number,
# no direct emails will be sent, default is 2000
DIRECT_EMAIL.limit=1500

# Must be one of All or onlyReportLatest . Reports errors either in all
# iterations or the last iteration of iterated objects, default is All.
REPORT_DOCUMENTS_FILTER=All
```

Detecting and Resolving Multiple Primary Content Files

The WMultiPrimaryDetect command line utility detects and corrects a data issue where multiple primary content files have been associated with one FormatContentHolder object. WMultiPrimaryDetect is run from the Windchill shell with the command `windchill wt.fv.tools.WmultiPrimaryDetect`, passing suitable parameters as required. If no parameters are supplied, the tool runs in the diagnosis mode.

Running this tool in the default mode outputs an XML file `multiPrimaryDiagnosis_<YYYYMMDD_HHMM>.xml` listing all the incorrect FormatContentHolders and their multiple primary content items. The tool requires that a Windchill method server instance is up and running.

Running the tool with the `-usage` argument lists all the valid arguments and their effects. The username and password can be supplied with the `-user` and `-password` arguments. If not specified, the application presents a dialog box asking for the username and password. The username provided must be a member of the Administrators group. The following table lists the arguments and their descriptions:

Argument	Description
<code>-user=<adminid></code>	User Id of the Administrator user
<code>-password=<adminpassword></code>	Password of the Administrator user
<code>-fix</code>	<p>Operate in repair mode. Needs <code>inputFile</code> parameter to be specified. If any content files are deleted, a copy of the content file is made in the <code><WT_HOME>/logs/ContentItems</code> directory.</p> <p>Note: The content files in vaults are not deleted, only the entries for them in the database are deleted.</p>

Argument	Description
-inputFile=<input full path and filename>	<p>Use the specified XML file to delete primary content items. The XML file used as an input is the same XML file which has been generated by running the tool in the diagnosis (default) mode. The diagnostic file would need to be updated to have a value of "Y" in the deleteThisItem field for those content items which are to be deleted.</p> <p>Note: The content files in vaults are not deleted, only the entries for them in the database are deleted.</p>
-outputPath=<output pathname>	<p>Use the directory path specified in outputPath parameter to save the XML output files.</p> <p>Note: By default, a file with the name of the timestamp of the log is created within the <WT_HOME>/logs directory and the XML file saved there. If this argument is specified and a path supplied, the specified path will be used instead.</p>
-confirmBeforeDelete=<Y/N>	Ask user for confirmation before deleting each file. (Default is Y.)
-usage	Print list of valid arguments and exit.

To operate in repair mode, the utility requires an input file to specify files to be deleted. The XML file used as an input is the same XML file which has been generated by running the tool in the diagnosis (default) mode, but which has been updated to have a value of "Y" in the **deleteThisItem** field for those content items which are to be deleted.

Note: Modifying the XML file in any other manner may lead to inconsistent results.

Also, note that at least one primary content item per ContentHolder needs to exist. If all of the content items for a ContentHolder have been marked for deletion in the XML file, the tool will print an error message (in an errorMessage field in the XML file) for all content items for that ContentHolder, deleting none of them, and continue processing the rest of the XML file. The errorMessage field is optional and not present for objects successfully deleted.

Upon running the tool in repair mode, a results XML file is generated which is stored in the same directory as the input file supplied. This file has a name of the format multiPrimaryResults_<YYYYMMDD_HHMM>.xml. The format of this XML file is very similar to the multiPrimaryDiagnosis report, except that the **deleteThisItem** field is replaced with an **itemDeleted** field. This field has a value "Y" for content items which have been deleted.

A copy of the deleted content file is saved as the original file name. If there is more than one file with the same name, it is suffixed with _1, _2, and so on. The saved name is also stored in the multiPrimaryResults XML file.

Note: In the WMultiPrimaryDetect tool, logging is controlled by the wt.fv.verbose setting in wt.properties. If the property is set to true, the tool provides verbose output.

4

Replication

Windchill 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*. It also allows you to compose schedules.

Topic	Page
About Replication.....	4-2
Modifying the Local Site.....	4-7
Replication Security	4-7
Editing Replication Properties.....	4-8
Out-of-the-box Background Queues for Replication	4-12
Replication and Windchill Visualization Service	4-14
Replication and Compression.....	4-14
Improving Replication Performance	4-15
Administering a Local Cache Vault	4-18

About Replication

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 (for example, because they are on a WAN) 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 (BLOB) 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, called Windchill File Server, which 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 File Server 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.

Windchill File Server

Windchill File Server is a remote or replica server for Windchill. It was previously referred to as the Windchill Content Cache Server and informally referred to as the lightweight replica server or the lightweight cache server. File Server is installed via the interaction of File Server Manager utility and the PTC Solution Installer (PSI).

The File Server Manager utility is also used for post-installation activities and implementation of File Server. For further information about the installation of File Server, see *Windchill Installation and Configuration Guide - Advanced*.

Specifically, using the File Server Manager utility allows you to synch remote (replica) servers with a master server in a central location. This means dramatic decreases in the time required to install, perform setup, configure, and maintain replica sites. File Server Manager also simplifies that process.

File Server Manager provides the following functions:

- Access to Windchill via a user interface for installing a File Server.
- Ability to install, register, and maintain the File Server.
- Automatic creation of a site, vaults, root folders, and a folder under that root folder where the content resides.
- Automatic creation of additional folders under a root folder if a folder gets full.
- Ability to set a File Server to read-only so that only downloads are allowed.
- Ability to automatically detect an upgrade of the master site and to trigger the upgrade of a File Server.

Note: The autoupgrade is limited to maintenance only releases and patches.

Note: When a File Server is downloaded and installed from a master site which is already at a maintenance (MOR) release (not a base release), the MOR updates are not a part of the downloadable installers. In such a case the registration of the File Server with the master will fail. To correct this, you can install and register a File Server directly from an MOR using the procedure in the section, [Installing and Registering a File Server Directly from an MOR](#).

Installing and Registering a File Server Directly from an MOR

To install and register a File Server directly from an MOR, using the following procedure:

1. Download Installers for File Server.
2. Install the File Server.

The above two steps are detailed in the *Windchill Installation and Configuration Guide - Advanced*. After the installation is done you must perform the following additional steps:

3. Download PTC Service Pack Update (CCSDsu.zip) to the File Server server at <Windchill>/codebase/CCSTools/update folder.
4. The update will be applied when Windchill is restarted on the File Server.
5. If you have problems starting the File Server, refer to the section [Updating a Site Manually](#) to apply CCSDsu.zip manually.
6. Configure/Register the File Server with the Master.

7. After registration, the status for File Server will be "Restart Required". Move CCSDsu.zip from File Server <Windchill>/codebase/CCSTools/update to <Windchill>/codebase/CCSTools/completedUpdate.
8. Restart the File Server so that "Restart Required" status is removed.

Updating a Site Manually

To update a site manually, use the following procedure:

1. Shut down all Windchill related server applications, including Server Manager, Method Server(s), Web server, and servlet engine, and exit all Windchill shells.
2. Open a system console and navigate to the following directory:
<Windchill>/codebase/CCSTools/bin
3. Execute the script file "install_ccdsu.xml" using ANT as follows:
ant -f install_ccdsu.xml
4. Create the <Windchill>/codebase/CCSTools/completedUpdate directory if it does not already exist.
5. Move the contents of the directory <Windchill>/codebase/CCSTools/update to the <Windchill>/codebase/CCSTools/completedUpdate directory overwriting the files to the destination folder.
6. Once the execution completes, start Windchill.

File Server Users

Windchill site, organization, library, and product managers are all able to use File Server Manager. The following limitations apply when they are using the File Server Manager as an Organization or Product/Library/Project Manager:

Note: Security measures called *authentication* are designed to verify the identification of a site, organization, library, or product.

- The File Server Manager can have only one host and one vault.
- Ability to add or remove replication rules for objects within the respective organization, library, product, or project.

Note: Adding or removing replication rules cannot be done through the File Server Manager utility. These tasks are performed through the Replication Administration utility.

- No access to the External Vaulting Configuration dialog or Site Administration Dialog. (This is an alternative way to configure the File Server.)

- Ability to schedule the replication processes for replica vaults managed at the respective level.

Note: Scheduling replication processes cannot be done through the File Server Manager utility. These tasks are performed through the File Server Administrator utility.

For more information about the implementation of File Server, see the online help.

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 objects 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 object for a replica vault. A schedule object describes an operation, for example, "Replicate to vault A at 4:00 PM." Each schedule object is independent, so an object needs to be created for each replica vault. By creating a schedule object, you can ensure that synchronization occurs on a regular basis.

Schedule objects 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 object on the vault in question in the immediate-once mode. Schedule objects 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 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 object. The file vaulting feature of Windchill allows you to specify whether, for a particular type of object in a specific life-cycle state, content files should be stored with the object 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 (File Server 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.

Replica Vaults with Automatic Folder Creation

The ability to create vaults that manage their own folders is also enabled for vaults at a File Server site (also known as a replica vault).

Two Types of Replication Sites

Two types of replication (replica) sites allow you to maximize performance and the use of resources: the File Server replica site and the full-scale replica site.

- File Server replica site -- A partial, dedicated Windchill installation, without database access, which can manage only 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.

The installation and configuration of both File Server and full-scale replica sites are detailed in the *Windchill Installation and Configuration - Advanced*.

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
```

About Security and Using `wt.intersvrcom.URLAuthenticator.useUrlCache`

The property `wt.intersvrcom.URLAuthenticator.useUrlCache`, if set to false (default is true) on the master and all replicas, allows a single URL to be used multiple times within the grace period specified by the property `wt.intersvrcom.security.graceTimePeriod`.

Allowing the multiple use of a URL in combination with a large grace period increases the risk of malicious use of the URL.

Editing 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. See [Using the xconfmanager Utility](#).

The following list describes the properties that are most relevant to 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
wt.fv.master.verbose Properties	X		Print out master service properties on method server start up.
wt.fv.master.verbose	X		Verbose setting for master service. Print out to method server log file.
wt.fv.revaultQuerySize	X		Specify maximum number of objects 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.replicationFileSize Threshold	X		Specify the minimum size of file to be eligible for replication.
wt.fv.master.moveItems BetweenReplicaVaults	X	X	Specify one copy of content on replica site.
wt.fv.master.millisecsTo Wait	X		Specify the duration of the delay before the master site resends replica configuration information.
wt.fv.master.siteConfig DeliveryAttempts	X		Specify the number of times the master site attempts to resend the replica configuration information.

Property	Master	Replica	Description
wt.fv.master.concurrent ReplMax	X		Control the number of parallel replication threads that would be run by Windchill. If this property is not specified, replication by default runs in three parallel threads.

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 [Replication and Windchill Visualization Service](#).

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 objects (previously called items) to the remote sites actually occurs, the following two queries are involved:

- Windchill checks which objects 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 objects need to be replicated from the master to the replica site(s). Only objects that have been replicated are moved. Those objects 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 objects are deleted from the database. default: true
wt.wrmf.transport.httptransport.supportInterruption	X		If true, upload or download resumes the http connection due to IOExceptions such as temporarily unavailable networking problems. This is useful for uploading or downloading large content files. default: true

Property	Sender	Receiver	Explanation
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		Used internally by the wrmf package.
wt.intersvrcom.transport.site.<site name>.useGzip	X		<p><site name>= name of the replica site</p> <p>If set to true, the content is compressed (zipped) while replicating to or pulling from the site replica server.</p> <p>If set to false, it is not compressed.</p> <p>If this property is not specified in wt.properties, the content is compressed by default.</p>

Out-of-the-box Background Queues for 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 object 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. For more information, see [Using the xconfmanager Utility](#).

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

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. For more information, see [Using the xconfmanager Utility](#).

Configuring Properties

To enable content replication for viewables it is recommended that the following property 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. For more information, see [Using the xconfmanager Utility](#).

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.

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 are replicated. See the online help for more details about setting up the rules.

Replication and Compression

By default, replication uses compression. To stop the compression, set the following property in the wt.properties file.

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

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.

Improving Replication Performance

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

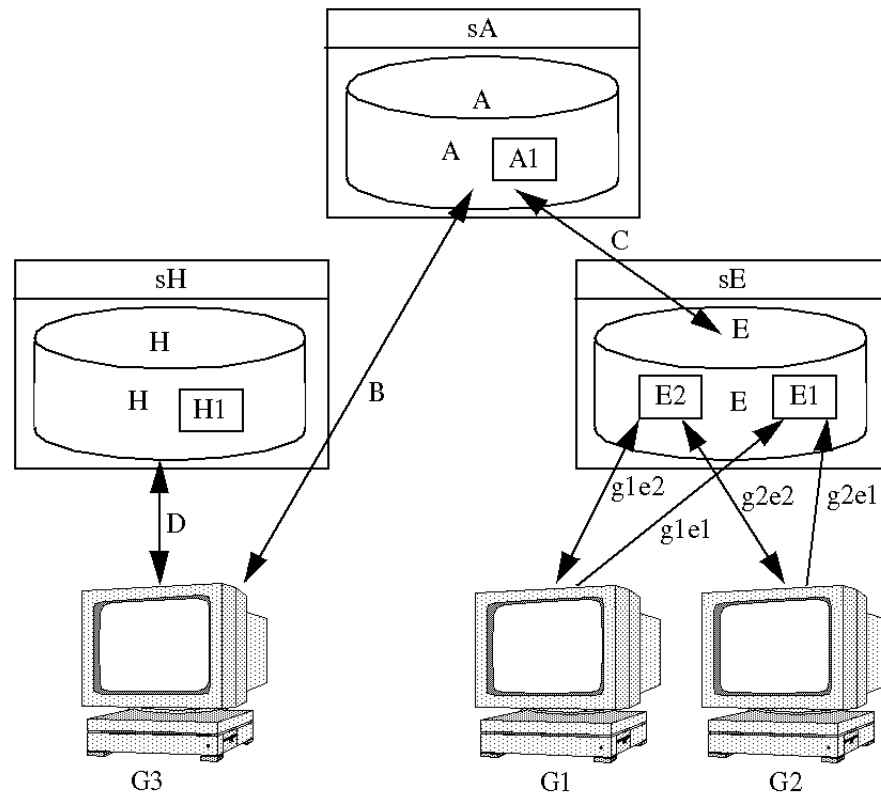
- **Local Upload** -- Places a user's uploaded content in the local cache vault as an intermediate location prior to transfer to a master site upon checkin.
- **File Server**-- Creates a location for rapid access to frequently requested content data.

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 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. For more information, see [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. For more information, see [Establishing Mirroring in the Local Cache Vault](#).

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 **Designated for Cache** 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 object. For example, a file is no longer referenced when all objects to which it was associated have been deleted, or when a user deletes the association to the file when updating the object.

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, and so on.), 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 **File ServerAdministrator**.
 - 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 existing ReplicaCacheVault as read-only.
 - a. In the **Vault Configuration** window, select an existing folder associated to the ReplicaCacheVault.
 - b. Select the **Read Only** check box.
 - c. Repeat steps a and b until all folders have been marked as read-only.
 - d. 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 object.

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 that 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 that has replication rules defined for it is replicated to the site at the next run of content replication. The replica site may still contain content that 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 Objects

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 objects for the target vault and schedule the replication again.

Selecting Reset Replication Undelivered Objects 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, the following option may be used.

To reset replication for undelivered objects

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

Predictive Replication

Predictive replication is a feature that enables Windchill to preemptively determine the content that is required at a said File Server instance based on patterns of user behavior. Predictive replication replicates future iterations of an object without the user/administrator having to set an explicit replication schedule.

The conditions that trigger the setup of a predictive replication are the following:

- The user has the site preference set to a File Server. However, the content requested by the user is not available at this site as a result of a prior scheduled replication. In this case, the content is dynamically replicated and a predictive replication schedule is set.
- The user uploads the content to the preferred site.

Creating New Rules

The system creates a special rule for predictive replication and then replicates the objects qualifying this special rule. These rules are hidden from the user.

After the creation of this special rule, a recurring replication schedule gets created (only if there is no existing schedule for that replica vault). This schedule runs daily at a specific time taken from the wt. properties file. The scheduleType of this schedule is named the following:

PREDICTVE

This schedule performs replication only for objects qualifying for the special predictive replication rule or scheduleType, such as the following:

PREDICTVE+SCHEDULED

This schedule performs replication both for a predictive replication rule as well as the normal rule.

Note: Only one such predictive replication schedule per vault can exist.

Before this special rule/schedule is created, the following checks are done:

- Whether a schedule for that vault exists or not
- If a schedule does exist, then whether a rule exists for the object class and the vault
- If both a schedule and a replication rule already exist, then no predictive replication rule/schedule gets created.

All the objects, as a part of predictive replication, get replicated to a vault that is selected as designated for the cache.

Note: This basically depends on the value set in the adhoc replication property.

Cleaning Up Expired Rules

When the method server starts up, the expired predictive replication rules gets cleaned up. A schedule gets created for such cleanup with the default periodicity as 1 day so that the cleanup happens every day.

The expiration of predictive replication rules is calculated based on the following property:

wt.fv.predictiveReplicationDuration

This property determines the duration for which the predictive replication rules run. The unit of measure is days. The default value is 90 days.

The expiration date of a predictive replication rule is calculated on the basis of this duration. For example:

If the creation date of the rule is earlier than *<current date>* - 90 days, the rule is treated as expired and gets deleted.

Preferences and Options

Following are the preferences and options:

Preference or Option	Description
wt.fv.predictiveReplication Duration	The duration for which the replication rule that runs predictive replication rule runs. The unit of measure is days. This is an Organization-level preference. Possible value is <i><number></i> . The default is 90.
wt.fv.predictiveReplication Enabled	Determines whether to turn predictive replication on or off. Turning predictive replication on would start predictively replicating future iterations of only that content, which is accessed after the property has been turned on.
wt.fv.predictiveReplicationAll Versions	Determines whether to replicate all versions or only the current version. Possible values are true or false. The default is true.
wt.fv.predictiveReplication Immediate	Determines whether to perform the predictive replication immediately or scheduled at some other time. Possible values are true or false. The default is true.
wt.fv.predictiveReplication Periodicity	Determines the how often (periodicity) of replication should occur Possible value is the number of days, weeks, or months the replication should take place.
wt.fv.predictiveReplication ScheduleTime	Contains the time for scheduling Possible value is a time span.

Ad Hoc Replication

Ad hoc replication means the copying of content from a master to a Windchill File Server if the content is not available on a replica and the user has set preferences to a Windchill File Server.

If a user attempts to download data that does not exist on the local File Server, the content is delivered to the user from the nearest proximity site having the data and is also replicated to the local File Server. The next time another local user downloads the same document, it is delivered directly from the local File Server.

This also results in the creation of predictive replication rules and schedules.

The administrator can control ad hoc replication by setting the property `wt.fv.master.adhocCaching.flag` in `wt.properties`. This property can have one of the following values:

- 0 = ad hoc replication is disabled.
- 1 = ad hoc replication is enabled only if a rule exists for content on a user's preferred File Server.
- 2 = ad hoc replication is enabled for all conditions (default value).

The property, `wt.fv.master.adhocCaching.ThreadPoolSize` sets the number of concurrent threads that are available for ad hoc replication. The default value for this property is 3.

Site Proximity

Users can determine a preference in order of sites from the closest to the furthest of a given site when creating a new site or by editing a site. By default, the master site is added to the end of the list, if it has not already been placed in another order.

Proximity is used to find out a suitable site from where the content for ad hoc replication should originate in order to fulfill user requests in the minimum possible amount of time.

User Initiated Replication

This feature allows users to collect and replicate objects that cannot be easily bound together by replication rules. The replication is executed immediately. This action is available from the **Search Results** table.

Note: For the replicate icon to appear and for the function to be active, the user must be a member of a Replication Managers group. For a procedure to add a member to the Replication Managers Group, see the online help.

When the users click the replicate icon, the **User Initiated Replication** window appears. They can perform two tasks in this window.

- Select target File Server sites.

A list of all the File Server sites appear. For each site for which you want to initiate an adhoc replication, add that site. You can add multiple sites.

- Replicate the following dependents for CAD documents.
 - All dependents
 - Required dependents
 - None

Note: This action applies only to the CAD document objects that were selected. For each target CAD document, dependents use the "as stored" config specification if it exists. If it does not exist, the latest config specification is used.

Once the user completes the information and clicks **OK** in the **User Initiated Replication** window, the following process occurs:

1. The replication job is started in the background.
2. The results (success/failure/errors) of the job are reported in the **Event Console** window.
3. A predictive replication rule and schedule are created to take care of future iterations of the replicated object.
4. All the objects as part of the User Initiated Replication get replicated to a vault, which is selected as default target for site.

Note: Every site must have a vault that is the default target for the site. If a vault that is a default target for a site already exists and you select a new vault, the new vault becomes the default target for the site rather than the existing one.

Parallel Replication

The following information defines the parallel replication concepts, describes the parallel replication process, and describes the properties that set the parameters for the process.

About Parallel Replication

A mechanism exists that allows Windchill to accomplish parallel replication or run replication sessions for several target replica vaults situated on any of the sites at the same time. Replication sessions move content in two directions:

- Pushed from the master site to the replica site

Content that resides on the master site needs to be copied to a target replica vault, as determined by existing replication policy rules. During the execution

of a replication session, such content is **pushed** from the master site to the replica site, and stored in the space allocated to the target replica vault.

- Pulled to the master site from the replica site

Content that had been uploaded to a replica site uploads to the content cache site preference you have set. This content needs to be copied to the vaults on the master site for permanent storage. During the content replication session, such content is **pulled** to the master site from the replica site.

Replication Process

The execution of a replication session proceeds as follows:

Note: Each replication session executes in two phases: active phase and passive phase.

Active Phase

1. References to the content that are to be **pulled** are gathered and placed into special pull queue(s), which start processing the content immediately.
2. References to the content that are to be **pushed** (content files) are gathered.
3. The content files are sent to the destination replica site one by one.
4. Once all the content files are pushed, the phase completes.
5. One of the following actions now occurs:
 - If all the content has been pulled by the pull queues already, the session completes.
 - Otherwise, the session enters the passive phase.

Passive Phase

- The session waits for all content to be pulled from the replica sites and then completes.

Note: Once the session enters the passive phase, its run slot is available for any waiting sessions.

Property Controls

Running multiple active sessions of replication concurrently, as well as spreading the pull work among multiple queues executing in parallel, can be achieved by the use of the following properties:

Note: By default, up to 3 active replication sessions can be run concurrently and the pull is handled by a single queue.

- `wt.fv.master.Replication.NumOfQueues=<3>`

This property defines the number of queues that are used to simultaneously execute replication sessions. Default settings allow for up to three concurrent sessions.

Use caution when increasing the maximum number of concurrent sessions, as each session is computationally intensive and uses the master sites network bandwidth for content transfers.

As more replication sessions are scheduled, they get queued in the parallel queues so that at any time, active sessions are equal to or less than the number of queues.

Note: The queues can be configured to run on a specific method server. This feature provides for another dimension to the scaling of replication activity.

- `wt.fv.CCSUpload.NumOfQueues=<1>`

This property specifies the number of processing queues to share that are created and used for pulling the content.

The queue(s) dedicated to pulling content from the replica sites to the master site are shared among all the replication sessions. Each queue pulls one content file at a time. The default for the number of queues is one. Increasing this value increases the number of the processing queues dedicated to pulling the content.

Deleting Registered File Servers

A registered File Server cannot be deleted under normal circumstances. A registered File Server can be deleted only if all the cached objects in the cache vault of the File Server have been pulled back to the master site by doing a replication run.

In addition, if there are any failed or partial uploads where the files are uploaded to the cache vault, the File Server cannot be deleted. The system deletes references to such files only after one month during the cleanup process.

Updating File Servers

Once a File Server is installed, the master site is often updated with maintenance releases or patches and an update automatically occurs.

Preparation

Following are the preparation steps on the master site:

1. The respective installer packages the files and properties into special cumulative files in a dedicated directory (WT_HOME\CCSTools\update).
2. The files accumulate, containing all files and properties for the cumulative maintenance releases and cumulative patches that need to be propagated to the File Server.

Note: The following property must be set in wt.property on the master server:

```
wt.intersvrcom.autoManageCCS=true
```

Propagation

Once the Site Manager on the master site has prepared the information and files, performed an update on the site, and restarted that site, the following automatic process begins:

1. The master system notifies the File Server of the need to upgrade.
2. The site is set to **readOnly** status.
3. The master system sets the status modifier on the File Server to **Update in Progress** status.
4. The File Servers recognize the need to update itself.
5. The master server pushes the update to the File Server.
6. The status modifier changes to **Restart Required** status.
7. The system sends notifications to appropriate managers asking them to perform the restart.
8. Restart is performed.
9. File Server updates itself during the restart.
10. The master system does the following:
 - Verifies the File Server release level
 - Removes the **Restart Required** status modifier from the File Server
 - Removes the **readOnly** status from the site.

Note: A Java utility called CCSLauncher can be used to start, restart, and stop the File Server. This utility can update the File Server database and xonfmanager properties from the update files and then start the File Server.

Upgrade Considerations

You need to set the container reference of the Site container as follows:

- Site object - To all pre-upgrade sites, set the read-only flag to false and the enable flag to true.
- ReplicaVault object - to all objects, set the flags to what they were before the upgrade.

Enhancing the File Server User Experience

The following functions help with the user experience when the File Servers are configured and in use:

- Site Monitor

This is a daemon thread which, at Windchill defined intervals, pings all replica sites and holds the ping status in the Master. The following property specifies the interval between each replica ping:

```
wt.intersvrcom.sitePingIntervalInMinutes
```

The monitor is present on every method server on the system. The result of each ping is stored in the cache. Method servers check the ping results before each attempt to ping the replica sites and if the existing pings are recent enough, they skip the next ping.

- Automatic Mount Validation

This feature, similar to Site Monitor, for validating mounts by folder, file, or sit allows mount validation in clustered environments and on replica sites.

Mount validation can also be initiated through the Vaulting Configuration user interface.

Following is the process flow for mount validation.

1. The System Administrator selects an object, such as site, vault, or folder from the tree in the left pane of the **Vaulting Configuration** window and clicks **Validate**.

Note: If there is at least one mount that is yet to be created, a warning message appears.

2. The system sends all of the data needed for the mount validation to the respective replica site.

3. The system on the replica site validates all the existing mounts related to the selected object.
 4. The system sends the validation status to the master site.
 5. The system on the master site updates the status of all the mounts related to the selected object to **Valid** or **Does not exist**.
 6. The system updates the Last Validated timestamp of all the mounts related to the selected object.
- Ability to quarantine corrupted files

Any content that Windchill deems corrupted is marked as quarantined. Quarantined content is content that is found to be defective in some way by Windchill. Such content cannot participate in revaulting or replication until it is removed from quarantine.

For the quarantined objects, the folder must be accessible; it is not necessary for the content to be accessible. This helps to avoid quarantining files for the dismounted folders. The revaulting and replication functionality filters out the objects, which are quarantined.

The following dedicated property collects data on currently quarantined files and remove a subset of these files from the quarantine.

```
wt.ContentQuarantinedContentManager
```

QuarantinedContentManager [-reportAllQuarantined] [file]

This tool generates reports for all quarantined contents in the system and stores it in the specified file.

QuarantinedContentManager [-UnQuarantineAll]

This tool removes any quarantine flag from all contents that are currently in quarantine.

A separate log is also created, which receives a record for each quarantined file with the details of why this object is being quarantined. Windchill administrators can also sign up for email notification of quarantined content.

The following other property exists:

```
wt.content.QuarantineNotifyIntervalHours
```

This property specifies the interval, in hours, during which an Administrator, who subscribes to the JMX quarantine notification and receives, at most, one notification of content quarantine events.

The default is 24 hours. As such, the notification on quarantined event is sent out, at most, once in a 24 hour period.

If the property value is set to 0, notification is sent out for each quarantined event.

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
```


5

FvLoader and ReplicaLoader

FvLoader and ReplicaLoader are utilities that help you perform a number of tasks for both the master and replica sites.

Topic	Page
About FvLoader and ReplicaLoader	5-2
Configuring External File Vaults or Rules	5-4
Supporting Replica Site Vaulting.....	5-5
Removing Rules	5-7
Listing Domains	5-7
Listing Rules.....	5-9

About FvLoader and ReplicaLoader

PTC supplies two classes of utilities that you can use for both the master and replica sites. The utilities are the following:

- FvLoader -- (master site) A shortened version of File Vault Object Loader.
- ReplicaLoader -- (replica site)

For both utilities, some actions are controlled by command line arguments, and other actions are controlled by data in files.

Tasks

FvLoader and ReplicaLoader are utilities that help you perform a number of tasks for both the master and replica sites.

FVLoader

FvLoader can be used for the following tasks:

- To create a file vault
- To create a file vault host
- To create a file vault mount
- To create a file vaulting rule
- To enable a file vault
- To enable a file folder
- Remove a file vaulting rule

See [Configuring External File Vaults or Rules](#) and [Removing Rules](#). You may need to use FvLoader to list domains to perform the removal actions efficiently.

ReplicaLoader

Replica Loader can be used for the following tasks:

- To create a replica file vault
- To create a replica file vault host
- To create a replica folder
- To create a replica file vault mount
- To create a replication rule
- To enable a replica vault

- To enable a replica folder
- To remove a replication rule

See [Supporting Replica Site Vaulting](#) and [Removing Rules](#).

Command Line

You can use the command line to do the following:

- To list domains -- a two-step process. See [Listing Domains](#).
- To list vaulting policy rules -- use the output for batch deletion or recreation of policy rules. See [Listing Rules](#).
- To support local replication -- provides accelerated content replication through a series of steps. Users can invoke ReplicaLoader 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.

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

Commands

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

FvLoader

- 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 enter in the command window is the following:

```
java wt.fv.FvLoader
```

- Create your own comma-separated value (.csv) file. The command that you enter 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.

ReplicaLoader

- Modify fvloader.csv file in the directory /loadFiles directory to specify the task.

Note: The same loader file gets used for the ReplicaLoader.

Running ReplicaLoader with no arguments loads data from this file. The command that you enter in the command window is the following:

```
java wt.dataops.replication.ReplicaLoader
```

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

```
java wt.dataops.replication.ReplicaLoader <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.
- V,vaultName,autoFolderCreate -- Creates external file vault with the name vaultname and the flag autoCreate

where

autoCreate = true, which means that the vault is designated for automatic folder creation.

If the vault created is an autoFolderCreate vault, then the Create Folder command (such as F,folderName,vaultName) creates a RootFolder instead of a folder. All the folders automatically get created under this root folder.

- H,hostName -- Creates a file vault host with the host name hostName.
- F,folderName,vaultName -- Creates a 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)

If the `folderName` is the name of root folder, then a mount for that root folder gets created under which folders are automatically created as and when required.

- `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 [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`.
- `RV,vaultName,replicaSiteName,autoFolderCreate` -- Creates replica file vault with the name `vaultName` and attaches it to the replica site with `replicaSiteName` with the flag `autoCreate`

where

`autoCreate = true`, which means that the vault is designated for automatic folder creation.

If the vault created is an `autoFolderCreate` vault, then the Create Replica File Vault Folder command (such a `RF,folderName,vaultName`) creates a `RootFolder` instead of a folder. All the folders automatically get created under this root folder.

- `RV,vaultName,replicaSiteName,autoFolderCreate,defaultTargetForSite` -- Creates a replica file vault with the name `vaultName` and attaches it to the

replica site with replicaSiteName with autoCreate flag and default target for the site flag

where

defaultTargetForSite = true, which means that the replicavault is the default target for the site and is used for system replication.

- 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

If the folderName is the name of root folder, then a mount for that root folder gets created under which folders are automatically created as and when required.

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 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, see [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 namevaultName.
 - 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, see [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 entering 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 subtopics.

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  
  
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 to 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 Rules

Use the following arguments to print a list of existing vaulting policy rules for given site to the console:

```
java wt.fv.FvLoader-listFvPolicyRules <site name>
```

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.

Use the following arguments to print a list of vaulting policy rules for replica sites to the console:

```
java wt.dataops.replication.ReplicaLoader-listReplicationRules <site name>
```

See the prefixes R, RR, RemoveLocalR, and RemoveReplicaR in the preceding discussions.

Note: Note that output and the input for the following prefixes is almost identical.

```

-listFvPolicyRules
-listReplicationRules

```

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>
```

where

<master> = the site name

```
###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.dataops.replication.ReplicaLoader -listReplicationRules <replica_11>
```

where

<replica_11> = the site name

The output will be of the following form:-

```
[Rule Name]==>RuleName
```

```
[TargetVault]==>replica vault 1, [ObjectType]==> Document,
```

```
[LifeCycleState]==>ACCEPTED, [Context]==>wt.pdmlink.PDMLinkProduct:10196;ALL
```

```
C:\> java wt.dataops.replication.ReplicaLoader -listReplicationRules replica99
```

The following command lists rules for only the master site. It does not list rules for the replica site.

```
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]
```

```
replica_vault_1,wt.doc.WTDocument,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
```

```
replica_vault_1,wt.part.WTPart,[/wt.inf.container.OrgContainer=PTC]/Default/Project,ALL
```

```
C:\>
```

In the output, each line has the following prefixes which specifies the type of the rule:

- LocalPolicyRule -- Rule is used for external vaulting.

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,v1,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.

Note: By default, the fvLoader.csv contains a comma separated value (csv). However, the separator can be configured. The follow property can be used to configure the separator:

```
wt.fv.FVLoaderSeparator
```

The default value for this property is a comma (,). The user can set any separator to this property and then the same can be used in fvLoader.csv.

The same property applies to the ReplicaLoader as well.

6

Windchill Export and Import

Windchill Export and Import (not available in Arbortext Content Manager) can assist you in moving content and metadata to and from sites by placing the data in JAR files. The JAR file contains XML files representing the object's metadata and content files.

Topic	Page
About Export and Import	6-2
Context Considerations	6-4
Supported Objects List	6-8
Configuration Specification Settings.....	6-26
EPMDocuments.....	6-27
Export User Interface	6-28
Import User Interface	6-34
Additional Export and Import Actions.....	6-38
Properties.....	6-39
Windchill Export Properties.....	6-40
Windchill Import Properties.....	6-41
Access Control	6-42

About Export and Import

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

- Objects in Cabinets and Folders -- The content of objects supported by Windchill Export that is located in cabinets and folders is compressed. Folders and cabinets are not supported, but for each supported object, data is included in the JAR file about the cabinets and folders that held it.
- Product Structure (built with active configuration specification) -- A WTPart serves as the seed object for a complete product structure, which includes its dependent child WTParts and associated WTDocuments, built with the active configuration specification.

The following can serve as seed objects:

- Subclasses of WTParts
- Serial numbered parts
- Instances of soft typed WTParts
- End objects, which are also known as WTProducts.
- CAD Document Structure (built with latest configuration specification) -- A CAD Document serves as the seed object for a complete CAD structure generated with the latest configuration specification. No WTParts or links between WTParts and CAD Documents are exported.
- Product Structure with CAD Documents (built with active configuration specification) -- A product structure of WTParts with CAD Documents combines the two preceding options. It supports exporting the product structure of the WTParts and CAD Structure of the CAD Documents along with the build rules between the two structures. This leads to the export of WTParts, product structure, CAD Documents, CAD structure, and content files.
- Document -- Ordinary or Soft Type instances of Documents can be included in the JAR file.
- Soft Type Definition -- WTDocuments and WTParts can have soft type definitions. You cannot check out type definitions on export

The JAR suffix is automatically appended unless you specify another suffix. You can filter objects by their time of last modification to control what is included in a JAR file. Any software that expands ordinary zip files can also expand the JAR files produced by Windchill Export.

Windchill Import does not import an object 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 objects, such as WTPart, WTDocument, and EPMDocument instances, the unique identifier is known as the UFID (unique federated identifier) that is composed of the following:

- Local ID
- Domain
- Site

Different object types can have different uniqueness identifiers.

For example:

An instance based attribute (IBA) or soft type definition object can be identified by its name and its path.

An object's business identity is derived from the value of certain attributes, which are as follows:

- For a WTPart -- Number, Version, Iteration, and View
- For a WTDocument -- Number, Version, and Iteration
- For a CAD document -- CADName, Number, Version, and Iteration

If an object to be imported has the same UFID, but a different business identity than a database object, the import fails 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 object.

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 object context during import or export.

The way objects in the database can be created or modified during import and export operations is governed by the use of policy files or selected actions available in the 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.

Windchill Export and Import does not support the export and import of JAR files across releases. In other words, an earlier version of a JAR file cannot be imported into a new version of Windchill, unless explicitly enabled.

If a cross release import is necessary, it can be performed by enabling an Export Import property.

Note: This should be done only by user discretion. Any data inconsistency or integrity issues arising after a successful import would not be supported.

Specify the following property in the wt.properties and restart the method server in order for it to properly reload:

```
wt.ixb.import.allowCrossReleaseImport=true
```

Alternatively, for dynamic reloading of this property, specify it in a Export Import mapping rule file with the following format:.

```
<?xml version="1.0" encoding="UTF-8">
<userSettings>
<properties>
wt.ixb.import.allowCrossReleaseImport=true
</properties>
</userSettings>
```

Note: This restriction currently has been applied to import operations being performed through Export/Import Manager in Windchill.

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

The software manages objects within logical entities called containers. The container concept is used to separate objects that belong to different working contexts.

This topic discusses the following in its subsections:

- Access to objects and the export or import of objects at the appropriate context level
- Controlling context in import operations

Export Container Availability

This topic explains export container availability for product, library, project, organization, and site.

Product, Library, or Project Level Container Availability

At the product/library/project level, the export action is available to end users (with read access) and administrators of the context itself, its parent context (the organization), or a Site Administrator.

The export action is available on the property (details) page of any of the following types of objects:

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

Any IBA values and IBA definitions must be exported with the object instances. The associated type definitions, either soft types or hard types, must be exported as well.

- All soft type and hard type definitions. In most cases these are instances of modeled subclasses of WTPart, WTDocument, and End Item which is also known as WTProduct.
- Supported objects in a folder.

The export must include the definition of the modeled subclass as if it were a soft-type.

You can export EPMDocuments through the **Export/Import Manager**.

Additionally, the **Export** window is 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 objects to export. The search returns objects created within the context container. To export data from a project context, the user should use the export action from the **Project Details** page.

Note: In order to export an object, you must have READ access to the object and all its attributes. For example, you must have READ access to the Team of an object, 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 objects to the organization or site level.

Site Level Container Availability

On the **Utilities** tab, the export action is available to administrators of the site container.

Import Container Availability

This section explains import container availability for product, library, project, organization, and site.

Product, Library, or Project Level Container Availability

At the product/library/project level, the import action is available to the context or parent context (organization or site) administrators. The imported objects are 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 objects may be imported into this level.
- When type instances are being imported, if the type equivalent is not found in the destination system, the type is created in the target container's organization container, provided the organization container has a properly configured internet domain; otherwise the type is created at the site container. The user is assumed to have the permission to create types at the appropriate container levels; otherwise the import fails. See the following section for [Type Definition Equivalence](#).
- Do not use the **Export/Import Manager** to import data into a project. Instead, use the Import from File action available from the **Project Details** page or the folder Import from File action. There is a slight difference in how these two actions work. The Import from File action from the **Project Details** page provides provide the default behavior for the folder structure of all the objects in the target set. When the Import from File into folder action is performed, all foldered objects are placed into the target folder.

Type Definition Equivalence

For import purposes, a type definition in an import file is considered equivalent to a local type definition if all of the following criteria are met:

- It has the same name or the name is mapped to a local name, as well as mapped to the same parent type, unless this type is a root-level type. The names of hard types cannot be changed.
- It has the same values for the following attributes: instantiable, userAttributeable, and 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 and have the same value.
- The two types have the same set of constraints on their attributes as well as the same set of constraints on the soft type itself.

If a type matching the above criteria is found in the system, it must be visible to the context into which the import is being performed.

Organization Level Container Availability

The import action is available to the Organization or Site Administrator. Folder contents can be imported into an organization context. Type definitions can be mapped to locally defined type definitions.

Site Level Container Availability

The import action is available to the Site Administrator. Folder contents can be imported into a site context. Type definitions can be mapped to locally defined type definitions.

Controlling the Destinations of Imported Objects with Context Mapping Files

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

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 object
      at the export site</source-container>

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

  <container>
```

```

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

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

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

        <target-container>containerReference of the context where
        the object 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>

```

Supported Objects List

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

The following tables are included:

- [Supported Windchill ProjectLink Objects](#)
- [Supported Windchill PDMLink Objects](#)
- [Configuration Specification Settings](#)

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

Supported Windchill ProjectLink Objects

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

Windchill ProjectLink Object	Supported by a Loader?	Can Be Exported?	Comments
Folder			
Folder structure	Yes	Yes	
Folder access rules	Yes	Yes	
Links	Yes	Yes	Parts and documents supported only
Imported attachments	No	No	
Document			
Document metadata	Yes	Yes	
Template document	No	No	Exported as WTDocument
Document content	Yes	Yes	
Document discussions	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.
Document related tasks	No	No	
Document change history	No	No	
Document access rules	Yes	Yes	Role based access rules
Part			
Part metadata	Yes	Yes	

Windchill ProjectLink Object	Supported by a Loader?	Can Be Exported?	Comments
Part viewable	No	No	
Part markup	No	No	
Part thumbnail	No	No	
Part EPM docs	No	Yes	
Part structure	Yes	Yes	
Part discussions	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	
Project			
Project files	Yes	Yes	
Project metadata	Yes	Yes	Not all project metadata is exported
Project discussions	No	No	
Project template	Yes	Yes	
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	

Windchill ProjectLink Object	Supported by a Loader?	Can Be Exported?	Comments
Project workflow instances	No	No	
Team			
Team metadata	Yes	Yes	
Team roles (including dynamic)	Yes	Yes	
Team members	Yes	Yes	
Activity			
Activity metadata	Yes	Yes	
Activity deliverables	Yes	Yes	
Activity discussions	No	No	
Activity resources	Yes	Yes	
Activity tasks	No	No	
Activity notebooks	No	No	Also called references
Summary activity			
Summary metadata	Yes	Yes	
Summary child links	Yes	Yes	
Summary discussions	No	No	
Summary notebooks	No	No	Also called references
Summary resources list	Yes	Yes	
Summary task list	No	No	
Subproject			
Subproject metadata	No	No	
Subproject project notebooks	No	No	Also called references
Milestone			
Milestone metadata	Yes	Yes	

Windchill ProjectLink Object	Supported by a Loader?	Can Be Exported?	Comments
Milestone deliverable links	Yes	Yes	
Milestone discussions	No	No	
Milestone notebooks	No	No	Also called references
Meeting			
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	
Discussion			
Discussion topics	No	No	
Discussion postings	No	No	
Discussion posting links	No	No	
Discussion posting attachments	No	No	
User Notebook			
Reference folder structure	No	No	
Reference links	No	No	
Reference attachments	No	No	
Action item			
Action item metadata	No	No	
Action item discussions	No	No	
Action item notebooks	No	No	Also called references
Action item meeting links	No	No	

Supported Windchill PDMLink Objects

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

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

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Document			
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	
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

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Associated change objects	No	No	
Associated meetings	No	No	
Associated discussions	No	No	
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, and so on.</p>
Associated processes	No	No	
Associated template	No	No	
Subscriptions	No	No	
Template documents	Yes	Yes	Loaded as part of a Container Template
Part	LoadPart		
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	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Product structure	Yes	Yes	
Config spec	No	No	
Views	Yes	Yes	Views are supported, but the target system must have the corresponding View object
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	<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, and so on.</p>
Associated processes	No	No	
Associated annotations	No	No	
Version history	No	No	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Iteration history	No	No	
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	Loader is LoadValue
Discussions	No	No	
Subscriptions	No	No	
Meetings	No	No	
Part templates	No	No	
EPM Docs			
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 Object	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, and so on.</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	
Save As history	No*	No	
Discussions	No*	No	
Subscriptions	No*	No	
Meetings	No*	No	
EPMDoc template	No*	No	
Serialized Parts			
Metadata	No	Yes	
Serialized Part instances	No	No	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Product/Library Container	LoadContainer		
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	
Product/End Object	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 object

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
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	
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	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Change Issue/Problem Report			
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	
Change Request			
Metadata*	No	No	
Associated life cycle*	No	No	
Associated processes*	No	No	
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	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Affected data*	No	No	
Subscriptions*	No	No	
Change Order/ECN			
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	
Change Activity			
Metadata*	No	No	
Associated content/attachments*	No	No	
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	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Discussions	No	No	
Subscriptions	No	No	
Change Investigation			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated analysis activity*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
Change Proposal			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated analysis activity*	No	No	
Discussions*	No	No	
Subscriptions*	No	No	
Analysis Activity			
Metadata*	No	No	
Associated content/attachments*	No	No	
Associated data*	No	No	
Associated change proposals*	No	No	
Associated change investigations*	No	No	
Discussions*	No	No	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Subscriptions*	No	No	
Worklist			
Tasks*	No	No	
Metadata*	No	No	
Associated change objects*	No	No	
Associated processes*	No	No	
Associated data*	No	No	
Cabinets & Folders			
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
Links	N/A	Yes	
Meeting			
Metadata	No	No	
Discussion	No	No	
Action items	No	No	
References	No	No	
Subject list	No	No	
Discussion			

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Topics	No	No	
Postings	No	No	
Posting links	No	No	
Posting attachments	No	No	
Organization	LoadContainer		
Metadata	Yes	No	
User	LoadUser		
Metadata	Yes	No	
Group	LoadUser		
Metadata	Yes	No	
Members	Yes	No	
Domain	LoadUser		Requires createDomain method
Metadata	Yes	No	
Structure	Yes	No	
Access control rules	Yes	No	
Indexing policies	No	No	
Notification policies	No	No	
Team Templates	LoadTeam		
Metadata	Yes	No	
Roles (including dynamic)	Yes	No	
Team Instances			
Metadata	No	No	
Roles (including dynamic)	No	No	
Members	No	No	

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Groups	No	No	
Life Cycle Templates	LoadLifecycle		
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	
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	
Workflow Templates	LoadWorkflow		
Metadata	Yes	Yes	Import/export is possible using the Workflow Administrator
Process	No	No	
Workflow Instances			
Metadata	No	No	
Process status	No	No	
Associated data	No	No	
Report Templates			

Windchill PDMLink Object	Supported by a Loader?	Can Be Exported?	Comments
Metadata	Yes	No	
Miscellaneous Admin			
Preference settings	Yes	No	
Replication policies	No	No	
Type definitions	Yes	Yes	Loader is LoadValue
Attribute definitions	Yes	Yes	Loader is LoadValue

Configuration Specification Settings

An export operation refers to a configuration specification to determine the data to include in the JAR file, and the selection of configuration specification is made in the following manner:

- If the user performing an export operation for the first time does not select a configuration specification, the current preference for configuration specification determines the objects exported. In this case, the current preference for configuration specification is saved as a modifiable default for the future.
- If the user performing an export operation for the first or any other time selects a configuration specification, the selection determines the objects exported and is saved as a modifiable default for the future.

EPMDocuments

This section discusses some limitations related to EPMDocuments and the behavior related to exporting and importing EPMDocuments as checked out.

Attribute Limitations

Because the attributes of CAD documents are tightly related to content files, there are limitations on which attribute can change outside the workgroup manager clients. The following import actions are not supported for CAD documents:

- Create a new object with new identities
- Substitute for an existing object
- Ignore object

Mapping rules allow a user to change any attribute specified in an import or export XML file. When working with CAD documents, only the following attributes should be changed by mapping rules:

- name
- number
- CADName
- description
- folderpath
- versionInfo
- lifecycleInfo
- teamIdentity

When importing CAD documents, mapping rules should not be used to change the following attributes:

- ownerApplication
- authoringApplication
- epmDocType
- epmDocSubType
- extentsValid
- contentItem
- iba

Mapping rules should not be used to change any attributes on other EPM link objects, including:

- EPMMemberLink
- EPMReferenceLink
- EPMVariantLink
- EPMContainedIn
- EPMDescribesLink
- EPMBuildLinksRule
- EPMBuildHistory

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.

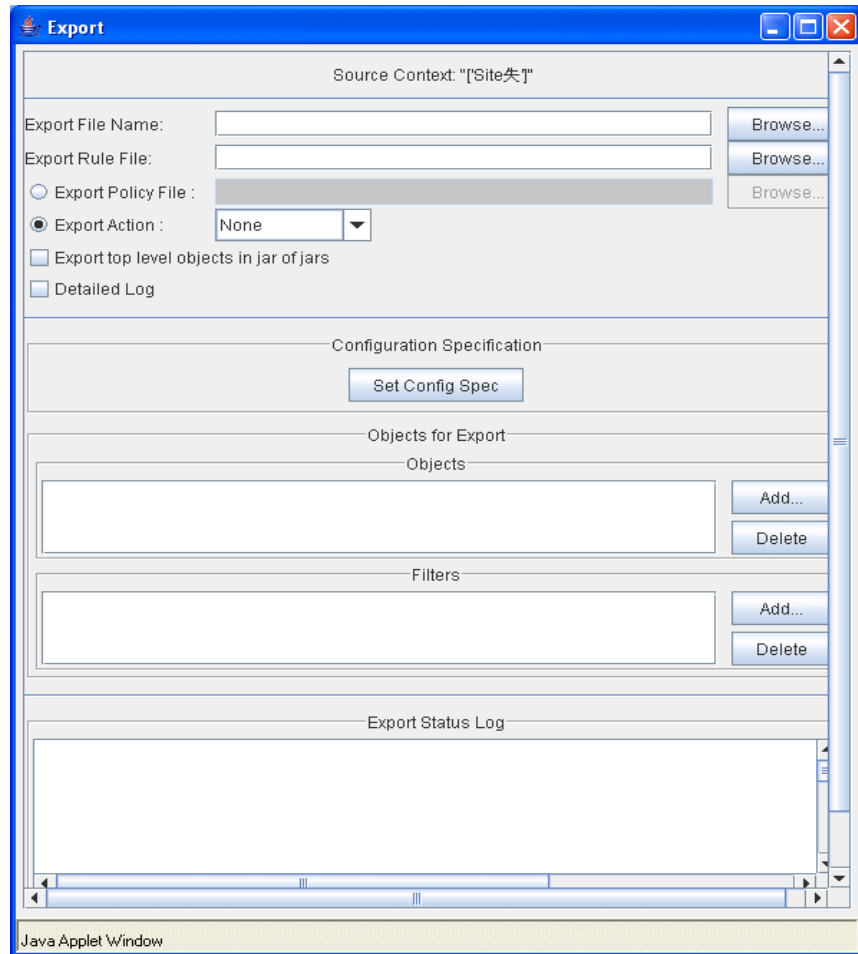
Export User Interface

If you have read access to an object, are an administrator of the object's context (for example, product, library, or project), are an administrator of the parent context (the organization), or are a site administrator, you can perform an export. Administrators can access the **Export/Import Manager** from the **Utilities** tab of the context. Export from a project should be done via the Import from File action on the **Project Details** page, or through the export action on the object/folder action list.

Exporting from the Export Window

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

1. Navigate to **Site > Utilities > Import/Export Manager**.
2. Click **Export** to display the **Export** 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.

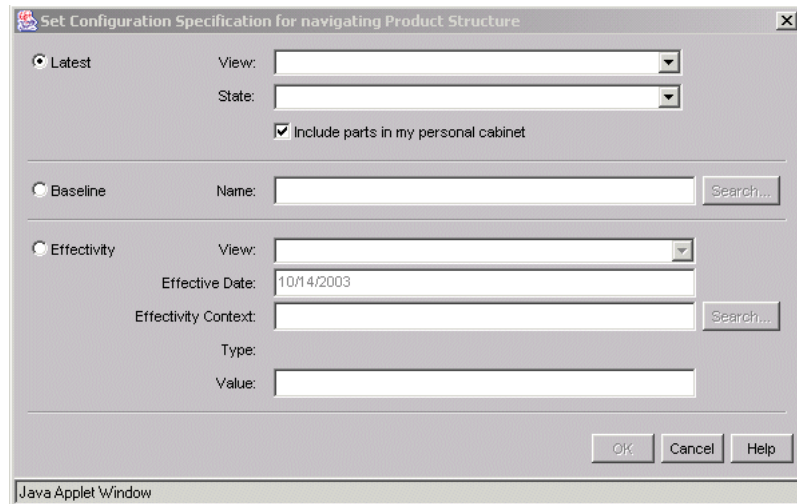
5. You may optionally specify export policies by one of the following two methods.
 - Select the **Export Policy File** option and click **Browse** for the **Export Policy File** box. Specify the folder and file in the dialog box that appears. Export actions in the file will be combined with ones found 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 objects in the export file). The lock action is not shown as a selection, but it is applicable through an export policy file or the system registry.
 - None -- If you are sure no actions such as checkout and lock are needed, this is an appropriate selection.
 - System default -- Actions specified in the system registry will be applied.
 - Check out -- Upon export, the database object will become checked out by you. The software attempts to check out objects that are necessary to describe an object that you are exporting, such as a document that describes a part that you are exporting. You cannot check out type definitions on export.
6. You may optionally export the top-level objects in separate jars. Select the Export top-level objects in jar of jars check box.

You are also able to import individual jars in an exported jar file.

Note: When such a JAR is imported into a Windchill system, individual JARs are imported in separate transactions. Failure in the import of a JAR does not affect the import of the other top level object JARs.

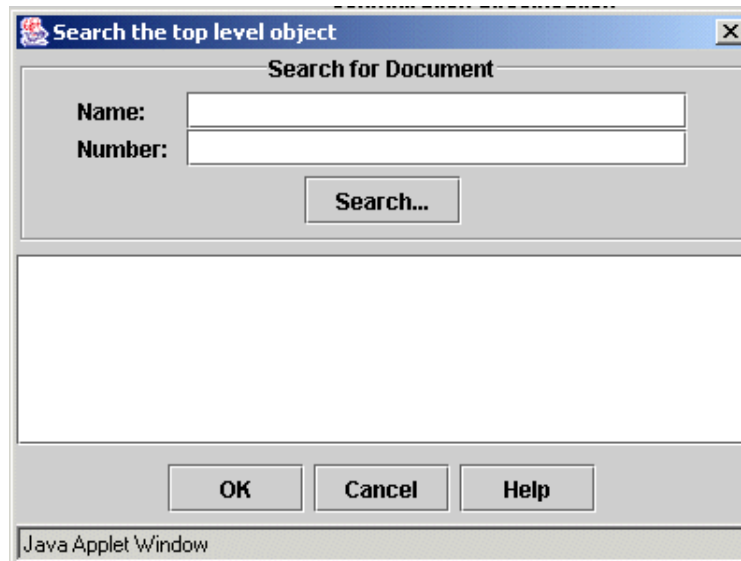
7. You may optionally create a log. Select the **Detailed Log** check box.

8. Set the configuration specification in the **Configuration Specification** section of the window to specify a configuration specification, baseline, or effectivity for the exported object. Setting the configuration specification is optional.



The dialog box is titled "Set Configuration Specification for navigating Product Structure". It contains three radio buttons: "Latest", "Baseline", and "Effectivity". The "Latest" radio button is selected. Below the "Latest" radio button, there are two dropdown menus labeled "View:" and "State:". Below these, there is a checkbox labeled "Include parts in my personal cabinet" which is checked. The "Baseline" radio button has a "Name:" text box and a "Search..." button next to it. The "Effectivity" radio button has a "View:" dropdown menu, an "Effective Date:" text box with the value "10/14/2003", an "Effectivity Context:" text box, and a "Search..." button next to it. Below the "Effectivity" section, there are two more text boxes labeled "Type:" and "Value:". At the bottom right, there are three buttons: "OK", "Cancel", and "Help". The status bar at the bottom says "Java Applet Window".

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



The dialog box is titled "Search the top level object". It has a subtitle "Search for Document". It contains two text boxes labeled "Name:" and "Number:". Below these text boxes is a "Search..." button. Below the "Search..." button is a large empty rectangular area. At the bottom, there are three buttons: "OK", "Cancel", and "Help". The status bar at the bottom says "Java Applet Window".

10. Click **Add** in the **Filters** section of the window to specify a time period that defines the objects for export.

Adding filters is an optional step. You can filter objects to be exported by their time of last modification in all languages, but Time Range user interface is available for English locale only. For other languages, the only user interface option is "during previous ..days/hours/months". This variation affects the features available in the **Filter by Time** window.

Note the following when working with filter:

- 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.
- Time-based filters are not recommended for navigators that are used to export objects along with their relationships. Unpredictable results can happen for export as well as import.

For example:

Use of the time-based filter with Product Structure Navigator results in an incomplete or broken product structure.

- When you export using a filter with a time range of one hour, a jar file is not created.

11. If you need to, click **Preview** to understand what will be exported. With **Detailed Log** not checked, you can see how many objects will be exported and how many XML files will be processed. With **Detailed Log** checked, you can see what files will be exported.

12. Click **Submit**.

Messages in the **Export Status Log** section of the **Export** window show progress or problems that you can resolve. See the appendix that explains mapping rules, policy files, and conflict messages in the *Windchill System Administrator's Guide* for information that can help in resolving conflicts.

Exporting an Object from Its Properties Page

A part, document, product, or a folder's contents, can be exported from its properties page, by performing the following steps. When exporting from the properties page, you cannot add additional objects for export.

1. Navigate to the properties page of the object you want to export.
2. In the **Pick an Action** drop-down list, select **Export** and click **Go**. The **Export Manager** window appears, followed by the **Export** page with the object already selected in the **Objects for Export** field.

3. Specify a folder and file name in the local file system for the exported data jar file: Click **Browse** for the **Export File Name** box. Specify the folder and file in the dialog box that appears.
4. You may optionally specify a mapping rule file in the local file system to control the export process. Click **Browse for the Export Rule File** box. Specify the folder and file in the dialog box that appears.
5. You may optionally specify export policies by one of the following two methods.
 - Select the **Export Policy File** 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 are applied to all objects in the export file). The lock action is not shown as a selection, but it is applicable through an export policy file or the system registry.
 - None -- If you are sure no actions such as checkout and lock are needed, this is an appropriate selection.
 - System default -- Actions specified in the system registry is applied.
 - Check out -- Upon export, the database object becomes checked out by you. The software attempts to check out objects that are necessary to describe an object that you are exporting, such as a document that describes a part that you are exporting. You cannot check out type definitions on export.
6. Click **Set Config Spec** in the **Configuration Specification** section of the window to specify a configuration specification, baseline, or effectivity for the exported object.
7. In the **Navigators** section of the window, select either **Product Structure** (built with active configuration specification) or **Product Structure with CAD Document** (built with active configuration specification).
8. Click **Add** in the **Filters** section of the window to specify a time period that defines the objects for export.

Adding filters is an optional step. You can filter objects to be exported by their time of last modification in all languages, but Time Range user interface is available for English locale only. For other languages, the only user interface option is "during previous ... days/hours/months". This variation affects the features available in the **Filter by Time** window.

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 objects will be exported and how many XML files will be processed. With **Detailed Log** checked, you can see what files will be exported.
10. Click **Submit**.

Messages in the **Export Status Log** section of the **Export** window show progress or problems that you can resolve. For more information, see [Import and Export Policies, Mapping Rules, and Conflict Messages](#).

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 objects from various source contexts. Then, if an import object exists in the target database, import policies or import actions selected from the **Import** window are applied to determine how that object is modified.

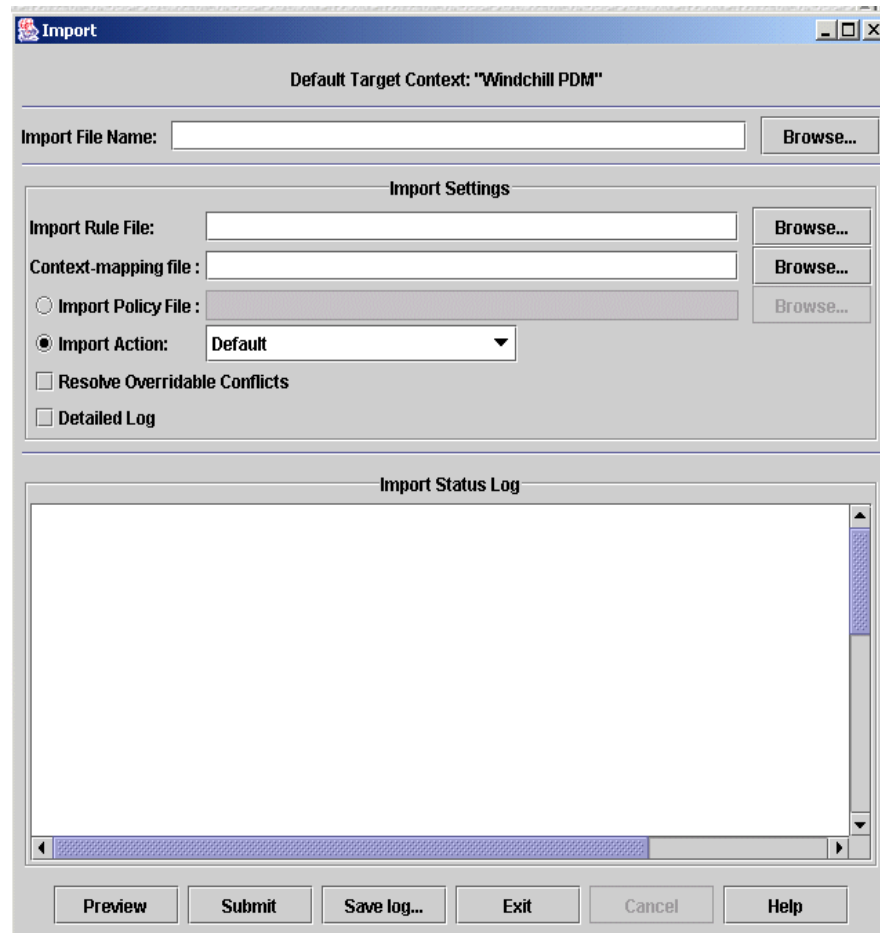
Note: The Access Control List (ACL) applies to import operations. For example, a user without Revise privileges for a particular object type cannot defeat ACL control by using the import action, **Import as new version**. If an import attempt specifies an object action for which you do not have privileges, the entire transaction will fail.

The import action is available to administrators of a context or its parent context (for example, organization or site). Imported objects are created in containers as specified by context mapping rules, provided that the administrator has write access to the mapped context. Type definitions should only be imported at the site or organization levels. Product or library objects should not be imported into the site or organization levels.

Do not use the **Import/Export Manager** to import data into a project. Instead, use the import action available from the **Project Details** page or the folder import action. There is a slight difference in how these two actions work. The import action from the **Project Details** page will provide the default behavior for the folder structure of all the objects in the target set. When the import into folder action is performed, all foldered objects are placed into the target folder.

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

1. Navigate to **Site > Utilities > Import/Export Manager**.
2. Click **Import**.
The **Import** window opens, displaying your current context at the top of the window.
3. Perform the following steps in the **Import** window to import data.



4. 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.
5. 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.

6. You may specify a context mapping file in the local file system to identify into which target context the import file objects are placed. If you do not specify a context mapping file, objects will be imported into the context from which the import action was launched, the **Default Target Context** listed at the top of the **Import** window. Click **Browse** for the **Context-mapping File** box. Specify the folder and file in the dialog box that appears. For a more complete explanation see a later section in this document, [Controlling the Destinations of Imported Objects with Context Mapping Files](#).
7. 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 objects in the import file):
 - **Default** -- If the import object matches the full object identity of an existing database object, the existing object is picked and no import takes place. If the import object is new, it will be created with a version.iteration matching the version.iteration in the import XML file.
 - **Import as latest iteration** -- If the import object is newer than the existing db object, the import process will create an object with the next available iteration on the latest iteration. If the object does not exist in the target database, applying this action will create a new object. If you are sure that a new iteration should be created for all objects that can be iterated, this option is appropriate.
 - **Import as new version** -- If there is a version of the import object in your site's database, the import process will create an object with the next available version and the first iteration. Otherwise, a new object will be created. If you are sure that a new version should be created for all objects that can be versioned, this option is appropriate.
 - **Import as checked out** -- If there is a version of the import object in your site's database, the import process will create a checked out (working) copy of the existing object with the same version as in the import XML file. The newly imported object will be kept checked out until you check it in. If you are sure that all existing objects should be checked out for all objects that can be checked out, this is an appropriate option.

- Modify non-versioned attributes -- This option will update certain non-versioned attributes (for example, life cycle) of an existing database object without iterating the object.
 - Update checked out object in place -- This option will replace the content, attributes, and links of the checked out object.
8. 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`.

9. Click **Preview** to understand what will be imported. With the **Detailed Log** cleared, you can see how many objects will be imported and how many XML files will be processed. With **Detailed Log** checked, you can see what files will be imported, what conflicts may arise during import, and whether the import process will be completed or will fail. PTC recommends using Preview, especially for checking the effect of policy files, which have the potential of creating significant changes to the database. The Preview action does not perform actual import, nor does it report all conflicts, especially those from runtime.
10. Click **Submit**.

Messages in the Import Status Log section of the **Import** window show progress or problems that you can resolve. For more information, see [Import and Export Policies, Mapping Rules, and Conflict Messages](#).

Additional Export and Import Actions

The following export and import actions are available, but are not presented as options in the graphical user interface.

Additional Lock Export Action

The lock action can be imposed through an import policy file or the system registry.

Additional Import Actions

These actions are imposed through an import policy file or the system registry:

- **UnlockAndIterate** -- This action finds an object in the database with the same UFID or the same name, number, version, and iteration as the object in the XML file. If such an object exists and it is locked, this action unlocks and iterates it, and then updates it with information from the XML file. Otherwise, the action generates an error.
- **Add new attributes** based on what you would like added to the configuration as well as the user's preferred setting. This not only makes the installation process easier, but also makes it much easier to track when it comes time to upload your data to the load cache server.
- **CreateNewObject** -- This action creates a brand new object with new name, new number, new version, and new iteration provided in the import policy file. Other information is extracted from the XML file. This functionality must be used with a policy file that provides new identities for the object.

The format of new information that must be provided in 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 the following:

- `<actionInfo>` must always exist.
- Criteria can be any valid attribute of the object in XML file.
- Between `<xsl:choose>`, there can be many `<xsl: when test>` with different criteria and different action names.
- Only `CreateNewObject` and `SubstituteObject` can have action parameters, and there are only four action parameters: `<newName>`, `<newNumber>`, `<newVersion>`, and `<newIteration>`, and all of them must be provided.
- `SubstituteObject` -- This action substitutes the object in the XML file for an object in the database that has the name, number, version, and iteration provided in the Import Policy file. If such an object 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 object in the XML file. This action does not require any actor.

Properties

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 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 ensures that the export operations generate valid XML files for virtually any XML parser, when the XML files contain "strange" characters.

If this property is set to false, some values in the XML files (mostly for the attributes, which are user editable, such as description of WTDocument objects) may not be acceptable by the XML parser even if they are manually wrapped by Cdata Sections.

- wt.ixb.export.filterByConfigSpec (name in wt.properties file or in mapping files)

The default value is false. The property allows to turn off the Config Spec filtering for a single export session and allows to export non-latest WTParts and WTDocuments from their detail page. Possible values of the property are true and false.

Windchill Import Properties

Certain 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 is assigned to the object.

- import.default.lifecycleInfo.lifecycleState

When the life cycle state is missing from the XML file, the default state that is the value of this property will be assigned to the object.

- wt.iba.definition.hierachicaldefinition.enabled

You should 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.

- wt.ixb.import.allowCrossReleaseImport

This property allows a user to be able to import a cross release export jar into a Windchill system. Currently, this property is applicable only for the import operations done through the Windchill Export/Import Manager. Cross-release imports are not supported Out of the Box (OOTB); this property simply allows the import operation to proceed.

The cross release import may or may not succeed even after enabling this property considering major feature/functionality changes in the subsequent releases. Furthermore, the integrity of the imported data cannot be guaranteed. It is highly recommended to avoid performing cross release imports.

The possible values for this property are true or false. Its default value is false.

- `wt.ixb.import.allowTypeDefinitionUpdate`

This property enables the update of typedefinitions on the target server during an import operation. When there exists a mismatch between the typedefinitions of the target system and the export jar, enabling this property allows the typedefinition in the export jar to override the system's typedefinition. Possible values of this property are true and false. The default value is false.

Note: This property is valid only for import actions capable of updating typedefinitions.

Access Control

The Access Control List (ACL) applies to both export and import operations. If an export or import attempt specifies an object action for which you do not have privileges, the entire transaction fail.

For example, a user without Revise privileges for a particular object 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 on as an 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. 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 Rules for WTParts

Domain	Context	Type	State	Principal	Grant Permissions
System	Site	View	All	non-administrator user	Full Control (All)

Import Access Control Rules

You create the WTParts, WTDocuments, EPMDocuments, and Folders as an administrator, and you import them as a non-administrator.

Import Rules for WTParts

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

7

Content Holders and Content Objects

This chapter describes the following:

- Content Handling Configuration
- Adding and Updating Data Formats (which define MIME types for downloading content objects)

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

About 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 [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 [Replication](#).

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 false . This value should be set to true only if Windchill needs to store files from some other application or process that actively utilizes 0-byte placeholder files.

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 true . This value should be set to false 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.

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

Background Queues

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.

Topic	Page
About Background Queues	8-2
Queue Entry States	8-2
Out-of-the-box Background Queues	8-3
Configuring Background Queues and Related Properties.....	8-10
Regular Queue Maintenance	8-14

About Background Queues

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 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 Windchill Index Search, 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](#).

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 **Site > Utilities** page.

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 Ready 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 Ready . Changing a state can be accomplished from the Queue Manager tab in the System Configurator.
Executing	Indicates that an entry is selected for execution. From the Executing 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 Completed 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 Failed entries so that the size of a queue does not exceed the storage capacity of the database.
Reschedule	Indicates that the queue entry has been executed and automatically rescheduled.

State	Description
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 [External File Vaults](#) and the queues used for content replication are described in the [Replication](#).

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 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.useScheduledRecompute` properties 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 Windchill Index Search. These queues are used in the indexing process. You should periodically check the Indexing Queue and Bulk Indexing Queue for failed entries.

If an indexing queue entry has a **Severe** status, ensure that the indexing and the search engine are configured correctly and that the Windchill Index Search software 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).

There are two types of subscriptions/notifications supported by the Notification Service:

- E-mail subscriptions/notifications
- Object Listener subscriptions/notifications

These are subscriptions that can be created where the subscription "recipient" is a Windchill object that implements the `wt.notify.ObjectSubscriptionListener` interface. `WfSynchRobot` is one of these objects. Additionally, the Discussion Forum, Discussion Topic and Discussion Posting interfaces implement this. Whenever these Object Listener subscriptions are satisfied, instead of an e-mail being sent to a user, the Notify Service invokes a method on the subscription's `ObjectSubscriptionListener` "recipient" object and the object does whatever processing is desired.

The NotificationQueue is used for both the e-mail subscriptions and the Object Listener subscriptions; therefore, NotificationQueue must be started for

WfSynchRobots to be notified so they can do the desired processing when their subscriptions are satisfied. Stopping the NotificationQueue prevents e-mail notifications from being sent and it prevents Object Listener subscribers from functioning.

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.

PartsLinkQueue

The PartsLinkQueue is an optional queue that is set up when Windchill 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 past 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.

PublisherQueues

The PublisherQueues (Publisher QueueL, PublisherQueueM, PublisherQueueH, and PublisherQueueI are used by Visualization Services to manage the publishing of data. These queues are created out-of-the-box.

All submitted publish jobs are added to PublisherQueueL, M, or H (low, medium, and high priorities), depending on the way the job was submitted and the type of data.

This process is controlled by property settings in `wvs.properties`. For details, see `publish.publishqueue, priorities.00` in `wvs.properties.xconf`. The executing job in the PublisherQueueL/M/H is looking for an available queue with a name in the form of PublisherQueueN (where N is a sequential integer starting at 1).

By default, PublisherQueue1 is create; additional publisher queues (for example PublisherQueue2, PublisherQueue3, and so on) can be created with the Windchill Queue Manager for publishing scalability. When the executing job in the PublisherQueueL/M/H finds an available PublisherQueueN, it submits the publish job to that queue, which immediately executes it.

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.

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 in 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.

If the performance of workflow queues is an issue, you can set up queue pooling that includes `WfUserWorkQueue` and `WfPropagationQueue` queues. The `WfPropagationQueue` queues in the pool are named `WfSharedPropagationQueue<n>`, where `<n>` starts at 1 and increments by 1 for each additional queue. For the details on setting up queue pooling, see the *Windchill Performance Tuning Guide*.

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.

If the performance of workflow queues is an issue, you can set up queue pooling that includes WfUserWorkQueue and WfPropagationQueue queues. The WfUserWorkQueue queues in the pool are named WfSharedUserQueue<n>, where <n> starts at 1 and increments by 1 for each additional queue. For the details on setting up queue pooling, see the *Windchill Performance Tuning Guide*.

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 a background method server, 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 a Method Server for Background Queues

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

To start up two method servers where one is dedicated to running background queues, use the `xconfmanager` to ensure that the following properties in `wt.properties` are set:

```
#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
```

Configuring Multiple Background Method Servers

Under most circumstances, a single background method server is sufficient for processing background queues. For information on setting up multiple background method servers, see the *Windchill Advanced Deployment Guide*.

Background Queue Properties

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

Property	Description
<code>wt.queue.<queuename></code>	Assigns the queue named <code><queuename></code> to a queue group. This property is set through the Queue Manager utility when it assigns queues to groups. For additional information, see the <i>Windchill Advanced Deployment Guide</i> .
<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.
<code>wt.queue.execEntriesCount</code>	Sets the number of entries queried from the queue to be executed. Default is 6.
<code>wt.queue.executeQueues</code>	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 <code>wt.queue.queueGroup</code> property that is set. Default is false.
<code>wt.queue.max.processQueues</code>	Sets the maximum number of process queues that the queue service creates before throwing an exception. Default is 25.
<code>wt.queue.max.scheduleQueues</code>	Sets the maximum number of schedule queues that the queue service creates before throwing an exception. Default is 16.
<code>wt.queue.queueGroup</code>	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 About Background Queues .

Property	Description
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 About Background Queues .
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 60000.
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

Logging for queues is configured using the log4jMethodServer.properties file located in codebase/WEB-INF. These properties can also be set in jconsole.

Remember that when these properties are set in the log4j file, they must be prefixed by log4j.logger and be assigned one of the following values:

- TRACE
- WARN
- INFO
- ERROR
- FATAL
- OFF:

Property	Description
wt.queue.processingQueue .queueName	Provides debug information specific to the actual polling threads. 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.scheduleQueue .queueName	Provides general schedule queue information that can be used to debug problems. Default is false.

Other Background Queue-Specific Properties

There are other Windchill properties specific to queuing. The following table includes two 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.

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 slowed performance.

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 available from the **Site > Utilities** page.

Follow these steps to develop a regular routine for queue maintenance:

1. Locate the row corresponding to the queue for which you want to do the regular maintenance.
2. From the **Actions** drop-down list in the corresponding row, select **View Entries**.

The **Entries** window opens.

3. From the **View entries with status** drop-down list, select **Failed** and then click **Show** to view failed entries.
4. 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.
5. From the **View entries with status** drop-down list, select **Severe** and then click **Show** to view severe entries.
 6. 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.
7. Repeat the process until you have either deleted or reset all failed and severe entries in each queue.

9

Publishing Agent

Note: The Publishing Agent is traditionally known as the *CAD Agent*. Configuring the CAD Agent is also pertinent to publishing non-CAD data, such as Dynamic Documents and WTDocuments. The remainder of this chapter and the user interface uses the term “CAD Agent”.

The CAD Agent is used to manage the relationship between one or more workers and Windchill Visualization Services.

Topic	Page
Introduction to CAD Agent.....	9-2
Before You Begin.....	9-2
Guidelines for Configuring a Worker "Common File System"	9-3
Configuring the CAD Agent	9-4
CAD Worker on the Same Machine.....	9-4
Configuring a CAD Worker on a Different Windows Machine	9-9
CAD Worker on a Different UNIX Machine	9-14
Configuring a Non-English UNIX Worker to Run FTP Daemon	9-16
Defining Publish Queue and Worker Availability Settings	9-16
Testing the CAD Agent Configuration	9-17
Using WVS with Multiple Background Servers	9-20
Configuring the CAD Agent in a Windchill Cluster	9-21
Making Thumbnails Viewable for Microsoft Word or Excel Files	9-23

Introduction to CAD Agent

The CAD Agent and CAD workers are an important link in achieving the full functionality of the visualization server. Through the worker, the visualization server can communicate with a number of ProductView Adapters and other workers, which are described in the About Visualization Services chapter.

When the visualization server receives a client request to convert CAD data to ProductView format (or non-CAD data to another format), the server calls a worker. This worker is either an instance of a ProductView Adapter or another type of worker. The worker then opens a background session of the application, and converts the designated objects to the appropriate format.

In a standard worker configuration, when a publishing request is received, the Publisher passes the file to the CAD Agent, which invokes the worker to convert the data and to store the resulting published data in Windchill.

In other words, when instructed by the CAD Agent, the worker opens a non-graphical session of the appropriate application and converts the designated data into another format. For CAD data, the CAD worker converts the designated CAD objects to ProductView format. The Agent tells the workers what to convert, where to obtain the objects, and where to place the converted output files. For non-CAD data the process is similar, although the application used for conversion and the output type would be different, such as using Acrobat Distiller or the Arbortext Publishing Engine to create a PDF file.

This functionality allows automated conversion of CAD data by Windchill Visualization Services, using the ProductView Adapter as a worker. In this mode, the ProductView Adapter is continually available to convert native CAD objects, as the visualization server requires.

Before You Begin

Note: The remainder of this chapter refers mostly to ProductView Adapters and CAD workers. Keep in mind that this information is relevant for non-CAD configurations as well.

When you configure the ProductView Adapter as a worker for the CAD Agent, this process generates a worker script, which launches the ProductView Adapter in a server-controlled mode. In this mode, the ProductView Adapter is referred to as a CAD *worker*. The complete path to the worker script is required when configuring the CAD Agent to use the worker, as detailed in the next section. Also, you must know the visualization server host name and port number to complete the configuration process.

Note: The ProductView Adapter must be configured separately. For more information, refer to the section titled *Server-Controlled Publishing* in your relevant ProductView Adapter configuration chapter.

Configuring the CAD Agent to use a CAD worker follows the same pattern, regardless of the type of worker used by your system. The subsequent sections of this chapter use the Pro/ENGINEER ProductView Adapter as an example, which uses the **proeworker** script to launch the worker.

Guidelines for Configuring a Worker "Common File System"

“Common File System” refers to the physical folder on the server and worker machines that, when combined using a network sharing technique (e.g., Windows Mapped Drives or UNC paths, the Windows IIS or UNIX FTP servers), are considered as a logically common file-system. Once configured, they are used by the CAD Agent to transfer Windchill content to the worker machine and retrieve the converted viewable files. You can configure “Common File Systems” to communicate between various Windows and UNIX server/worker combinations. This section describes which tools are supported for the various configurations.

Note: For any combination other than Windows to Windows, you must use an FTP setup between file systems.

The configurations that are tested and supported by PTC are listed below.

- Windows server to Windows worker:
 - Uses the WorkerDaemon service to start the worker.
 - Transfers data via a common file system using a Windows mapped drive or a UNC path.
 - Transfers data via a common file system using FTP to the IIS FTP Service installed with Windows on the worker machine.
- Windows server to UNIX worker:
 - Uses Telnet to start the worker.
 - Transfers data via a common file system using FTP to the bundled FTPD service installed with the UNIX operating system on the worker machine.
- UNIX server to Windows worker:
 - Uses the WorkerDaemon service to start the worker.
 - Transfers data via a common file system using FTP to the IIS FTP Service installed with Windows on the worker machine.
- UNIX server to UNIX worker:
 - Uses Telnet to start the worker.
 - Transfers data via a common file system using FTP to the bundled FTPD service installed with the UNIX operating system on the worker machine.

Configuring the CAD Agent

The CAD Agent communicates with the CAD worker under four possible scenarios. The worker is located either on the same machine as the Agent or on a different machine.

This chapter contains procedures for four different worker configurations:

- Scenario 1: The worker is on the same Windows machine as the visualization server.
- Scenario 2: The worker is on a different Windows machine than the visualization server.
- Scenario 3: The worker is on the same UNIX machine as the visualization server.
- Scenario 4: The worker is on a different UNIX machine than the visualization server.

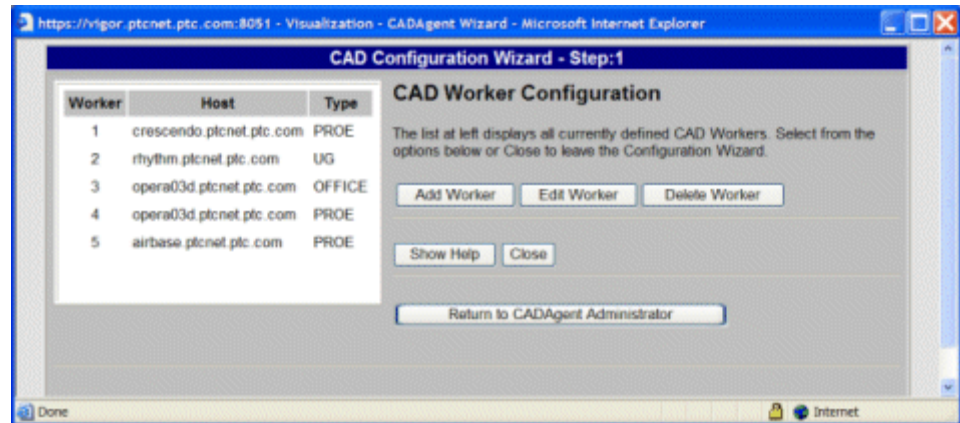
CAD Worker on the Same Machine

In the simplest scenario, the CAD worker and the CAD Agent are on the same machine. The instructions for Windows and UNIX are the same when the visualization server and the CAD worker reside on the same machine.

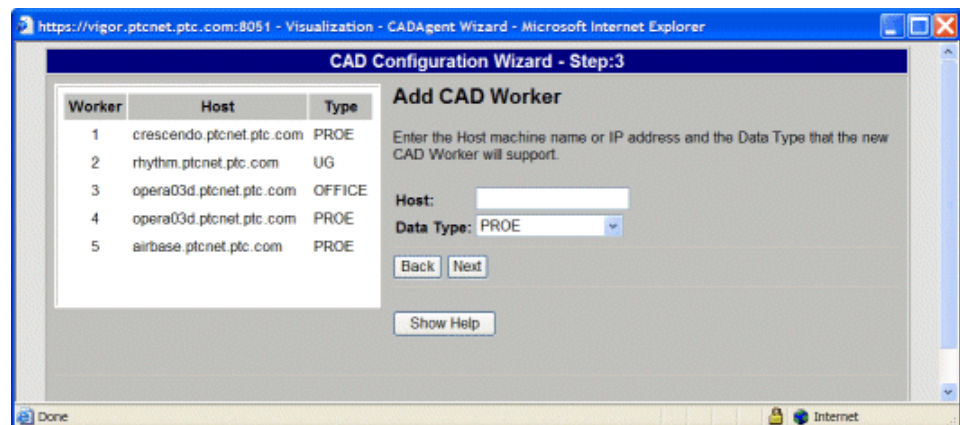
1. Open Windchill in a Web browser.
2. In the Windchill application, do the following to navigate to the CAD Agent Administrator: Click on the **Site** (tab) > Utilities (tab) > **CAD Agent Administrator** (link located under the System Administration heading).
3. The **CAD_Agent Administration** window opens.

This window displays a list of the configured CAD workers and general statistics about the worker, for example, its running status. From this window, you can start and stop, and reload the CAD worker.

- Click **Configure** to configure a CAD worker.



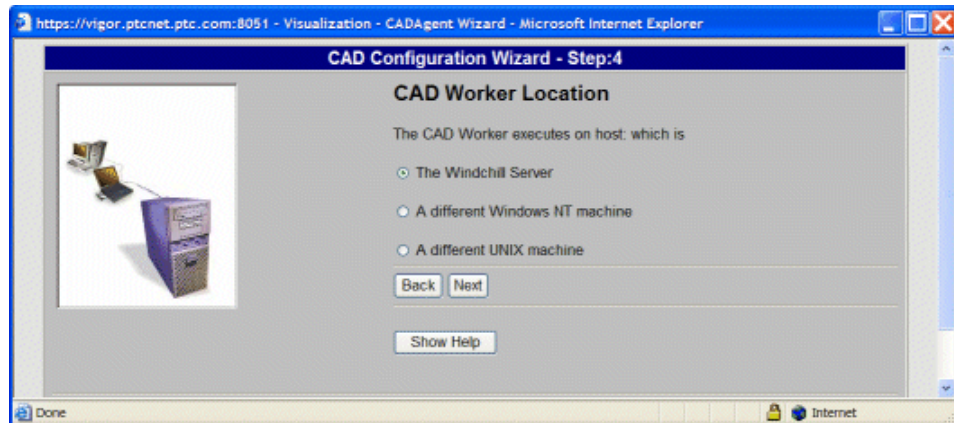
4. In the **CAD Worker Configuration** panel, you are presented with three options: **Add Worker**, **Edit Worker**, and **Delete Worker**. In this example, the **Add Worker** option is described.
5. Click **Add Worker** to add a new worker configuration.



6. In the **Add CAD Worker** panel, you specify the host name of the visualization server and the data type of the CAD Worker to be created. For example, if you are processing Pro/ENGINEER CAD Documents or parts, then you will create a Pro/ENGINEER worker to translate the CAD objects into a viewable image.
 - **Host** — Specify hostname of the visualization server machine.

- **Data Type** — Specify the data type of the worker to be created. The data type options available are:
 - PROE
 - CADDs
 - CATIA
 - Unigraphics
 - IDEAS
 - SolidWorks
 - PROD
 - SOLIDDESIGNER
 - ME10
 - ECAD
 - THUMBNAIL
 - CATIAV5
 - IGES
 - VRML
 - OFFICE
 - Autodesk Inventor
 - JT
 - Arbortext
 - STL
 - DGN
 - GBF
 - PGL
 - STEP

7. Click **Next**. This displays the **CAD Worker Location** panel.

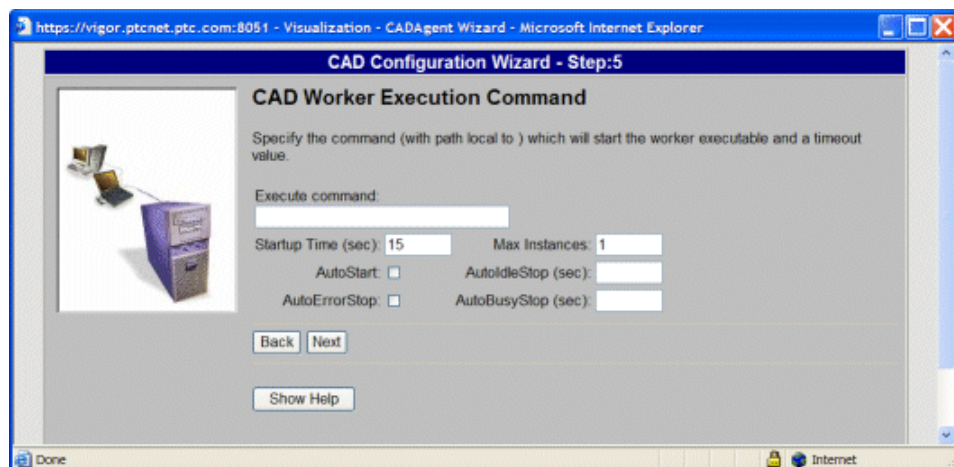


8. Select the location where the CAD worker resides:

- **The Windchill Server** — The CAD worker resides on the same machine as the Windchill server.
- **A different Windows NT machine** — The CAD worker resides on a Windows machine other than the Windchill server.
- **A different UNIX machine** — The CAD worker resides on a UNIX machine other than the Windchill server.

In this example, since the CAD worker resides on the same machine as the visualization server, select **The Windchill Server**.

9. Click **Next** to display the **CAD Worker Execution Command** panel.



In this panel, you specify the script used to start the worker and the execution settings. Enter values for the following parameters:

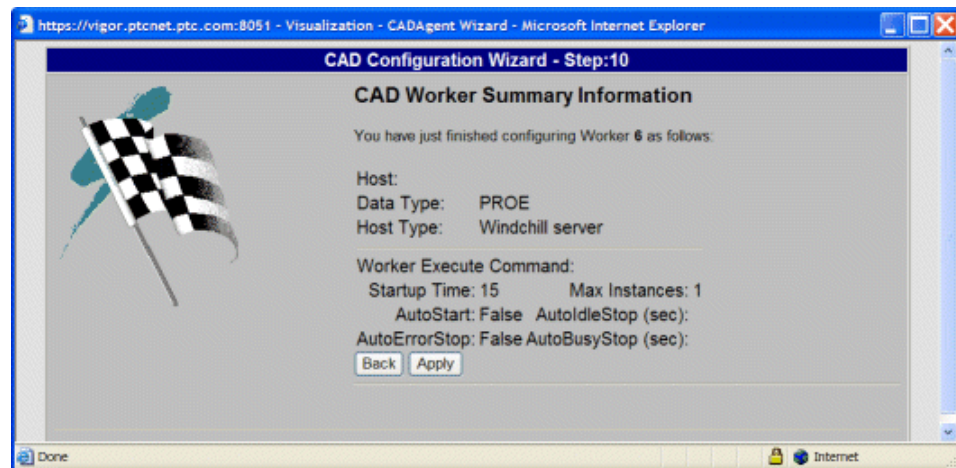
- **Execute Command** — The execute command is the CAD worker script created during the ProductView Adapter configuration. See the *Server-Controlled Publishing* section in the relevant ProductView Adapter configuration chapter. Type the complete path to this script in the Execute Command text field.
- **Startup Time (sec)** — Specify an amount of time in seconds that it takes for the worker and the CAD system to completely start, initialize, and be able to return a message confirming this. If this value is not large enough, the worker does not start reliably. Adjust the setting to work on your system.
- **Max Instances** — Specify the maximum number of instances of this worker that can be started. You should only set this value to be greater than one (1) if the worker machine is capable of running multiple copies of the CAD application effectively (that is, a multiprocessor system with sufficient memory).
- **AutoStart** — Select this option to set the worker to start automatically whenever it is needed. You should use this option if you want your system to automatically start up after a reboot or if you want the CAD agent to automatically satisfy heavy demand.
- **AutoIdleStop (sec)** — Specify an amount of time in seconds that determines how long the worker can remain idle before it is automatically stopped. If you leave this field empty or set it to zero (0), the worker is not stopped automatically at idle time. Use this option if you need to release the CAD license when it is not being used.
- **AutoErrorStop** — Click this option to set the worker to stop automatically whenever it returns an error from a conversion request. Use this option if you need to make sure that the CAD application is not left in an unstable state after an error condition.

Note: It is better to control the worker's error behavior using the related worker Recipe settings (**Recipe Editor -> Advanced -> System -> Error Handling**).

- **AutoBusyStop (sec)** — Specify an amount of time in seconds that determines how long the worker can spend on a single conversion request without automatically being stopped upon finishing that conversion. If you leave this field empty or set it to zero (0), the worker is not stopped automatically when it has completed a conversion.

Note: It is better to control the worker's error behavior using the related worker Recipe settings (**Recipe Editor -> Advanced -> System -> Worker Configuration**).

10. Click **Next** to display the **CAD Worker Summary Information** panel. This panel displays the values you entered or selected during the add worker process.



11. Review the CAD worker configuration information in this panel. If needed, use **Back** to return to the appropriate wizard step and make any corrections. When you have verified that the CAD worker information displayed in this panel is correct, click **Apply**.
12. You are returned to the **CAD Worker Configuration** window (step 1 of the wizard). Click **Save File** to update the <Windchill>\codebase\agent.ini file with your configuration changes.
13. Click **Reload CadAgent** to commit the changes and update the **CADAgent Administration** worker entries.

You have now finished configuring the CAD Agent to use a CAD worker on the same machine as the visualization server. Refer to [Testing the CAD Agent Configuration](#) to confirm your configuration is correct.

Configuring a CAD Worker on a Different Windows Machine

The CAD worker can be configured on a Windows machine that is different from the visualization server machine. The configuration procedures for this scenario are exactly the same as the example of configuring the CAD worker on the same machine as the visualization server, except for these differences:

- The need for a common file system shared between the two machines.
- The use of the Worker Daemon is to execution of the worker startup script on a different Windows machine. This communication is handled by using a small executable named WorkerDaemon.exe. The WorkerDaemon.exe is installed and optionally configured as a Windows Service called “GS Worker

Daemon”, as part of the ProductView Adapters installation. These instructions assume that you installed the Worker Daemon.

Creating a Common Windows to Windows File System

For the visualization server machine and the CAD worker machine to exchange files, the two machines must share a common file system directory. This directory, which can be located on either machine (or even on a third machine), serves as a file transfer point between the server and worker machines. It is highly recommended that the physical file system resides on the worker machine, as this provides the best performance.

Note: Because there are numerous ways in which a common file system can be created (mapped Windows drives, FTP, and so forth), no attempt is made to describe the creation of a shared file directory in these procedures.

Tip: If you are using a network file system (NFS), this directory can exist on any machine in the network; but you must be able to mount the directory on both the Windchill machine and the CAD worker machine. If you are using FTP, the Windchill machine must be able to connect with an FTP server running on the CAD worker machine and the worker path must be a directory which is local to the CAD worker machine.

After creating the common file system, make note of the full path to the common directory used by the server machine and by the worker machine. Each of these paths must be entered during subsequent procedures.

For example, from the server machine, the full path to the common file system could be:

```
D:\shared_Worker_dir
```

The full path from the worker machine to the same common file system could be:

```
X:\D_drive\shared_Worker_dir
```

Note: This common file system must be fully accessible from the user account specified during the initial installation of the visualization server. This can require that both machines have the same log on account. This is a security concern the administrator must address by policy before installing the worker.

Adding CAD Worker to a Different Windows Machine

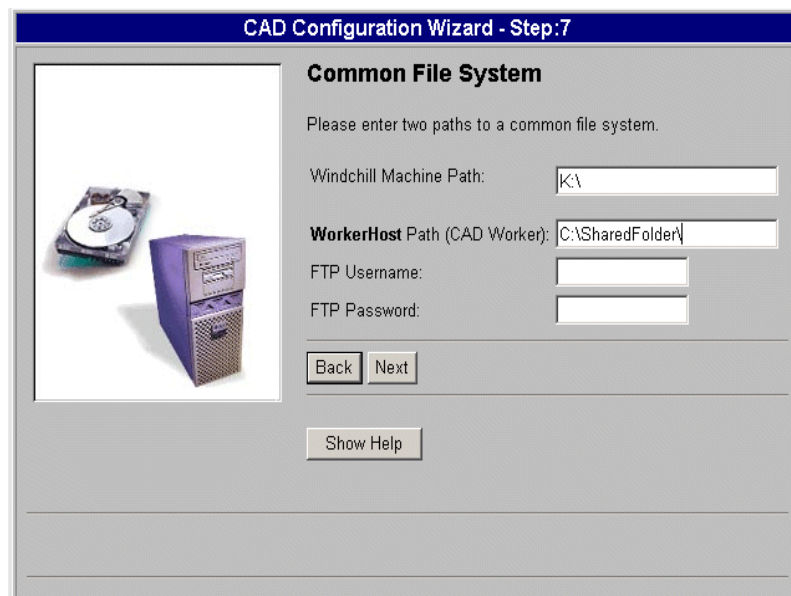
The CAD Agent must know where the CAD worker is on the system. The next set of procedures explains how to configure the CAD Agent to use this worker.

Because the worker configuration steps are similar regardless of whether the configuration being performed is local or remote, the common configuration steps are not repeated. Where possible, you are directed to the detailed instructions described in the CAD Worker on the Same Machine section.

1. Follow steps 1 through 5 in the section [CAD Worker on the Same Machine on page 9-4](#).
2. In the **CAD Worker Location** window, select **A different Windows NT machine** option as in this scenario the visualization server and the CAD worker machines are different, and the worker resides on a Windows machine.
3. Complete the **CAD Worker Execution Command** instructions as described in the [CAD Worker on the Same Machine](#) the section.
4. In the **Worker Daemon Port** window, specify the port number that the Worker Daemon, running on the Windchill server, listens to. This instruction assumes that the Worker Daemon was installed and configured during the installation of the ProductView Adapter(s).

Accept the default, 601, or enter the correct port number.

5. In the **Common File System** window, enter the information needed to connect to the common file system.



CAD Configuration Wizard - Step:7

Common File System

Please enter two paths to a common file system.

Windchill Machine Path:

WorkerHost Path (CAD Worker):

FTP Username:

FTP Password:

- **Windchill Machine Path** — Enter the Location of the common file system relative to the Windchill machine; for example, the drive mapped to the share on the worker machine. In this example, the path is:

K:\

- **<machine_name> Path (CAD Worker)** — Enter the location of the common file system on the Pro/ENGINEER CAD worker machine; for

example, the shared folder to which the Windchill server must connect to via a mapped drive. In this example, the path is:

```
C:\SharedFolder
```

- **FTP Username** — If you used FTP to connect to the common file system location, type the FTP user name; otherwise, leave this field blank.
 - **FTP Password** — If you used FTP to connect to the common file system location, type the FTP password; otherwise, leave this field blank.
6. For the remaining steps, follow the instructions through the end of the [CAD Worker on the Same Machine](#) section.

Running Worker Daemon as a Windows Service

The Worker Daemon procedures described in this section allow you to configure the Worker Daemon to run either as a program (which is the default) or as a service. The next procedure details how to start the Worker Daemon if you are running it as a service:

When you ran the ProductView Adapters installation, the WorkerDaemon.exe was installed in *<ProductView adapter>\i486_nt\obj* directory.

Installing the Worker Daemon

Note: During installation, there was an option to “Configure Worker Daemon as a Windows Service”. If you selected this option during install, then Step 1 is not necessary. Proceed to Step 2 to ensure that the service is available.

1. Open a **Command Prompt** window and navigate to the *<ProductView adapters>\i486_nt\obj* directory.
 - At the prompt, enter the following command to install the GS Worker Daemon as a Windows service.

```
workerdaemon -install
```

2. Open the **Services** dialog box, which now lists a "GS Worker Daemon" service. **Start** the Worker Daemon to begin running it as a service.

Changing the System Account

1. By default, the Worker Daemon Windows service is configured to run as the local SYSTEM account. You can also use the Control Panel **Services** utility to change the log on account it runs under.
 - Open the **Properties** window for the GS Worker Daemon service.
2. From the **Log On** tab, select **This account** option, and pick the account you want to run under. Use **Browse** to display the **Select User** list, and select a name from the list.

For example, you can run the Worker Daemon from the administrator account. Enter a password, and click **OK**. The account you selected is now configured to run the Worker Daemon as a service.

Configuring the Port Number

To configure the port number for the WorkerDaemon, when it is running as a service, at the command prompt enter the reinstall\install command:

```
workerdaemon -install <port#>
```

This command ensures that, when the service is operating, it will be listening on the configured port.

About the WorkerDaemon Log File

When the Worker Daemon is running as a service it produces a log file called workerdaemon.log, which is important, when debugging a Worker Daemon problem.

The log file shows when the Worker Daemon is running, the commands it receives, what port it is listening on, along with other event information.

Removing the Worker Daemon Service

You can remove the Worker Daemon service by executing the following command at the prompt:

```
workerdaemon -remove
```

Running the Worker Daemon as a Program

These procedures describe how to run the Worker Daemon as a program.

There are several ways to start the Worker Daemon. The Worker Daemon can run not only as a service in the background, as described above, but it can also be run as a program. To run the Worker Daemon as a program, select one of the following options:

- Start the Worker Daemon by double-clicking the **WorkerDaemon.exe** file. The Worker Daemon executable on the CAD worker machine must be running at all times, as explained in the next note.

Note: After the user is logged off on the worker machine, the Worker Daemon process ends. Alternatively, the Worker Daemon can remain running by locking the workstation instead of logging out. In this case, the machine is left in a secure state. To lock the workstation, press **CTRL+ALT+DEL**, then click **Lock Computer**.

- Start the Worker Daemon using a command line parameter. By default, the Worker Daemon listens on port 601. The Worker Daemon can be forced to

use another port by specifying the port number as a command line parameter when starting the Worker Daemon.

- Start the Worker Daemon using **Auto Start**. The Worker Daemon is started automatically when the computer starts by creating a shortcut to it and then placing the shortcut in the following directory:

```
%ALLUSERSPROFILE%\Start Menu\Programs\Startup
```

Where %ALLUSERSPROFILE% is the traditionally out-of-the-box Windows environment variable pointing to the "All Users" profile directory.

You have now finished configuring the CAD Agent to use a CAD worker on a different Windows machine. Refer to [Testing the CAD Agent Configuration](#) to confirm your configuration is correct.

CAD Worker on a Different UNIX Machine

The configuration procedures for this scenario are exactly the same as for configuration of the CAD worker on the same machine as the visualization server, except for one difference. The difference is the need for a common file system shared between the two machines.

Note: On UNIX there is no need for the Worker Daemon.exe used on Windows.

Creating a Common File System

For the server machine and the UNIX worker machine to exchange files, the two machines must share a common file system directory. This directory, which can be located on either machine (or on a third machine), serves as a file transfer point for the server and UNIX worker machines.

Create a shared directory that is accessible to both the server machine and the UNIX worker machine. It is important that the path the CAD worker machine uses does not contain any special characters which the worker cannot handle. Because there are many ways to create a shared file directory specific procedures are not provided in this guide. One common method of creating a shared directory is by creating the directory on the UNIX machine and accessing it using FTP on the server machine.

After creating the common file system, make note of the full path used by the server machine and the UNIX worker machine to access the common file system. These paths must be entered during a subsequent procedure.

In the example used in the following procedure, FTP is used to mimic a common file system. The full path from the server machine to the common file system is:

```
FTP:/CW/shared_dir
```

In this example, the full path from the UNIX worker machine to the shared file directory is:

```
/CW/shared_dir
```

Adding a UNIX CAD Worker to the CAD Agent

The CAD Agent must know where the CAD worker is located on the system. The next set of procedures explains how to configure the CAD Agent to use this worker.

1. Follow steps 1-5 in section [CAD Worker on the Same Machine](#).
2. In the **CAD Worker Location** panel, select the **different UNIX machine** option, as in this scenario the visualization server and the CAD worker machines are different, and the worker resides on a UNIX machine.
3. Complete the **CAD Worker Execution Command** instructions described in section [CAD Worker on the Same Machine](#).
4. In the **Telnet Account** panel, specify the information needed to access the remote worker. Enter values for the following fields:
 - **Username** — Specify the account to log on to the machine where the worker resides. This account must already exist and the log on process must not require any interactive log on scripts, the required environment variables must be defined automatically, and no special commands must be required to start the worker.
 - **Password** — Specify the password used to log on to the telnet account.
 - **Shell Prompt** — Enter the shell prompt that the CAD Agent will see when logging onto the account. When the CAD Agent logs onto the account, it will wait to receive this prompt. Once the prompt is detected, the worker executable will be started.
5. In the **Common File System** panel, specify the common file system information you prepared in the [Creating a Common File System](#) section.

If you used FTP, the **Windchill Machine Path** and the **<machine name> Path (CAD Worker)** must be exactly the same except that the **Windchill Machine Path** starts with 'ftp:'. Enter values for the following parameters:

- **Windchill Machine Path** — Specify the mount path of the common directory as seen from the Windchill server machine.
- **<machine name> Path (CAD Worker)** — Specify the mount path of the common directory as seen from CAD worker machine.

Note: It is important that the path the CAD worker machine uses does not contain any special characters that the worker cannot handle.

- **FTP Username** — If you used FTP, specify the account used to start the UNIX worker. The FTP connection is established based on this account. If you are using a network file system, the FTP account information is not required.

- **FTP Password** — Specify the password for the user name connecting to the account.
6. For the remaining steps, follow the instructions through the end of the [CAD Worker on the Same Machine](#) section.

You have now finished configuring the CAD Agent to use a CAD worker on a different UNIX machine. Refer to [Testing the CAD Agent Configuration](#) to confirm your configuration is correct.

Configuring a Non-English UNIX Worker to Run FTP Daemon

For any data access ProductView Adapter, including CADDs File System (CaddsFS), CATIA File System (CFS), Unigraphics File System (UGFS) or Pro/INTRALINK, when retrieving converted files from a remote UNIX worker and when the CAD agent uses FTP for the retrieval, the CAD agent can have problems determining the correct file names, if the file names are not in US/English.

There are special considerations for configuring a CAD worker on a UNIX platform, if a non-English language is involved. The reason for these special instructions is to reduce the likelihood of FTP file transfer issues occurring between a ProductView Adapter running in a non-US/English UNIX locale and the Windchill server using US/English on Windows server.

Note: This occurs in cases where the UNIX ProductView Adapter host is configured with a non-US/English locale (European or Asian), for example, DBCS. If, for example, the CADDs worker is running on Japanese UNIX with Japanese FTP server, the CADDs5 ProductView Adapter can not transfer converted CADDs models from a Japanese FTP server, and only creates a zero (0) byte viewable on the Representation in Windchill.

The use of non-English file names has resulted in various file transfer issues, the most common being the retrieval of zero (0) size files. This can occur because the FTP API was given an invalid file name, for which it does not generate an error condition.

See Technical Application Note (TAN) #106245 for further information.

Defining Publish Queue and Worker Availability Settings

It is possible to control the availability of additional WVS Publisher Queues and Workers, thereby increasing WVS Publishing throughput during off-peak hours. These settings are defined in the WVS properties file, as well as the Worker agent.ini file, as specified below.

In wvs.properties, the publish queue settings use the format:

```
publish.publishqueue.availabletimes.<queueName>=
```

For example:

```
publish.publishqueue.availabletimes.PublisherQueue1=  
publish.publishqueue.availabletimes.PublisherQueue2=
```

Individual workers use the property in that worker's section of the agent.ini file:

```
availabletime=
```

The format of the string that sets the time/days is the same for the worker availabletime as for the publish queue setting:

```
<time1> , <time2> , <timeN>
```

where each time is defined as

```
<start-time> - <end-time> [- <day-of-week>]
```

The start and end times can be repeated as many times as needed. The time of day format applies to the local area where the Method Server is running, as defined by java. For example, the USA would use AM/PM format, whereas the UK would be 24 hour format. Each of the time of day specifications can also include a day of week qualification

An example of a UK day and time format would be:

```
availabletime=19:00-06:00, 05:00-20:00 - sat, 05:00-20:00 - sun
```

where the worker is available each night from 19:00 to 06:00 the next morning, and on Saturday and Sunday is available from 05:00 to 20:00.

An example of USA format for the same times and days would be:

```
availabletime=7:00 pm-6:00 am, 5:00 am-8:00 pm - sat, 5:00 am-  
8:00 pm - sun
```

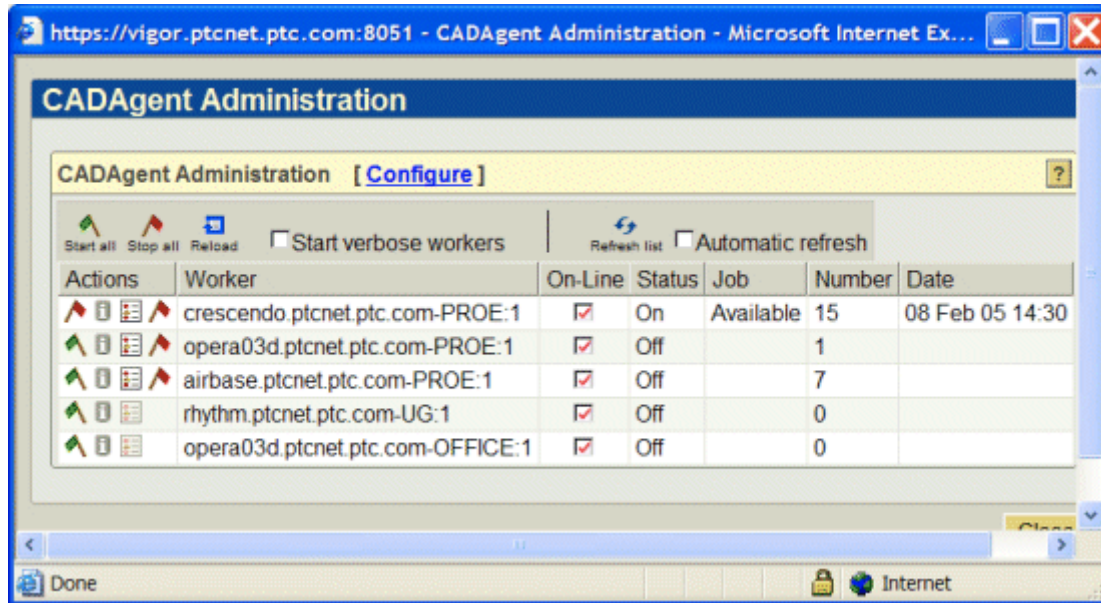
If the format is incorrect, Method Server messages will display.

Testing the CAD Agent Configuration

Once you have completed any one of the four worker configuration scenarios, you can test the operation of the worker. The tests are fairly generic between ProductView Adapters. This section uses the Pro/ENGINEER Adapter as an example.

Test 1: Starting the CAD Worker

1. From within the CAD Agent Wizard, return to the **CADAgent Administration** panel. This panel displays a list of the configured CAD workers and their current status.



2. To start the CAD worker, click the green flag icon in the **Actions** column. After the CAD worker has started, the **Status** column displays **On**.

If the status does not change to **On**, refer to the appendix [Troubleshooting the Publishing Agent](#).
3. When the connection tests prove successful, you can turn off the CAD worker by clicking the left-most red flag icon in the Actions column.

Test 2: Create a Viewable on Demand

To perform this test, you will need CAD document data loaded on the Windchill system. This data must be in its native state, meaning that it has not already been converted to a viewable (published). You will use this data to test the publishing capabilities. This test is valid for 3D models and Office Documents saved with Preview Picture.

Note: The CAD worker must be running to perform this test. If you did not enable **Auto Start** in the worker configuration, the worker must be manually started. See [Test 1: Starting the CAD Worker](#).

1. Open Windchill in a Web browser.

2. Browse to a CAD Document, and open the information page for that document. When CAD Documents are loaded into Windchill through workgroup managers or an EPM gateway, Windchill Visualization Service can publish a representation of this data.

A **ProductView** icon appears in the visualization panel in the upper left corner of the information page. Mousing over the icon displays **Create Representation**.

3. Select the **ProductView** icon.
4. A page opens that notifies you that the request was “Sent to Publisher”, followed by a link to open the Publish Monitor. Click the link and the **Publish Monitor** page opens, displaying the publishing processes. You can observe the CAD worker, through the Publish Monitor, as it converts the CAD Document to a viewable.

If successful, a thumbnail image is generated.

5. Select the thumbnail image to open the new Viewable.

If the object fails to publish, refer to [Troubleshooting the Publishing Agent](#).

Test 3: Creating a Representation from Local Data

To perform this test, you will need CAD part data loaded on the Windchill system. This data must be in its native state, meaning that it has not already been converted to a viewable (published). You will use this data to test the publishing capabilities.

1. Open Windchill in a Web browser.
2. Create an empty WTPart, which will serve as a placeholder for the CAD part.
3. Open the information page for the WTPart.
4. Open the Representation wizard by clicking on the **Display list of representations** button in the corner of the visualization panel located in the upper-right corner of the information page.
5. Select **New representation** to create a representation to associated with WTPart.
 - In Step 1, assign the representation a name.
 - In Step 2, select “Local data” and then select the CAD worker data type from the **CAD type/name** drop-down list.
 - **Browse** to a CAD part on the local system.

These steps will set the action for the Windchill server to retrieve a CAD part (for example, a Pro/ENGINEER part), and the properly configured CAD worker (PROE) is triggered to create the viewable and populate the representation.

6. Complete the remaining steps described by the wizard accepting the defaults.
7. Upon finishing, you are presented with a link to the **Publish Monitor** where you can observe the viewable conversion take place.

If the CAD worker has been properly configured and the data is valid, the Publisher will conclude successfully and a thumbnail link is presented. Click the thumbnail link to view the image.

If the object fails to publish, refer to [Troubleshooting the Publishing Agent](#).

Using WVS with Multiple Background Servers

You can configure multiple background method servers, and have different "queue groups" process in each background method server. For specific instructions on doing this, refer to the *Windchill Performance Tuning Guide*.

By default, the WVS publishing queues, the CADAgent, and the Loader execute in the background method server that is processing the queue group "default". This requires no further configuration.

However, if you want to optimize performance and scalability, you can dedicate a background method server to WVS processing. For example, if the queue group "wvs" is being used, you must change the WVS queues to specify that queue group using the Queue Manager as a Site Administrator, and update Background MethodServer startup command (for example, `wt.manager.cmd.BackgroundMethodServer`) in `wvs.properties` to include the setting:

```
wvs.queueGroup=<group_name>
```

where, in this example, the `<group_name>` is `wvs`.

This property setting instructs the CADAgent and loader to start in the background method server, which is processing the queue group "wvs". This is the same background method server that the publishing queues are running in.

Configuring the CAD Agent in a Windchill Cluster

The WVS CAD Agent has a "host" parameter that can be used for cluster configurations. This parameter should be manually added to the [agent] section of the agent.ini file on the slave machines in the cluster (for example, those machines not running the background method server). The value of the host parameter would be the hostname of the machine that runs the background method server, and hence the CADAgent.

The "host" parameter enables the CAD Agent Administration Pages, running on Foreground MethodServers, to connect to the Background MethodServer (BGMS), which might be running on a different host machine, by identifying the hostname of the machine running the BGMS.

Note: The property **publish.tempuploadir** is set to a common directory between all the method servers in the cluster, in the case of creating representations from local data only.

Generating Viewables and Thumbnails in Clustered Environments

All publishing and thumbnail conversion is processed using background queues. In a cluster, only one Windchill server processes the queue, using a dedicated MethodServer called the Background MethodServer. As a result, in a clustered environment, all viewable generation is done through the Background MethodServer.

By default, the Thumbnail Generator is executed by the CAD worker on the worker machine, before the viewables are retrieved (along with the thumbnail) by the CAD Agent. However, if this is disabled, and unless you are using remote thumbnail generation, the thumbnail generation is also done from the Background MethodServer, as the CAD Agent runs on the method server that is processing queues.

ProductView Collaboration in a Clustered Environment

When a ProductView collaboration session is initiated, the MethodServer that handles this client request starts a dedicated process for managing the meeting. The command syntax for starting this process is defined by collaboration.server in wvs.properties, and in our example, is called "pview_collaboration".

All users participating in the collaboration session must access pview_collaboration directly through TCP/IP. The ranges of port for this can be restricted by passing the "-p" and "-m" options when starting the process. For example, the following option for collaboration.server would specify that the processes only listen to ports between 1000 and 1999:

```
collaboration.server=$(wt.home)\\bin\\pview_collaboration.exe -p 1000 -m 1999 -i 600
```

By default, the pview_collaboration process broadcasts itself as running on the host on which it was started. For secure configurations, this host may not be

accessible. In this case, you can use the "-a" option to indicate a specific hostname or IP address, which the clients can resolve.

For example, the following option for `collaboration.server` would specify that the processes identify themselves as running on host "windchill.company.com":

```
collaboration.server=$(wt.home)\\bin\\pview_collaboration.exe -a  
windchill.company.com -p 1000 -m 1999 -i 600
```

Note: Connections to an externally accessible machine that is not running collaboration must be forwarded to the machine that is running collaboration.

In addition, the MethodServer needs to access the `pview_collaboration` process through a socket. When `pview_collaboration` starts, it generates a random port number that is used by the MethodServer, and inserts this port number in a temporary file created in `collaboration.tempdir`, as defined in `wvs.properties`. When it needs to connect to this process, the MethodServer will read this file to obtain the IP address and port number to connect to. During this process:

- The communication is one-way: from the MethodServer to the `pview_collaboration` process.
- The port number generated is random and cannot be restricted in the current releases of Windchill.

Suggested configuration for a Windchill cluster

In a Windchill cluster, there are several physical hosts that run MethodServers. Users access a virtual URL and are then forwarded to one of the hosts running a Windchill instance.

In the sample architecture below, the generic host is "windchill.company.com", which is a router that forwards requests on the cluster nodes, "node1" and "node2". When a user starts a meeting, this starts a `pview_collaboration` process on one of the nodes, and all users connecting to this meeting will be forwarded to the proper host:

- The `pview_collaboration` process on each node is identified as being on the router ("-a windchill.company.com")
- Each node uses different ports for the collaboration processes, for example:
 - on node1: "-p 1000 -m 1999" in `collaboration.server`
 - on node2: "-p 2000 -m 2999" in `collaboration.server`
- The router at "windchill.company.com" is configured to forward the requests on ports 1000-1999 to node1, and request on ports 2000-2999 to node2.

Suggested configuration to run pview_collaboration on a separate machine

You can also run pview_collaboration on a separate host than the MethodServer. To do this, replace the value of collaboration.server with the appropriate program/script. This needs to be completely transparent to the MethodServer, so it is necessary that:

- The pview_collaboration is visible to end users (for example, by using the "-a" option, or running it on a machine that is already accessible).
- The MethodServer can connect to the pview_collaboration host on any port (one-way traffic).
- The temporary folder collaboration.tmpdir points to the same physical location both from the MethodServer and pview_collaboration perspective, which requires a service like NFS.

Making Thumbnails Viewable for Microsoft Word or Excel Files

In order to display thumbnails for Windchill viewables containing Microsoft Word or Excel files, you must modify the Word or Excel file.

Note: This technique applies only to Microsoft Office files being converted to PDF format by the Microsoft Office worker. It does not apply when those files are not being published.

1. Open the file in Word or Excel, and select **File > Properties**.
2. Check the **Save preview picture** option on the **Summary** tab.
3. Save the file, and update it or check it back into Windchill.

After completing this procedure, a thumbnail of the file should now be displayed when viewing the details page of this file in Windchill.

10

File Synchronization

This chapter provides the instructions to configure file synchronization-capable workers to assist the Visualization server in publishing a viewable.

At the Windchill 9.0 release, the following workers are file synchronization-capable: Pro/ENGINEER, Unigraphics, SolidWorks, Autodesk Inventor, Arbortext.

Topic	Page
Before You Begin.....	10-2
Using a File Synchronization-Capable Worker to Download Files	10-2
Publishing Wildfire Assemblies as Positioning Assemblies	10-8

Before You Begin

For file synchronization to work properly, you must have a file synchronization-capable worker compatible with Windchill 9.0. You can use older versions of your authoring applications for designing and engineering. However, the CAD worker must be configured to use a higher version of the authoring application for publishing and generating viewables on the Windchill server.

Using a File Synchronization-Capable Worker to Download Files

These instructions apply if you are using one of the file synchronization-capable workers listed at the beginning of this chapter. This configuration allows Windchill Visualization Services to create a viewable by using the worker to download the original native CAD file from Windchill.

The following is an example of the configuration changes needed to tell Windchill Visualization Services to use file synchronization for publishing data. This general concept applies to all file synchronization-capable workers.

Note: In Windchill 9.0, the out-of-the-box `wvs.properties` file reflects the use of file synchronization by default. Nonetheless, please read through the following steps and note the need to create an `auth.properties` file - otherwise, publishing will not be successful. Be sure to note the differences in the command string for each of the file synchronization-capable workers.

1. Append the `useworkerdownload` variable to the `publish.cadconvert.<worker_name>` variable in the `wvs.properties` file using `xconfmanager`. This tells the publisher to use the worker for the file download (`useworkerdownload`) and that the username and password to give the worker can be found in the `auth.properties` file in the Windchill directory.

From a Windchill shell, execute the following commands:

- a. To display the current value of the `publish.cadconvert.<worker_name>` variable:

```
xconfmanager -d publish.cadconvert.<worker_name>
```

- b. Append the `useworkerdownload` variable definition to the `publish.cadconvert.<worker_name>` variable. Using the information results of the display command, append to that:

```
useworkerdownload=$(wt.home) $(wvs.dir.sep) auth.properties
```

Use a comma (,) separator when appending the new value. For example:

For Pro/ENGINEER:

```
xconfmanager -s
publish.cadconvert.PROE=com.ptc.wvs.server.publish.CadConvertPROE,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties
-t <Windchill>\codebase\wvs.properties -p
```

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

For Autodesk Inventor:

```
xconfmanager -s
publish.cadconvert.INVENTOR=com.ptc.wvs.server.publish.CadConvertINVENTOR,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties
-t <Windchill>\codebase\wvs.properties -p
```

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

For SolidWorks:

```
xconfmanager -s
publish.cadconvert.SOLIDWORKS=com.ptc.wvs.server.publish.CadConvertSOLIDWORKS,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties
-t <Windchill>\codebase\wvs.properties -p
```

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

For Unigraphics:

```
xconfmanager -s
publish.cadconvert.UG=com.ptc.wvs.server.publish.CadConvertUG,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties,assytpe=topdown,repiba=additional_reps
-t <Windchill>\codebase\wvs.properties -p
```

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

For Arbortext:

```
xconfmanager -s
publish.cadconvert.ARBORTEXT=com.ptc.wvs.server.publish.CadConvertARBORTEXT,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties,types=PDF HTML WEB POSTSCRIPT HTMLHELP SGML XML RTF DMP 3D,defaultttype=PDF
-t <Windchill>\codebase\wvs.properties -p
```

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

2. The publish.cadconvert.<worker_name> variable edited in the previous step references the auth.properties file. At this time, the file does not exist and must be created. Complete the following steps:
 - a. In the above example, WVS will look for the auth.properties file in the Windchill directory, but may be placed anywhere. For security reasons, the file should not be located in the codebase directory.
 - b. Create the auth.properties file using a text editor of choice.

- c. Add an authentication variable that includes a user name and password value. This will be the only entry in this file. The syntax is:

```
auth=<username>:<password>
```

Replace *<username>* and *<password>* with valid values. The specified user name must have read access to the Windchill objects to be published through Windchill Visualization Services and it must have the authorization to download the content from Windchill.

- d. Save your changes and close the auth.properties file.

3. Restart the Windchill server.

Configuring the CAD Worker

Previous releases of Windchill provided the ability to synchronize the values of properties shared between Pro/ENGINEER data and Windchill during viewable generation. This action is referred to as filesync. Windchill 9.0 extends filesync support during viewable creation to a number of other CAD tools.

The following three components need to be installed on the CAD worker system:

- CAD application
- ProductView Adapter – (refer to the *ProductView Adapters Installation and Configuration Guide* for installation information)
- Windchill Workgroup Manager – (refer to the Installing the Windchill Workgroup Manager chapter of the *Windchill Workgroup Manager Administrator's and User's Guide* for installation information)

Note: The following section applies to the Autodesk Inventor, SolidWorks and Unigraphics CAD workers only.

Post-Install Configuration of the CAD Worker on Windows

After the Windchill Workgroup Manager is installed, two system-level environment variables need to be defined as follows:

1. Copy the PTC_WGM_HOME environment variable definition from the user-level environment settings to the system-level environment settings. The value for this setting is the directory where the Windchill Workgroup Manager is installed. Remove the PTC_WGM_HOME environment variable definition from the user-level environment setting.
2. Create a new system-level environment variable called PTC_WF_ROOT. Set the variable value to a directory to which the System-level account has write access.
3. Restart the worker machine after defining these two environment variables to allow the new settings to take effect in the system environment.

Register the CAD application with the Windchill Workgroup Manager. Refer to the *Windchill Workgroup Manager Administrator's and User's Guide* for information on how to register the CAD application. Do not register the Windchill server with the Windchill Workgroup Manager.

Configuring the GS Worker Daemon Service Properties

For Unigraphics:

This service must have system access in order for filesync to function properly.

1. Edit the GS Worker Daemon service properties.
2. In the GS Worker Daemon Properties dialog, select the Log On tab.
3. Set Log on as to **Local System Account** and click **OK**.
4. On the Services Panel, select Stop if the service is running, and then Restart the service. If this is not done, the change you just made will not take effect, and the old environment settings will still be in effect.

For SolidWorks and Autodesk Inventor:

This service must have desktop access in order for the filesync to function properly.

1. In the GS Worker Daemon Properties set up window, select the Log On tab.
2. Set Log on as to **Local System Account**. Select the Allow system to interact with desktop check box, and click **OK**.
3. On the Services Panel, select Stop if the service is running, and then Restart the service. If this is not done, the change you just made will not take effect, and the old environment settings will still be in effect.

Post Install Configuration of the CAD Worker on UNIX

The CAD worker requires access to a valid X server display. Through a system shell on the X server system, allow display access to the CAD worker machine through the following command:

```
xhost +<CAD worker machine name>
```

Edit the CAD worker script to set the display to that X server system with the following line:

```
setenv DISPLAY <X server machine name>:0
```

For the following example, the worker system and the X server display are the same system.

Below is the command issued through a shell on the X server system that permits access from the CAD worker machine:

```
xhost +crescendo.ptcnet.ptc.com
```

The following is a sample modified proeworker script file with the X server display setting defined:

```
#!/bin/csh -f
echo Launching proe2pv in worker mode - see log file
proeworker.log
setenv DISPLAY crescendo.ptcnet.ptc.com:0
setenv LD_LIBRARY_PATH ""
setenv PVIEW_HOME "/viz/productview_adapters"
setenv PVIEW_WORKING_DIR `pwd`
cd "/viz/workers/qaxw6000_PDMPJL90/wildfire"

echo Launching proe2pv in worker mode > proeworker.log
"$PVIEW_HOME/bin/proe2pv" -vc -vL
"/viz/workers/qaxw6000_PDMPJL90/wildfire/proeworker.log" -EW -
CSsonata 2573 -r
"/viz/workers/qaxw6000_PDMPJL90/wildfire/proe2pv.rcp" $*:q &
```

Configuring the ProductView Adapter for HTTPS

Because the worker runs in the background, user dialogs that arise during communication with the server are to be avoided. In an HTTPS environment, for example, certificates should be pre-accepted and loaded in order for the worker system to avoid acceptance dialogs.

Accepting Certificates on Windows

For the Windows CAD worker to publish data, the GS Worker Daemon service must be configured with the specific logon information of a user who can log on to the local host and accept certificates from the Windchill server. To accomplish this, do the following:

1. Open the Windows Service Panel, and click **Start > Settings > Control Panels**. Once the Control Panel window opens, double-click the **Services** icon. The **Services** window opens.
2. View the properties of the **GS Worker Daemon**.
3. Return to the **Services** window and click **Stop** to stop the Service.
4. Click the **Log On** tab.
5. Choose **Select This Account** and enter the user logon and password information.
6. Click the **General** tab and start the Service.
7. Open Pro/ENGINEER Wildfire or the Windchill Workgroup Manager and browse to the Windchill server.
8. Install (Accept) the security certificate for the secure site.
9. Click **Exit** to exit Pro/ENGINEER Wildfire or the Windchill Workgroup Manager.

10. Refer to the "Configuring the CAD Worker" section for the steps to configure the CAD worker on the server.

Accepting Certificates on UNIX

A UNIX worker only publishes on HTTPS servers if you log into the worker machine to start an X server and an X display, and then disable display access control for it by executing the following command:

```
xhost +
```

In addition, the CAD worker script file must include the environment variable to display the machine name. For example:

```
setenv DISPLAY <machine name>:0
```

The following is a sample modified proeworker file:

```
#!/bin/csh -f

echo Launching proe2pv in worker mode - see log file
proeworker.log

setenv DISPLAY crescendo.ptcnet.ptc.com:0

setenv LD_LIBRARY_PATH ""

setenv PVIEW_HOME "/disk2/ndm-viz/local/productview_adapters"

setenv PVIEW_WORKING_DIR `pwd`

cd "/disk2/ndm-viz/local/workers/qaxw6000_PDMPJL90/wildfire"

echo Launching proe2pv in worker mode > proeworker.log

"$PVIEW_HOME/bin/proe2pv" -vc -vL "/disk2/ndm-
viz/local/workers/qaxw6000_PDMPJL90/wildfire/proeworker.log" -
EW -CSsonata 2573 -r "/disk2/ndm-
viz/local/workers/qaxw6000_PDMPJL90/wildfire/proe2pv.rcp" $*:q
&
```

A UNIX worker must be configured to accept certificates from the Pro/ENGINEER Wildfire or the Windchill Workgroup Manager browser. To accomplish this, follow these steps:

1. Open Pro/ENGINEER Wildfire or the the Windchill Workgroup Manager on the CAD worker system.
2. Browse to the Windchill server using the same logon information as the CAD worker on the server.
3. Select the option for **Remember this certificate permanently** and click **Continue**.
4. Click **Exit** to exit Pro/ENGINEER Wildfire or the Windchill Workgroup Manager.

Publishing Wildfire Assemblies as Positioning Assemblies

This section explains how to configure visualization to publish Wildfire assemblies as positioning assemblies.

A *positioning assembly* is a high-level assembly that is used to fix the relative location (position) of its child assemblies. As a result, there can be no assembly features used between these assemblies. In addition, if parametric positioning is used, ensure that changes in any child components do not impact the position of components within the assembly being published as a positioning assembly.

When an assembly is converted as a positioning assembly, ProductView branch links are used to reference the child assemblies' representations, rather than converting the data in the context of the parent assembly. When the parent assembly representation is created with a latest configuration spec, this allows the latest child assemblies to be automatically displayed for the parent, without the parent needing to be republished.

To publish a Wildfire assembly as a positioning assembly:

1. Specify that the EPMDocument should be published as a positioning assembly. You can do this using the Publish and Thumbnail Control user interface from the Utilities page for the appropriate Product/Project/Org/Site.

Or, you can specify a list filename, IBA name, or custom method to identify which EPMDocuments should be published as positioning assemblies, using the methods described next:

```
<Property default="" name="publish.positioningassembly.filename"/>
```

```
<Property default="" name="publish.positioningassembly.ibaname"/>
```

```
<Property default="" name="publish.positioningassembly.filtermethod"/>
```

- If you specify a list filename, that file should contain a list of EPMDocument CADNames that will be positioning assemblies. If the EPMDocument type does not define a CADName, the EPM number displays instead.
- If an EPMDocument is not specified in the file, the IBA name (if specified) will still be checked. If you specify an IBA name, the IBA should be Boolean, where a return of Boolean.TRUE indicates that the EPMDocument is a positioning assembly, and FALSE indicates that it is not. If the IBAname is not present, that EPMDocument would not be processed as a positioning assembly.

Additionally, an administrator can specify on the New Representation wizard that a representation be created by publishing an appropriate EPMDocument as a positioning assembly. This will override the other methods of specifying a positioning assembly.

- If you specify a custom method, only that method will be used. In this case, the property value format should be:

classname/methodname

with the following signature:

```
public static Boolean methodname(EPMDocument d)
```

where a return of Boolean.TRUE indicates that the EPMDocument is a positioning assembly, and FALSE indicates that it is not.

2. Specify whether to publish the children when publishing a positioning assembly, if there was no representation for the positioning assembly to use.

```
<Property default="false" name="publish.positioningassembly.publishchildren"/>
```

The property can be any of the following values:

- false - no publishing of children will take place.
- true or latest - If a positioning assembly is being created with a latest configuration spec, the children will be published if required.
- default - If a positioning assembly is being created as the default representation, the children will be published if required.
- all - If any positioning assembly is being created, the children will be published if required.

11

Arbortext Publishing Engine (APE) Worker and Publishing

This chapter provides instructions for configuring the Arbortext Publishing Engine (APE) Worker, configuring the WVS publisher for the Arbortext Authoring Application, and defining and loading publishing rules.

Topic	Page
Configuring the APE Worker.....	11-2
Configuring Publishing for Arbortext Authored Dynamic Documents	11-7

Configuring the APE Worker

Before You Begin

The APE Worker is automatically installed with Windchill. This worker publishes Dynamic Documents created using the Arbortext Editor into standard output formats such as PDF or HTML.

This chapter defines the steps to configure the APE Worker.

Prerequisites

Prior to configuring the APE Worker, you should have:

- Installed and configured the APE software and relevant licenses.
- The URL to a working instance of an Arbortext PE (this will be the `ape.server` value in the `wvsape.properties` file).
- Installed an Arbortext-compatible Windchill solution.

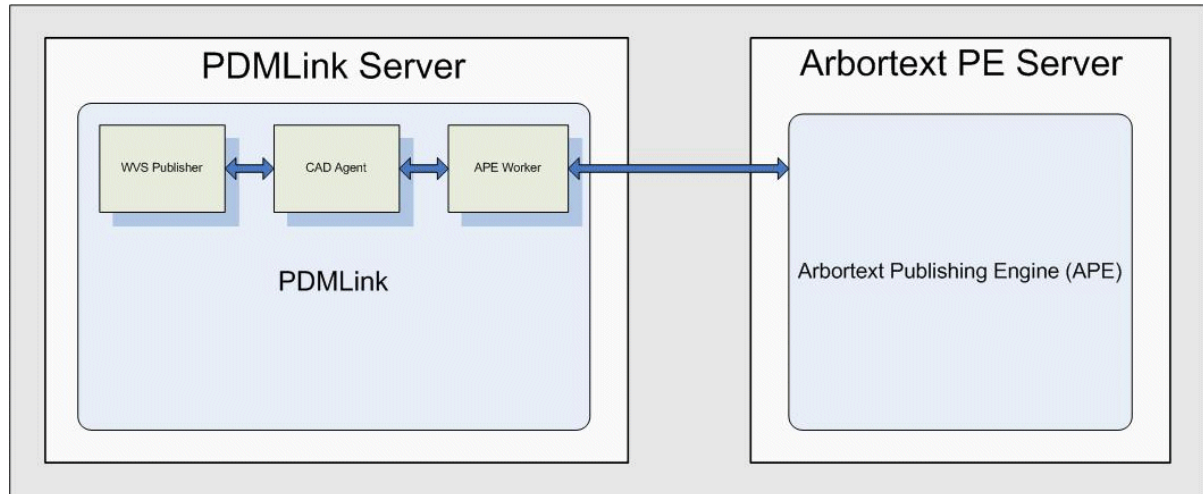
Naming Conventions

For the purposes of this documentation, the following directory name conventions will be used:

- *<Windchill>* — The Windchill installation directory.
- *<APEWorker>* — The directory that stores the worker configuration files: `arbortextEngineWorkerstart.bat` and `wvsape.properties`.

Process Overview

The diagram below illustrates the relationship between the Windchill server, Visualization Services, the CAD Agent, APE Worker and APE, to provide an understanding of what needs to be configured.



The following process is used to complete the APE Worker configuration:

1. Configure the APE Worker.
2. Configure the CAD Agent (agent.ini) using the CAD Configuration Wizard.
3. Verify that the CAD Agent, APE Worker, and AP Engine are all communicating properly.

A wvsape.properties file and a worker batch file are used to configure the APE Worker. Detailed descriptions of these files and their parameters are provided in the subsequent sections.

Configuring the APE Worker

The APE worker configuration involves a batch file that executes the APE Worker and a wvsape.properties file that is used to specify the APE Worker settings. PTC recommends that these configuration files reside in a new working directory that you create to preserve the installed worker files. This section explains how to create the batch and wvsape.properties files to configure the worker.

Follow these steps to configure the APE Worker:

1. Create a new *<APEWorker>* directory on the Windchill server, for example, C:\PTC\APEWorker.

Note: Remote APE Worker configuration is currently not supported. This is not to be confused with having a remote APE Server. The APE Worker resides with the Windchill Server, and is used to process the requests/reponses to and from the APE Server. Refer to the previous diagram for a visual depiction of this configuration.

2. Locate the sample configuration files in the %WT_HOME%\codebase\com\ptc\wvs\server\workers directory. The sample file is called **wvsape.properties** and the batch file name is **arbortextEngineWorkerstart.bat**.

Note: This procedure assumes you are using Windows. If you are on UNIX, you use the sample shell script (arbortextEngineWorkerstart.sh).

3. Copy the sample files to the new directory you created in step 1. Do not edit or move the files from their original location.
4. Using a text editor of choice, edit the worker batch file. This batch file is used as the APE Worker executable command when you configure the CAD Agent in a subsequent step.

The sample worker batch file is shown below, followed by descriptions of the settings.

```
@echo on
set WORKER_DIR=C:\ptc\apeworker
set DEBUG="-D"
set PORT="5600"
set HOST="localhost"
set TYPE="APE"
set CMDCLASS="com.ptc.wvs.server.workers.ArbortextEngineWorker"
```

C:

```
cd "%WORKER_DIR%"
```

```
java -Dwvs.ape.properties=%WORKER_DIR%\wvsape.properties
com.ptc.wvs.server.cadagent.GenericWorker %DEBUG% -PORT %PORT%
-HOST %HOST% -TYPE %TYPE% -CMDCLASS %CMDCLASS% -DIR %WORKER_DIR%
-LOG "worker_"
```

- set WORKER_DIR=C:\PTC\APEWorker - This value is critical and must be set to the <APEWorker> directory created in step 1.
- set DEBUG="-D" - This value is reserved for debugging and should not be changed.
- set PORT="5600" - This value can be changed, but it is not recommended to do so. If you change this value, you must ensure that it matches the "port" value in the [agent] section of the agent.ini file. 5600 is the default value.

- set HOST="localhost" - This value can be left as "localhost", but should match the hostname of the machine where the worker resides, which, for the APE Worker, will be the Windchill machine.
 - set TYPE="APE" - This value should not be changed.
 - set CMDCLASS="com.ptc.wvs.server.workers.ArbortextEngineWorker" - This value should not be changed.
5. Save your changes to the batch file in the *<APEWorker>* directory.
 6. Edit the wvsape.properties file as needed. A sample properties file is shown below, followed by descriptions of the settings.

```
ape.server=http://localhost/e3/servlet/e3
ape.worker.logfile.dir=C:\\ptc\\Windchill\\logs\\arbortext\\
ape.verbose=false
```

Note: The ape.server URL you provide should correspond to the URL associated with your current Arbortext release. The full URL should be specified.

- ape.server is the URL to the installed Arbortext PE server. It is required that you define this value.
- ape.worker.logfile.dir is the directory that stores the Arbortext worker log file. You should change this directory to match the location where you keep your log files.
- ape.verbose is a true/false switch for verbose mode.

Caution: The wvs.properties file contains a setting (publish.usesPublishRules=ARBORTEXT). If this property is deleted or ARBORTEXT is removed, WVS does not use Publishing Rules for Arbortext authored DynamicDocuments. *This is not recommended.*

7. Save your changes into the *<APEWorker>* directory and close the properties file.

UNIX

For UNIX, you create a script file instead of a batch file.

Configuring the APE Worker Monitor

Note: Using the Worker Monitor is recommended if you have access to a ProductView Adapters CD. If you do not have a ProductView Adapters CD, then you can ignore this section.

The Worker Monitor provides more robust communication with the worker by monitoring based on its own timeouts, rather than being completely dependent on

the worker for reporting errors. Additionally, the worker logs are also made available through the web browser via the CAD Agent UI when the worker monitor is configured.

1. Similar to configuring the worker in the previous section, locate and copy the `arabortextEngineMonitorstart.bat` from `%WT_HOME%\codebase\com\ptc\wvs\server\workers` to your `<APEWorker>` directory.

Note: This procedure assumes you are using Windows. If you are on UNIX, you use the sample shell script (`arabortextEngineMoniterstart.sh`).

2. In your `<APEWorker>` directory, edit the `arabortextEngineMonitorstart.bat` with a text editor of your choice.

The sample monitor batch file is shown below, followed by descriptions of the settings.

```
@echo on
set WORKER_DIR=C:\ptc\apeworker
set PORT="5600"
set HOST="localhost"
set TYPE="APE"
set PVIEW_HOME=C:\ptc\productview_adapters

C:
cd "%WORKER_DIR%"
"%PVIEW_HOME%\i486_nt\obj\workermonitor" -UH -s "%WORKER_DIR%\
arabortextEngineWorkerstart.bat" -vt -EW %TYPE% -CS%HOST% %PORT%
%1 %2 %3 %4 %5 %6 %7 %8 %9
```

- `set WORKER_DIR=C:\PTC\APEWorker` - This value is critical and must be set to the `<APEWorker>` directory referenced in Step 1.
- `set PORT="5600"` - This value can be changed, but it is not recommended to do so. If you change this value, you must ensure that it matches the "port" value in the [agent] section of the `agent.ini` file. 5600 is the default value.
- `set HOST="localhost"` - This value can be left as "localhost", but should match the hostname of the machine where the worker resides, which, for the APE Worker, will be the Windchill machine.
- `set TYPE="APE"` - This value should not be changed.
- `set PV_HOME=C:\ptc\product_adapters` - This value must be set to the location of the installed `productview_adapters` directory.

3. Save your changes to the batch file in the `<APEWorker>` directory.

This batch file is used instead of the worker batch file in the Execute Command of the CAD Agent (see next section).

UNIX

For UNIX, you create a script file instead of a batch file.

Configuring the CAD Agent

To configure the CAD Agent for the APE Worker, please refer to the *Configuring the Publishing Agent* chapter in this guide, and follow these guidelines for APE:

- Select data type **ARBORTEXT**.
- The **HOST** should be the name of the server.
- The **CAD Worker Location** should remain **Windchill Server**.
- The **Execute Command** should point to the batch file you created when configuring the APE Worker, in a previous section.

Note: If you are using the worker monitor, then the Execute Command should point to the monitor batch file instead of the worker batch file.

- The number of **Max Instances** in the **CAD Worker Execution Command** page of the **CAD Configuration Wizard** must match the **maxSubprocesses** value defined for the APE pool designated to WVS. Refer to the APE documentation for information about pools and maxSubprocesses.

Configuring Publishing for Arbortext Authored Dynamic Documents

This section describes the configuration process for publishing Dynamic Documents that were created using the Arbortext Editor.

The WVS framework allows the publishing of Dynamic Documents authored using Arbortext Editor. To configure publishing of these documents, you must configure the WVS Publisher for the Arbortext Authoring Application using WVS properties. You can then define and load rules for publishing the documents. These are called *publishing rules*.

Overview

Since its introduction at Windchill 6.0, Windchill Visualization Services (WVS) has provided a framework that allows the publishing of EPMDocuments to create a Representation for visualization. This publishing is accomplished using a worker, and is managed by the CAD Agent. Dynamic Documents are a soft-type of EPMDocument, so the framework supports them in the same manner.

WVS sends a request to the Arbortext Publishing Engine based on specified publishing rules. These rules specify when the Representation is created, such as upon check-in, and the desired Representation format, such as PDF and HTML. Once the Representations are created, WTDocuments can be created and linked to

each Representation of the Dynamic Document with a PublishedContent link. The new WTDocument can be used as part of a downstream business process.

Configuring the WVS Publisher for the Arbortext Authoring Application

In order for the Arbortext Publishing Engine to download input files from Windchill and subsequently upload the output file, an authentication file is needed. This file is typically called *auth.properties*.

Note: The *wvs.properties* file, located in the %WT_HOME%\codebase\ directory, contains the following property setting, which defines the location of the *auth.properties* file. If you change the location of this file, you must edit this property accordingly.

```
publish.cadconvert.ARBORTEXT=com.ptc.wvs.server.publish.CadConvertARBORTEXT,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties,types=PDF HTML WEB POSTSCRIPT HTMLHELP SGML XML RTF DMP 3D,defaultttype=PDF,
```

To configure the *auth.properties* file, follow these steps:

1. Create the *auth.properties* file in your <Windchill> root directory (also known as %WT_HOME%) using a text editor of choice
2. Add an authentication variable that includes a user name and password value. This will be the only entry in this file. The syntax is :

```
auth=<username>:<password>
```

3. Replace <username> and <password> with valid values. The user should have enough privileges to read and download all Dynamic Document structures stored in Windchill that need to be published.
4. Save your changes and close the *auth.properties* file.
5. In the **publish.cadconvert.ARBORTEXT** property described previously in this section, a default location of \$(wt.home)\$(wvs.dir.sep)*auth.properties* is provided for the *auth.properties* file. If you use a different location for this file, be sure to edit this property value to reflect the updated path. You must use the *xconfmanager* utility to make this change:

```
xconfmanager -s  
"publish.cadconvert.ARBORTEXT=com.ptc.wvs.server.publish.CadConvertARBORTEXT,useworkerdownload=$(wt.home)$(wvs.dir.sep)auth.properties,types=PDF HTML WEB POSTSCRIPT HTMLHELP SGML XML RTF DMP 3D,defaultttype=PDF" -t <Windchill>\wvs.properties  
-p
```

6. Restart the Windchill servers.

Publishing Arbortext Authored Dynamic Documents

Publishing of Dynamic Documents is controlled by using publishing rules. Refer to the *Windchill Business Administrator's Guide* for details on using publishing rules.

A

Windchill Runtime Environment

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, and 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.

The Java 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), which is the Java 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 (for example, 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.

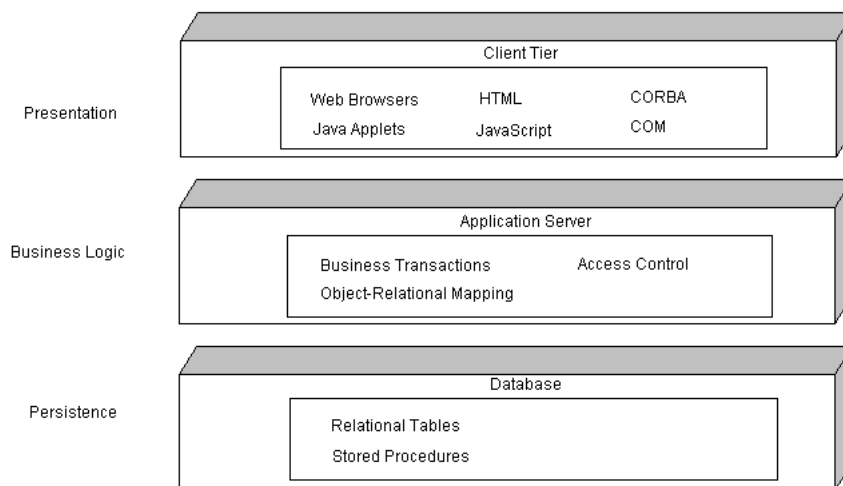
Internationalization is the process of designing and developing an application that can be adapted to the culture and language of a locale other than the one for which it was originally developed. Java facilitates this task by providing classes that convert dates and numbers to formats conforming to 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.

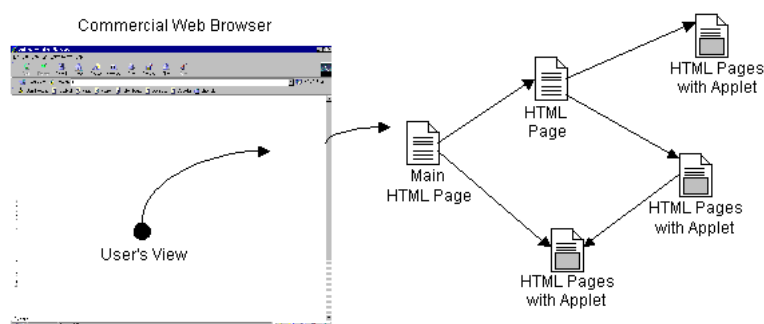
The ability to display HTML pages, although adequate for simple applications, does not provide enough functionality for all aspects of complex information authoring applications. Therefore, Windchill requires a browser capable of hosting Java applets based on the Java runtime and base classes. Two popular 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.

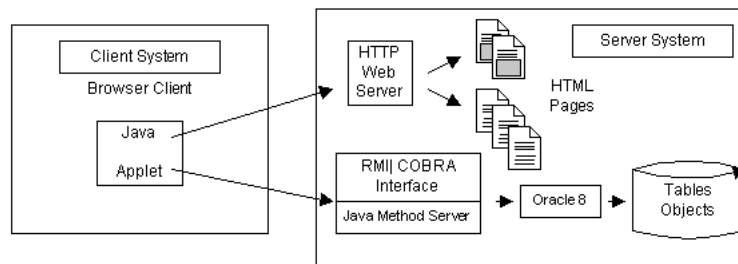
Many simple accesses to the system may use only HTML presentation, with HTML form data serving as input. However, the typical client session requires that applet tags (used to carry out complex user interactions involving complex data) be embedded in these HTML responses.

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.

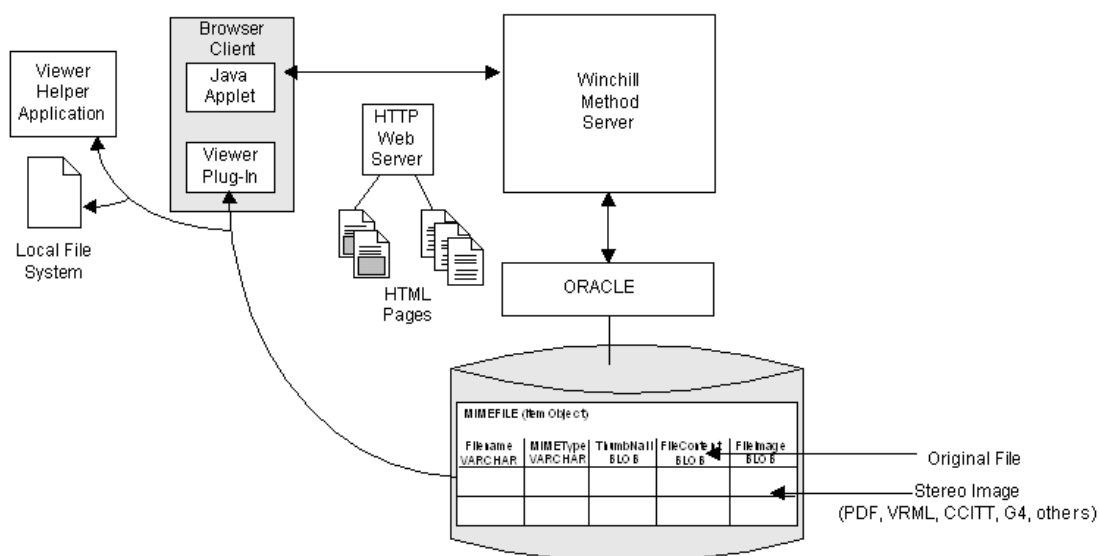
File Upload Using RMI

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.

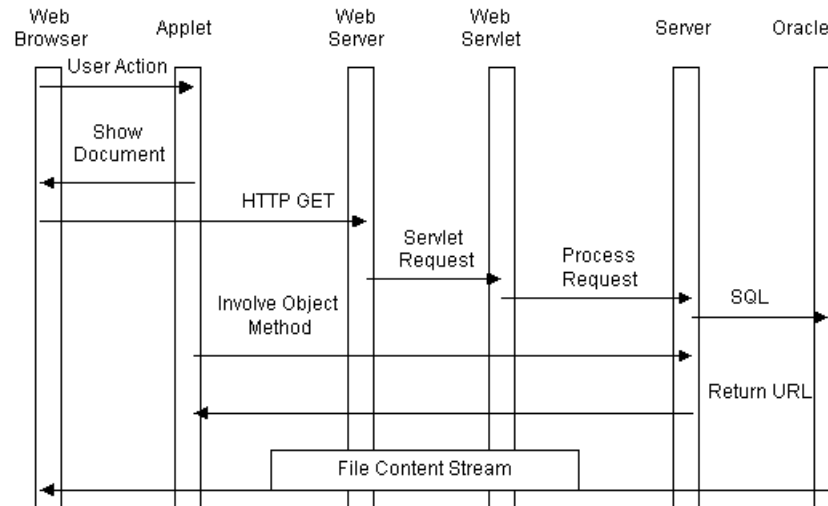
Be aware that this upload architecture addresses limitations in some browser's Java HTTP classes. The HTTP upload procedure is still available, but PTC does not recommend using it for content upload from an applet. Downloads from the Windchill server via HTTP do not exhibit the same limitations as uploads, and downloads can still use the HTTP architecture described below.

File Transfer Using HTTP

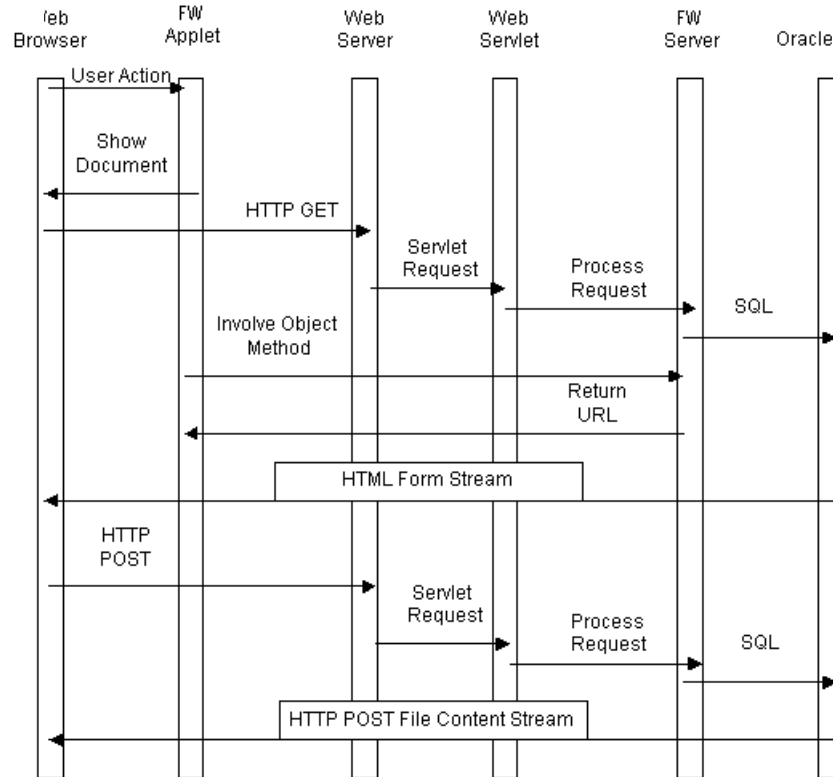
To leverage the Web browser's ability to view, save, and operate on a diverse set of content types, it must be possible to stream file content from the Windchill system to the browser through the HTTP gateway. As shown in the following figure, requests for file transfer are encoded into appropriate HTTP 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, 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 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 [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.

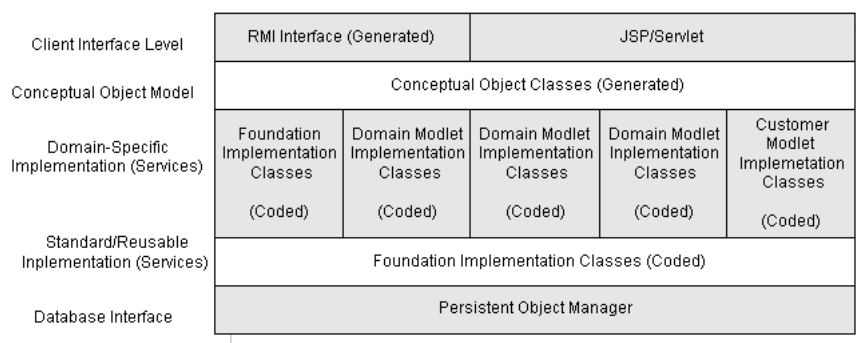
System Management

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 Background Queues and Related Properties](#).

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.

For additional information, see [Managing Logging](#).

Database Components

Following are the data 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.



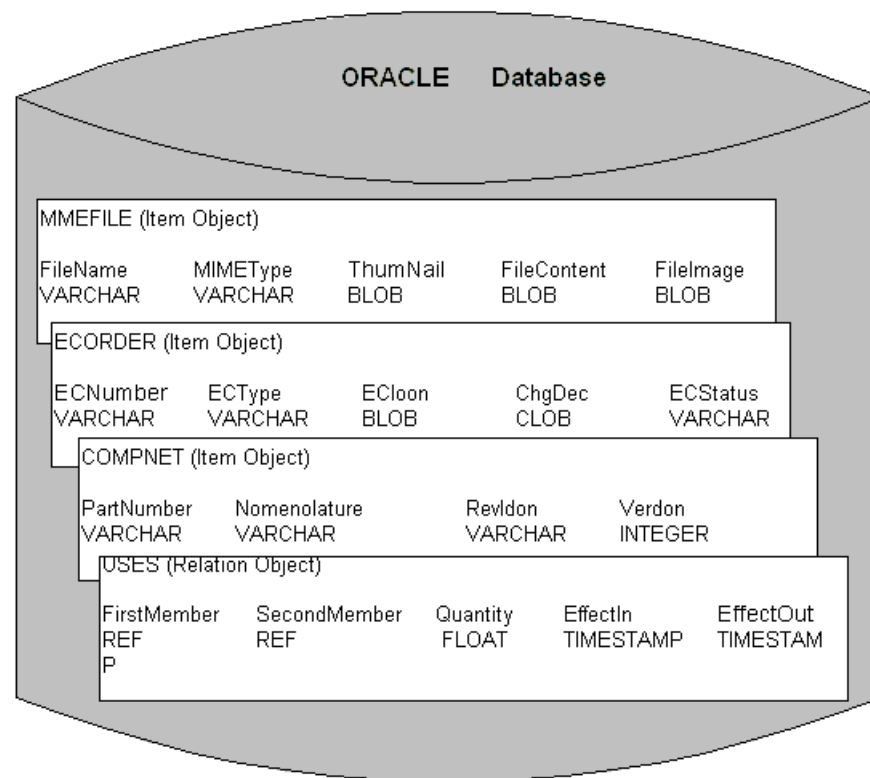
Caution: Windchill uses the object relational features of the database server to store data objects. In order to maintain the integrity of the associations among stored objects, users and administrators should avoid using vendor tools to directly manipulate database data. Directly changing data in the 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.

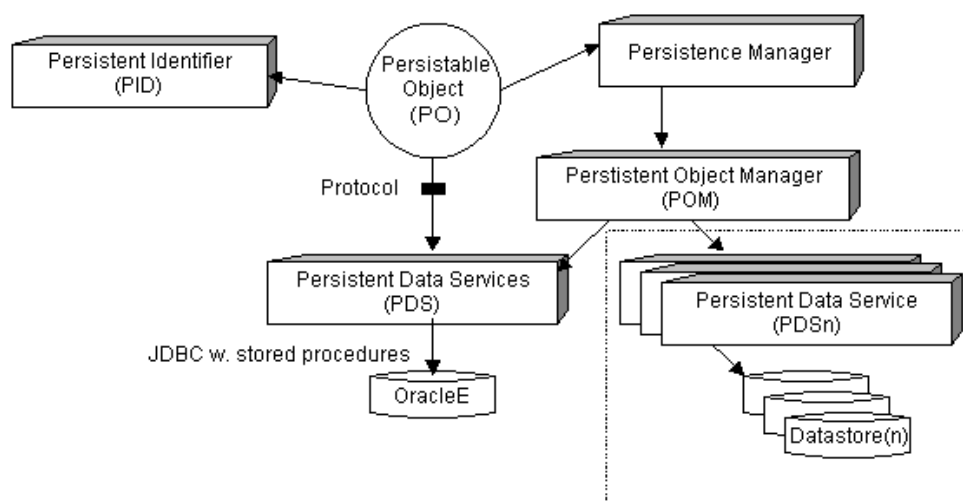
Storing File Content as Large Objects

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 the Windchill indexing capabilities. For information about the maintenance of collections with Windchill Index Search, see *Windchill Installation and Configuration Guide - Advanced*. 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 Windchill Index Search 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 Windchill Index Search 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 indexes and leverage Windchill Index Search 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 Windchill Index Search 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 Windchill Index Search 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 Windchill Index Search 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 Windchill Index Search 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 Windchill Index Search for indexing. It is invoked by submitting index data to an HTTP Servlet that runs on the host where the Windchill Index Search is located. The index loader invokes a Windchill Index Search API to initiate the indexing of metadata and content file information.

B

Windchill Considerations for Security Infrastructures

About Security Infrastructures

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 effect 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	

Client	Example	Communicate Protocols	Comments
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.

The host names for your Windchill system are set up during installation. For details, see the *Windchill Installation and Configuration Guide - Advanced*.

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.

The following sections provide some basic information about the use of RMI. For advanced RMI topics, see the *Windchill Advanced Deployment Guide*.

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.

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.

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.

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 Examples

The following examples show the syntax of an XSL-based policy file for both import and export. Comments explaining the use of policy files are embedded within the examples:

Import

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl=http://www.w3.org/1999/XSL/Transform"
version="2.0">
  <!--

    The following is a sample of a well-formed xsl. In the
    cases where there are no specific actions to be
    performed, the target xml (which is being transformed)
    should be copied as it is, which is achieved by the
    following:

    <xsl:template match="@* | node()" priority="-9">
      <xsl:copy><xsl:apply-templates select="@* | node()" />
    </xsl:copy>
    </xsl:template>

  -->
  <xsl:template match="@* | node()" priority="-9">
    <xsl:copy>
      <xsl:apply-templates select="@* | node()" />
    </xsl:copy>
  </xsl:template>
  <xsl:template match="WTPart">
    <actionInfo>
      <action:PickExistingObject</action>
    </actionInfo>
  </xsl:template>
  <xsl:template match="WTDocument">
    <actionInfo>
      <action>PickExistingObject</action>
    </actionInfo>
  </xsl:template>
  <xsl:template match="EPMDocument">
    <actionInfo>
      <action>PickExistingObject</action>
    </actionInfo>
  </xsl:template>
</xsl:stylesheet>
```

Export

```
<?xml version="1.0" ?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output method="xml" indent="no" encoding=UTF-8" />
  <!--
```


The syntax of Export 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>
</actionInfo>
-->
<!--
```

The following is a sample of a well-formed xsl. In the cases where there are no specific actions to be performed, the target xml (which is being transformed) should be copied as it is, which is the default action that would transpire as shown in the uncommented section below.

```
-->
<!--

<xsl:template match=@* | node()" priority="-9">
  <xsl:copy><xsl:apply-templates select="@* | node()"/>
</xsl:copy>
</xsl:template>

<xsl:template match='WTPart'>
  <actionInfo>
    <action>Checkout</action>
  </actionInfo>
</xsl:template>

<xsl:template match='WTDocument'>
  <actionInfo>
    <xsl:choose>
      <xsl:when test="number('TESTDOC-1')">
        <action>Checkout</action>
      </xsl:when>
      <xsl:otherwise>
        <action>Lock</action>
      </xsl:otherwise>
    </xsl:choose>
  </actionInfo>
</xsl:template>

-->
<!-- Copy everything as is by default -->
<xsl:template match=@* | node()" priority="-9">
  <xsl:copy>
    <xsl:apply-templates select=@* | node()" />
  </xsl:copy>
</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 the following:

```
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.

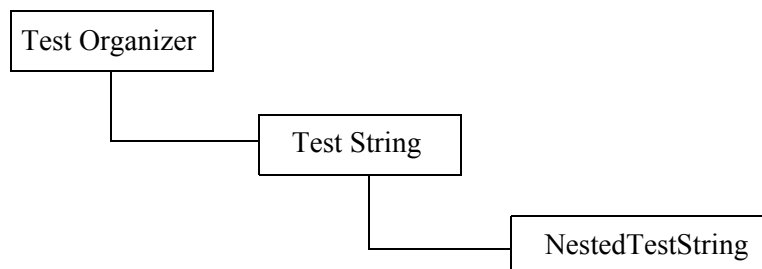
The creation of hierarchical IBA definitions without having the property set true is likely to have occurred in releases prior to Release 7.0 because no recommendation to set the property true existed for those releases. Nested Attribute Organizers were allowed in Release 7.0 as they were in prior releases, 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 for any XML files.

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>
```

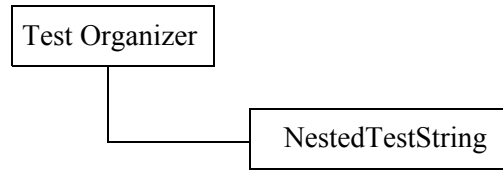
For all XML files to be imported, the preceding mapping rule will change the values for tag `<path>` to **TestOrganizer**, if the original value for this tag is **TestOrganizer/TestString**. Changing the values for other XML files other than `ibaDefinitions.xml` may not be the result that is expected. 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>
  <properties>
    #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 <i>Windchill Administrator's Guide</i> 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 <i>Windchill Administrator's Guide</i> for further information.
Location (Cabinet) does not exist	Create missing Cabinet if override conflicts = true, user notification otherwise	Cabinet <type> cannot be found. See the <i>Windchill Administrator's Guide</i> for further information. Note: In Windchill 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 Windchill PDMLink, the conflict should be ignored and the cabinet replaced by Default.
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 <i>Windchill Business Administrator's Guide</i> 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 <i>Windchill System Administrator's Guide</i> 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 <i>Windchill System Administrator's Guide</i> 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 which 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, `standardX10.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>standardX10.dtd</newDtd>
    </CHANGE_TAG>

</mappingRules>

</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>standardX10.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 <i>Windchill Business Administrator's Guide</i> 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 <i>Windchill System Administrator's Guide</i> for further information.
IBA Definition or Attribute Organizer does not exist	User notification	Definition of Instance Based Attribute <type> cannot be found. See the <i>Windchill System Administrator's Guide</i> for further information.
Quantity of Measure does not exist	User notification	Quantity of measure <type> does not exist.

Potential conflict	Behavior	Resolution or Message
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
About WebHelp.....	D-2
Customizing Topic Content.....	D-3
Customizing Navigation Pane Content	D-6
Customizing Topic Appearance	D-11

About WebHelp

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 ¹ .	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_whsl {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 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/whfwddata##.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.)

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.

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 tutorial link. 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. For more information, see [Windchill Maintenance 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 tutorials link. Do this 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](#).

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

Locale	Extension Value
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 pjIndex_<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

Overview Product Introduction Layout and Navigation Creating a Project	Creating a Company Part Working with Part Basics Doing Advanced Part Activities
Collaborating Discussing Projects Discussing Documents Subscribing Routing Documents Approval Routing Notify Routing	Managing CAD Models Pro/ENGINEER Wildfire Other CAD Tools Creating Parts Managing a BOM
	Microsoft Project Editing 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, pjllIndex_<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.

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*.
- 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 [Adding a New Tutorial to the Tutorial Menu Presented in a Windchill Solution](#).

Troubleshooting the Publishing Agent

Note: The Publishing Agent is traditionally known as the CAD Agent. However, configuring the CAD Agent is also pertinent to publishing non-CAD data, such as Dynamic Documents and WTDocuments. For the sake of simplicity, the remainder of this appendix and the user interface will use the term “CAD Agent”.

This chapter provides information you can use to analyze and resolve issues that may arise with the Visualization Services CAD Agent.

Common CAD Agent Problems

Several types of CAD Agent problems can occur, including:

- Connectivity
- Starting the worker: local worker, remote Windows worker, and remote UNIX worker.

To resolve problems involving the Distributed CAD Agent, all of the techniques for troubleshooting the CAD Agent still apply. Additional troubleshooting techniques are provided to address problems you may encounter when configuring a Distributed CAD Agent.

On Windows systems, be sure to read the information that is logged in the window running the Worker Daemon executable. Errors generated when starting the worker are logged in this window. If you are running the Worker Daemon as a service, then refer to the workerdaemon.log file for errors.

Troubleshooting the CAD Agent

The configuration of the workers to the CAD Agent can prove difficult. It does require that files are configured correctly, and when workers are on a remote system, that the network connectivity between machines is correct.

The CAD Agent will usually run as a service, defined in the wt.properties file as:

```
wt.services.service.nn=com.ptc.wvs.server.cadagent.  
CadAgentService/com.ptc.wvs.server.cadagent.  
StandardCadAgentService
```

It can run in its own Java Virtual Machine, as described in the Local Windows section.

The CAD Agent reads configuration settings from a file. The name of the file is defined in wvs.properties by the entry:

```
cadagent.inifile=$(wt.home)\\codebase\\agent.ini
```

Settings in this file should only be altered using the CAD Agent Configuration Wizard that, as an Administrator, can be accessed from the CAD Agent Monitor by selecting the Configure button.

The CAD Agent listens on a port for requests; the port number is defined in the agent.ini file. As shipped, the port is set to 5600. Should there be some other process on the system using this port, the CAD Agent will fail to initialize. In this case, manually editing the port setting in the [agent] section of the agent.ini file is the only way to change this port value.

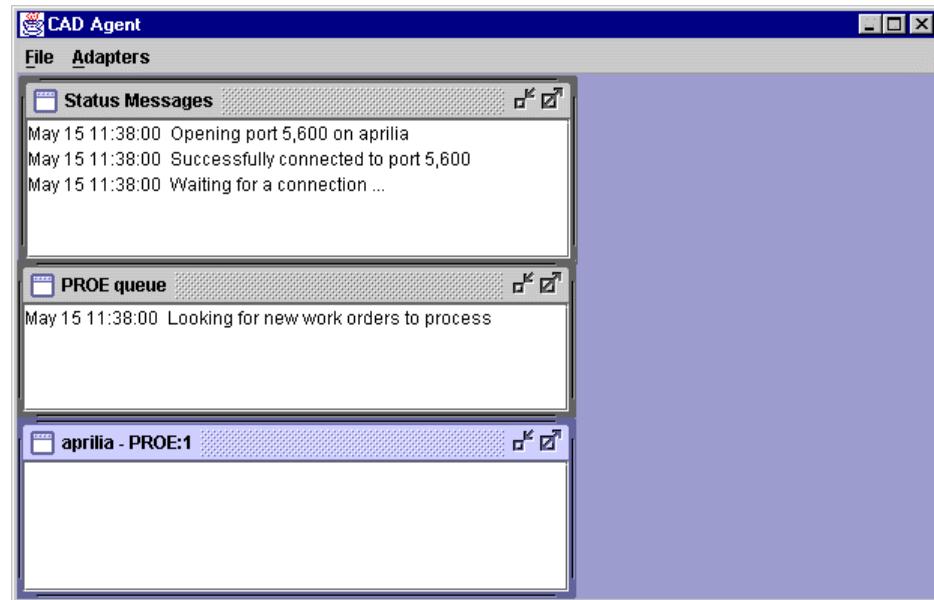
Running the CAD Agent in debug mode

If there are problems in configuring the CAD Agent and getting a worker to connect, it is best to start the CAD Agent in debug mode.

1. Stop the Method Server if it is already running
2. Start the CAD Agent.
3. From the Windchill shell, execute the command:

```
java com.ptc.wvs.server.cadagent.CadAgent -d
```

4. The following window opens:

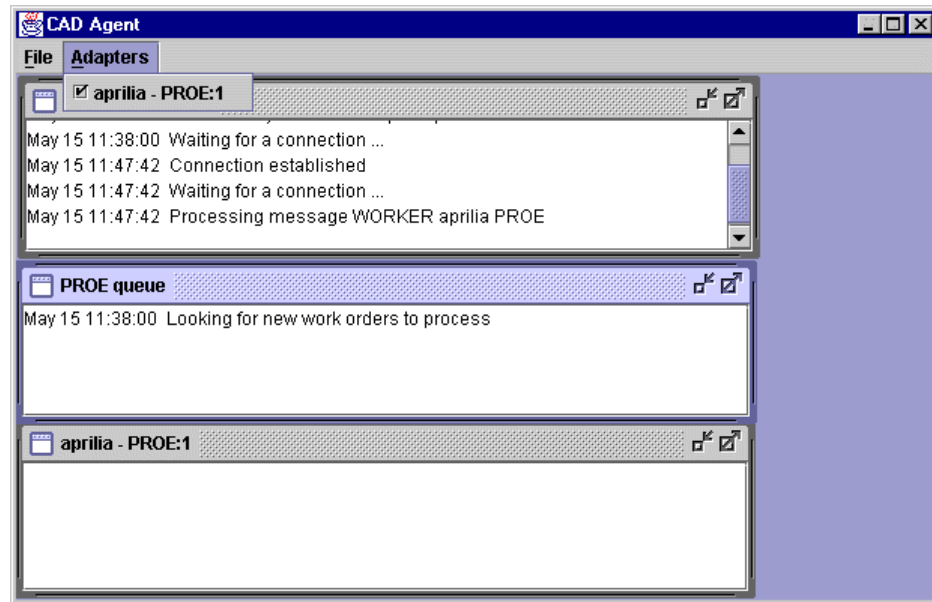


The number of sub panes displayed will depend on the numbers and types of configured workers. The **Status Message** pane shows the requests being processed by the CAD Agent on the listening port (5600). For each type of worker (PROE, CADDs, CATIA UG, and so on) there will be a pane showing the state of the Queue. Then each worker will have its own pane that will log its transactions. In this case, there is a single PROE worker configured, which is in fact local to the server on which the CAD Agent is executing.

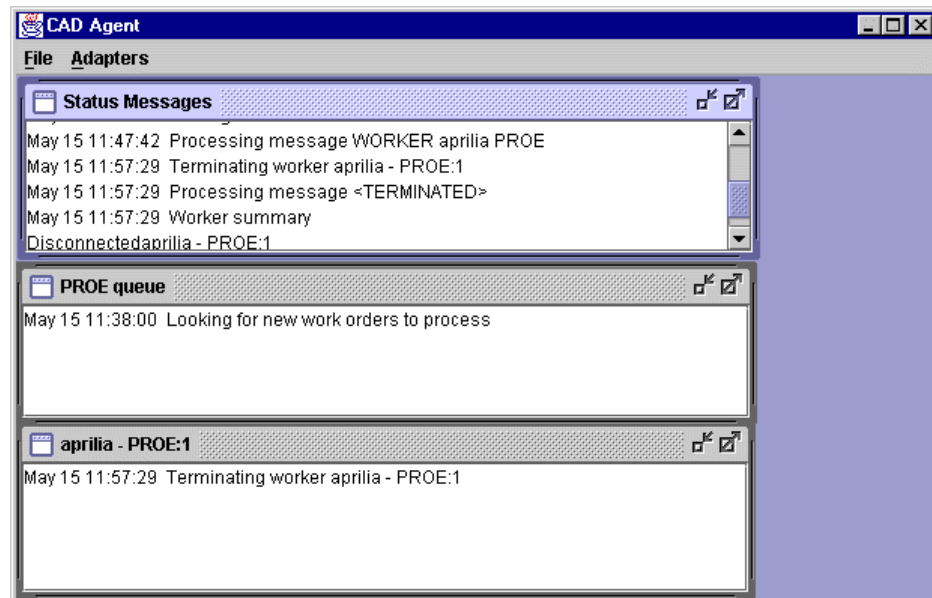
Manually Starting Workers

Next, start the worker by going to the system on which it is configured, and executing the BAT file or shell script that has been configured as the execute command to start the worker.

If this successfully connects, messages will be displayed in the **Status Messages** pane, and if under the Adapters drop down list, the entry for the worker will be checked.



To stop the worker, select the checked adapter entry, and this will send a message to stop the running worker.



If the worker fails to connect to the CAD Agent, then no activity will be observed in the CAD Agent windows. The worker may stay running, or exit after a period of time. This problem is most likely due to incorrect setting in the worker configuration file.

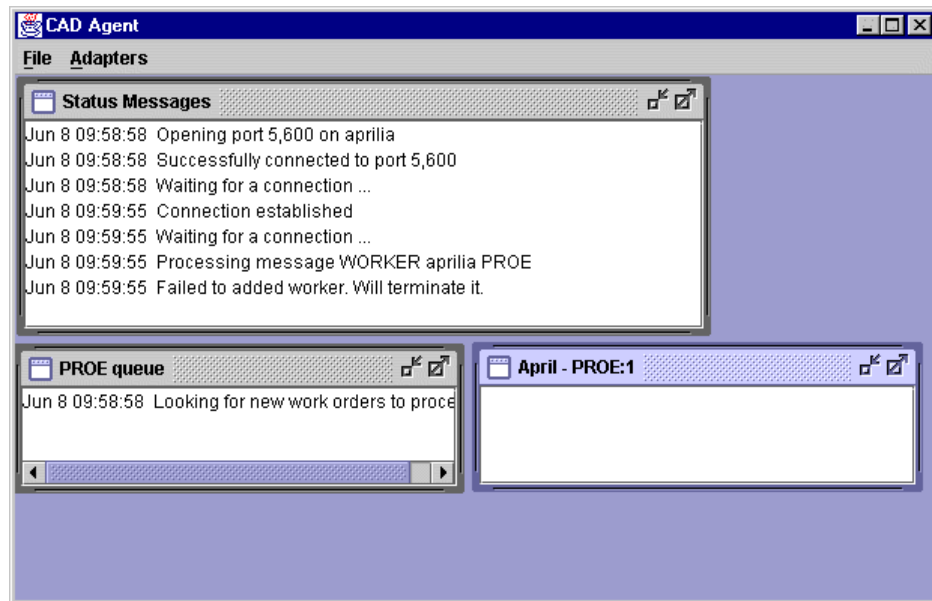
For example, for PRO/ENGINEER, ensure that the proeworker script correctly specifies the host (where the CAD Agent is running) and the port as defined in the agent.ini file (default 5600). The proeworker script is located in the directory that the worker is started from. The host and the port are defined in the script as arguments to the workermonitor command: “-CS<host> <port>”.

Also, if the worker is running on a different host to the CAD Agent, ensure that the worker host can communicate with the CAD Agent host, as the name specified in the worker proeworker.bat file.

In the case of remote UNIX workers, it is important that the network addressing is correct. It should be possible to use ping, ftp and telnet (no telnet required to remote Windows) to the remote worker machine specifying the actual hostname, and ping back from the remote worker to the CAD Agent server by hostname.

Manually starting the worker may cause a message similar to the following to be displayed in the **Status Messages** pane:

```
Processing message WORKER <hostname> <CADTYPE>  
  
Failed to add worker. Will terminate it.
```



This is due to a request being made from a worker that is not recognized as being configured in the agent.ini file. The hostname that is specified in the message must be the hostname that is specified in the agent.ini file. In this case, it is

looking for a hostname of aprilia, where as the host of April is set in the agent.ini file. In this case, the CAD Agent will send a message back to the worker to stop it.

Generally you will find when using DNS that the hostname sent from the worker will be the fully qualified name, for example Aprilia.ptc.com, and the fully qualified name should also be used in the agent.ini file. With Windchill, the CAD Agent will attempt to locate the entry in the agent.ini file both with and without the domain name. Windchill requires an exact match.

Starting Workers from the CAD Agent

When the worker has successfully been able to connect to the CAD Agent using a manual start of the worker. Attempt to start it by selecting the entry in the Adapter drop down list. When the entry is selected, the menu will stay selected, until the startup timeout is exceeded, or a connection from the worker is achieved.

If this fails, check the command specified to execute it is in fact correct.

Local Windows

In the case of a local worker on Windows, it is worth trying to run the command using cmd.exe. For example a PROE worker can be started as:

```
cmd.exe /C start "PROEWORKER" /MIN c:\ptc\ productview_adapters\  
proe_setup\proeworker.bat
```

Remote Windows

Workers that are running on a remote Windows client must be started using the Worker Daemon.

Remote UNIX

For remote UNIX workers, specifying nohup and putting the task in background is required. Also, Telnet is used to connect to the remote worker. To test, use Telnet from the CAD Agent host, to the worker host. Specify the hostname, username and password defined (in the agent.ini file) during the configuration of the worker. Assuming a successful connection, it is important that the system prompt (for that user) should not change. Manually executing the specified worker command from this environment should create a connection to the CAD Agent. If not, then there is probably a difference in the environment used by Telnet and that the default user login. Adjust the environment to ensure that the command causes a connection via telnet, look for things like DISPLAY, path, and shell type. The CAD Agent should then be able to start the worker.

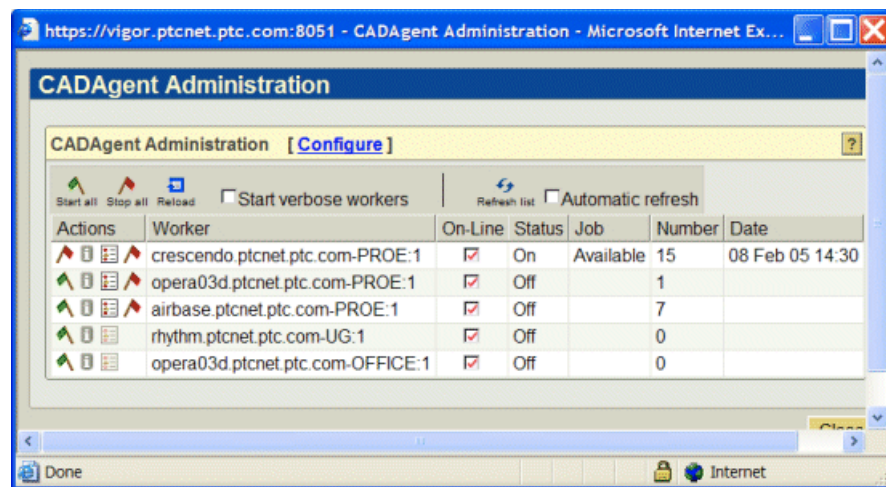
Using the CADAgent Monitor

The CADAgent Monitor provides various troubleshooting options. From this window, you can start and stop workers, test whether a worker is set up properly, view worker log files, and start verbose workers.

1. Open Windchill in a Web browser and log in as a Site Administrator.
2. In the Windchill application, do the following to navigate to the CAD Agent Administrator:
Click on the Site (tab) > Utilities (tab) > CAD Agent Administrator (link located under the System Administration heading).

Click on the Site (tab) > Utilities (tab) > CAD Agent Administrator (link located under the System Administration heading).


3. The CADAgent Administration window opens.



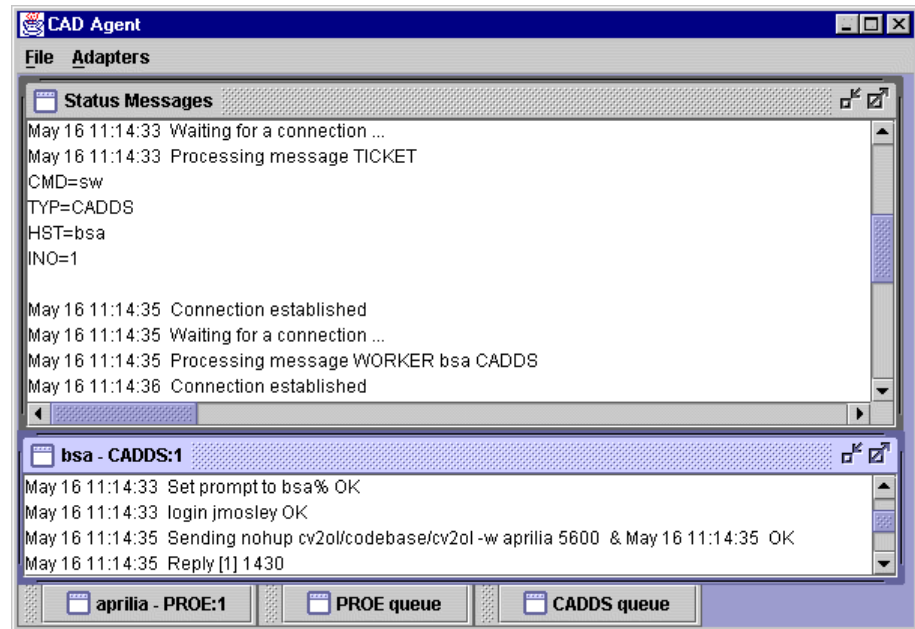
The next sections explain the features available in the CAD Agent Monitor.


Starting and Stopping workers Using the Monitor

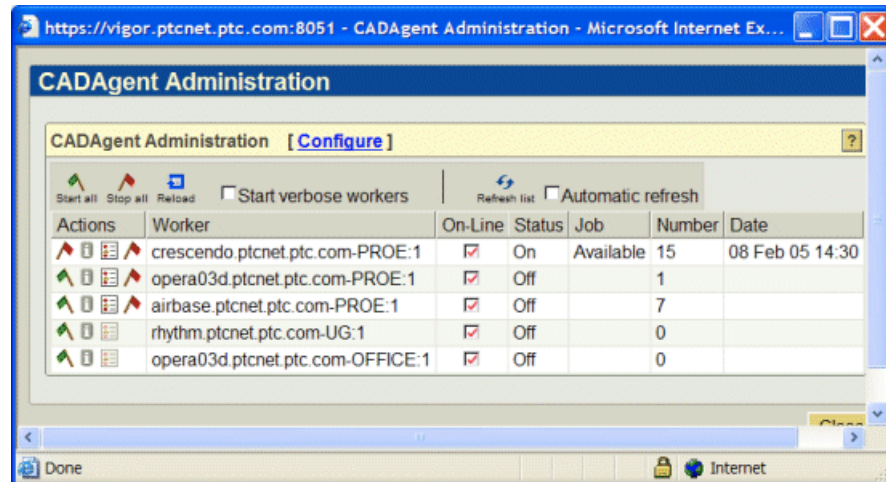
With the CAD Agent still running in debug mode, but with the workers disconnected, you can view the status of the workers in the CAD Agent Monitor.

1. To start a worker, select the green flag icon  in the **Actions** column of the CAD Agent Monitor. This sends a message to the CAD Agent to start the worker. The **Status** column in the **CAD Agent Monitor** shows that the worker is **On**, and the icon in the **Action** column changes to a red flag icon.

In the **CAD Agent Status Message** pane, messages indicate that the Agent has received the request, and has attempted to start the worker. A flag should appear against the Adapter menu entry for the worker.



- To stop the worker, select the red flag icon .



If the CAD Agent shows that the worker has successfully connected, but the status in the CAD Agent Monitor still says **Off**, select the **Automatic Refresh** option in the CAD Agent Monitor. If the worker then shows that it is **On**, this indicates that the start time specified for the worker is set too low. Select the **Configure** link at


the top of the CAD Agent Monitor window to invoke the CAD Agent Configuration wizard, where you can change this setting.

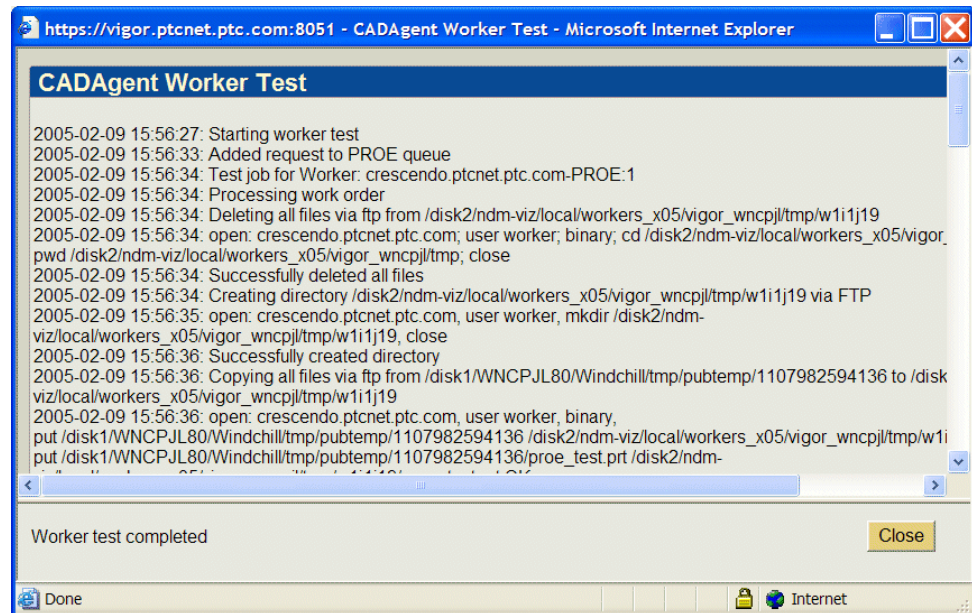
It is important that the worker starts within the specified timeout period. When the system is fully running, the worker should start automatically. If the timeout is set too low, the CAD Agent will make up to 3 attempts to start the worker, and if it does not achieve a connection, the worker will be marked as unable to start.

Testing a Worker

You can also test workers using the CAD Agent Monitor. This process ensures that the basic CAD Agent worker setup was successful. It determines whether the worker is responding, and runs a test conversion of data that is not stored in Windchill to ensure that it is being properly converted.

Note: Before you can test a worker, you must start the worker, and it must be offline (uncheck the **On-Line** option).


To test a publishing job, click the **Test Worker** icon  in the **Actions** column of the CAD Agent Monitor. This displays the CAD Agent Worker Test Window, shown below



The **CAD Agent Worker Test** window displays messages logged by the CAD Agent during the publication test process, including all files created by the worker, file size, and any failure or errors that occur. This information can be used for debugging purposes

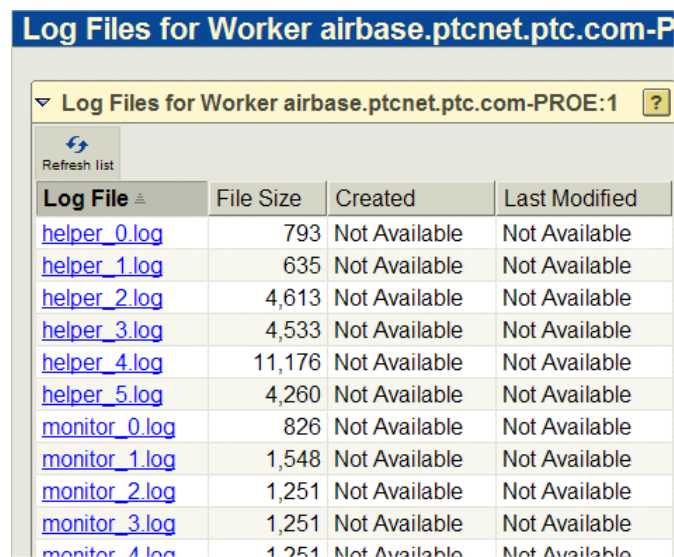
Viewing Log Files

In the CAD Agent Monitor, you can access a list of log files and view any log file in detail. Follow these steps to view log files:

1. Click the **View Worker Log Files** icon  in the **Actions** column. This displays a list of available log files on the worker machine, as shown below. You can view monitor, helper, and worker log files. The file names help identify whether the log file pertains to a monitor, helper, or worker. A sample list of log files is shown below.

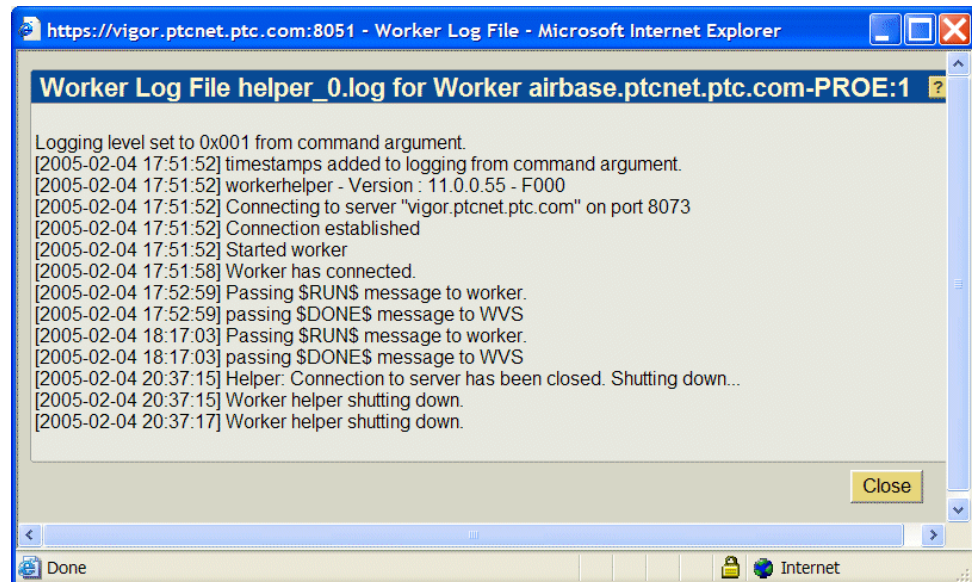
Note: If you are using a worker without a worker monitor, the ability to remotely view log files will not be available.

Note: Each publish job in the Publish Monitor displays the name of the worker log file that is being used.



Log File	File Size	Created	Last Modified
helper_0.log	793	Not Available	Not Available
helper_1.log	635	Not Available	Not Available
helper_2.log	4,613	Not Available	Not Available
helper_3.log	4,533	Not Available	Not Available
helper_4.log	11,176	Not Available	Not Available
helper_5.log	4,260	Not Available	Not Available
monitor_0.log	826	Not Available	Not Available
monitor_1.log	1,548	Not Available	Not Available
monitor_2.log	1,251	Not Available	Not Available
monitor_3.log	1,251	Not Available	Not Available
monitor_4.log	1,251	Not Available	Not Available

2. To view a specific log file, click the filename link in the **Log File** column. A sample worker log file is shown below.



Overview of Worker, Helper, and Monitor

Monitor log files

- The Monitor controls communication between the CAD Agent and the worker machine. It allows the server to communicate with the worker machine, regardless of whether or not the worker is actually running. This allows the server to view log files from each of the applications (monitor, helper, and worker) even when the worker fails to start. The Monitor also receives the start command from the server when the worker needs to start.
- If multiple instances of a worker are required, there will be only one “workermonitor” process, which starts a new instance of the worker for each instance required.
- The monitor starts, then WVS requests that the monitor start the worker. The monitor then starts the helper, which in turn starts the worker.

Helper log files

- The Helper acts as a command center for the worker, supervising the worker.
- The helper log file provides information such as whether the worker is up or down, how the publish job is proceeding, and overall worker behavior.
- The workerhelper monitors communication between the server and the worker and can implement the timeouts locally. If a problem occurs or the worker needs to be shut down, the Helper ensures that the worker is shut down completely and cleans up any processes or files that remain.

Note: The “workerhelper” cannot guarantee a 100% recovery from errors on the worker. In some cases a re-boot may be required, or processes must be ended manually.

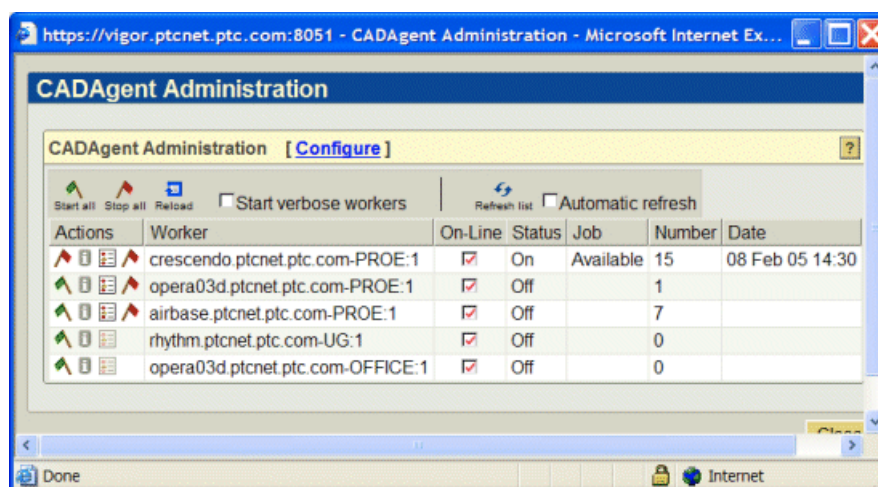
Worker log file

The Worker log file displays the publish conversion process. If a publish job fails, this file contains the debugging information for the conversion. You can set the debug option when publishing to specify what information recorded in this log file, using the debug options text file in the setup directory specified during configuration.


Starting Verbose Workers

Using the option to **Start verbose workers**, you can dynamically change the level of debugging verbosity to restart the worker. This option applies to workers manually started from the user interface. Selecting this option instructs the worker to use the command line options specified in the debug_options.txt file, located in the worker setup directory specified during the ProductView Adapter configuration.

1. To troubleshoot workers, select the **Start verbose workers** option in the **CADAgent Administration** window.
2. Check the green flag icon in the **Actions** column to start the worker.

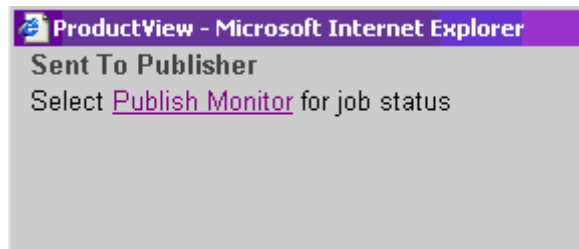


Manually Publishing a CAD Document

With the worker connected to the CAD Agent, the next step is to publish data. Locate a CAD Document stored in Windchill. From the CAD Document's information page, select the Visualization icon .



After the Visualization icon is clicked, the **Sent to Publisher** message is displayed.



From this window, you can access the Publish Monitor, which is described next.

Using the Publish Monitor

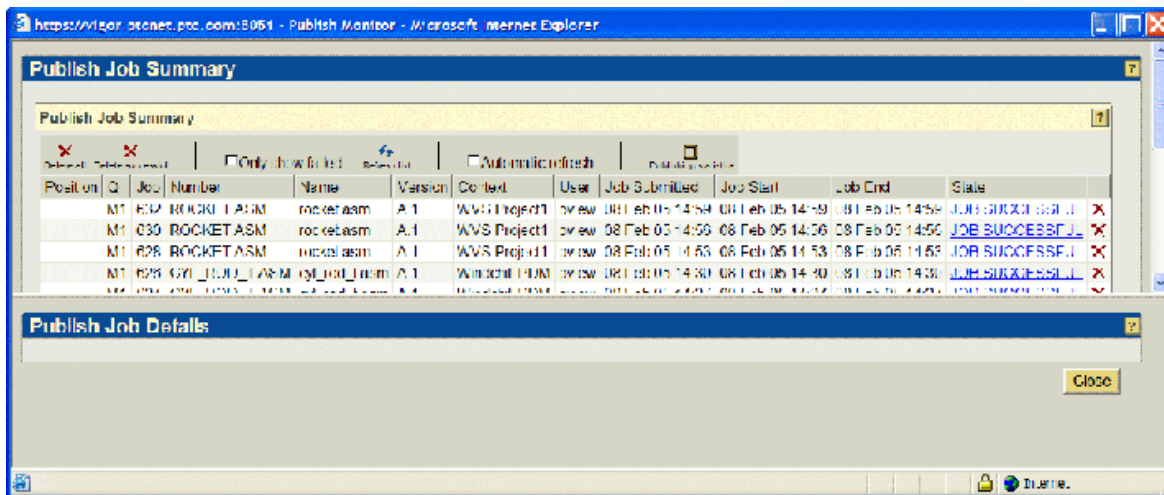
The Publish Monitor provides a set of tools to help assess the ongoing health of the publishing sub-system. This information includes:

- Number of jobs for each worker which have been processed
- Number of failed jobs
- Time and date for each job
- Priority of queue


- Submission, start, and end times
- Event that initiated the publishing job

To display the Publish Monitor, select the **Publish Monitor** link from the Utilities (tab) page. There are multiple Utilities links, and the behavior of the Publish Monitor is based on the context that it was launched from. For example, launching the Publish Monitor from the Home/Utilities page will show only jobs that you have submitted. Launching the Publish Monitor from the Utilities page of a product/project/library will show jobs that you have access to in that context. Similarly, this concept applies to the organization and site utilities context in the Visualization portal window or from the Windchill Visualization home page.

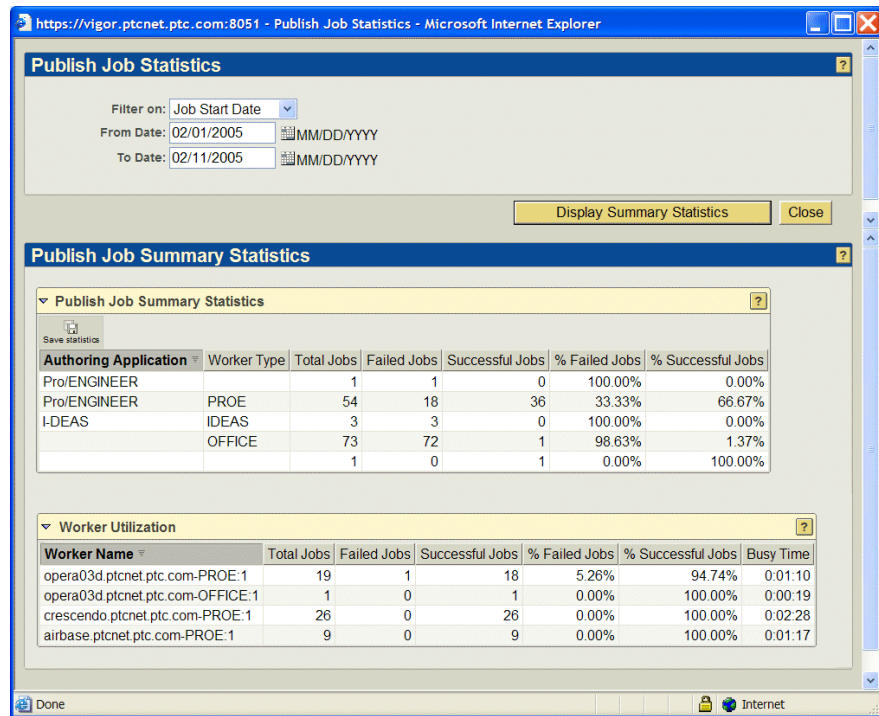
The Publish Monitor contains two windows: Publish Job Summary and Publish Job Details.



- To display the detail for a specific job, select the link in the **Status** column.

- To display publishing statistics, select the  icon in the **Publish Job Summary** area of the Publish Monitor.

The Publish Job Statistics window, shown next, displays information including the CAD authoring tool, worker name, number of jobs, both failed and successful, and the overall percentage of failed and successful jobs for each entry. This information helps you diagnose the overall system health.



- Below the **Publish Job Statistics** area, there is a **Worker Utilization** summary table, which displays the same publishing statistics, organized by worker, and tells you the busy time for each worker as well.
- You can click the **Save Statistics** button to save the publishing job statistics to a CSV (comma-separated value) file, which can be read into applications such as Microsoft Excel. This information includes more detail than the summary window.

Troubleshooting the Publish Monitor

If the **Publish Monitor** window is empty (for example, there is no Job Summary table heading), the WVS jsp pages might not be correctly authenticated.

If no entries appear in the Status panel, or they appear and then disappear once completed, ensure that the following settings are in wt.properties:

- For each PublisherQueue, keep the entries so that the log may be seen in the publisher.
`wt.queue.removeCompleted.PublisherQueue1=false`
- Additional queues can be added in the order that the solution scales. For each additional queue, the appropriate
`wt.queue.removeCompleted.PublisherQueue'n'=false` entry must be added to the wt.properties file.

For example:

`wt.queue.removeCompleted.PublisherQueue2=false`

Note: For more information on the publishing queues, see [Out-of-the-box Background Queues](#).

Note: The queue content is displayed to all users, but only the detail of a user's job is available.

When the job completes, the details indicate success or failure. The messages indicate any error that caused the failure.

Index

A

- Ad hoc replication, 4-26
- Administrative tasks, 1-68
- Architecture
 - three-tier, A-3
- Authentication
 - process, 1-62
 - text changes, 1-61
 - troubleshooting, 1-63
 - user, A-5, B-3

B

- Background queues. See Queues
- Backup
 - process, 1-55
 - replica sites, 4-33
- Best practices
 - administrative tasks, 1-68
 - installation, 1-76
 - maintenance, 1-76
- Bootstrap client, 2-1–2-3
 - configuration file, 2-3
 - JAR file cache location, 2-3
- Bootstrap loader, 2-2
- Bootstrap package
 - properties, 2-4–2-7
 - versioning, 2-7
- Bulk Indexing Queue, 8-5

C

- Caches
 - central cache vault, 3-5
 - local, 4-16
 - mirroring, 4-18
 - moving data from, to master site, 4-19
 - vaults for, 4-18
- CAD agent
 - Overview, 9-1
 - troubleshooting, F-1
- CAD license, 9-8

- CAD parts
 - indexing, 1-67
- Central Cache Vault, 3-5
- Changing file location in external vaults, 3-28
- Client Types
 - jconsole, 1-8
 - JManage, 1-8
- Codebase
 - administering, 2-8
 - server property, B-4
- Collections
 - bulk loading, A-22
- Command line parameter
 - port number, 9-14
- Common File
 - system directory location, 9-10
 - UNIX, 9-14
- Configuration properties, 1-2, B-7
- Configuring, 1-60
 - background method queues, 8-10
 - bootstrap client, 2-3
 - content handling, 7-3
 - queues, 8-10
 - Windchill environment, 1-60
- Conflict messages
 - administrative objects, C-20
 - common administrative objects, C-29
 - common business objects, C-18
 - dependency, C-22
 - metadata, C-23
- Content cache, 4-19
- Content Cache Server, 4-15
- Content holders, 7-1
 - about, 7-2
 - data formats, 7-4
- Content objects, 7-1
- Context in import and export, 6-4
- Context mapping files, C-13
- Converting multiple vaults to single vault, 3-26
- Creating
 - external file vaults with FvLoader, 5-1, 5-2
 - mounts, 3-14
 - vault rules with FvLoader, 5-1, 5-2
 - vaulting rules, 3-19

Custom modeled attributes, reforming, C-26

D

database

- backing up, 1-55

Database components, A-18

Default value, setting, 1-42

Dependency conflicts, C-22

DirContexts MBean, 1-26

Domains

- listing, 5-7

Download files

- Pro/ENGINEER, 10-2

Downloading

- JAR files, 2-8

DTD file

- xconf, 1-5

Dumper MBean, 1-26

E

Editing property files, 1-2

E-mail notifications, 1-46

Emailer MBean, 1-26

Expired rules

- cleaning up, 4-24

Export, 6-1

- about, 6-2

- Access control rules, 6-42, 6-43

- EPMDocuments, 6-27

- EPMDocuments as Checked Out, 6-28

- graphical user interface, 6-28

- hierarchical IBA definitions, C-13

- mapping rules, C-4

- object from its properties page, 6-32

- supported objects, 6-8

Exporting an object from its properties page, 6-32

External file vaults

- configuring, 5-4

External Storage Administrator window, 3-12

External Storage Scheduling window, 3-23

F

File Server, 4-23

- deleting registered, 4-29

- registered, 4-29

- updating, 4-29

- user experience, 4-31

File Size Limit for Content Replication, 4-14

File Vault Administrator, 3-12–3-17

File vaults

- about, 3-2

- creating with FvLoader, 5-1, 5-2

- external, 3-1, 3-2

- external in replication, 4-2

- maintaining, 3-17

- properties, 3-7

- replica, 4-2

- rules, 3-19–??

- rules for, ??–3-19

- rules for creating, 3-20

- summary, 3-2

- See also Revaulting

Forcing Content to Vault, 3-25

FTP user name, 9-12

FTP, UNIX, 9-14

Full-scale replica site, 4-7

FvLoader, 5-1

- commands, 5-3

G

GarbageCollection MBean, 1-27

H

Hierarchical IBA definitions, C-13

Hosts

- choosing names, B-5

HTTP

- gateway, A-6

- server, A-5

I

IeCalls MBean, 1-27

IeContexts MBean, 1-27

Import, 6-1

- About, 6-2

- Access control rules, 6-42, 6-43

- Conflict messages, C-17

- EPMDocuments, 6-27

- EPMDocuments as Checked Out, 6-28

- graphical user interface, 6-34

- hierarchical IBA definitions, C-13

- mapping rules, C-4

- overridable conflicts, C-23

- properties, 6-41

- supported objects, 6-8

Index Search

- backup, 1-56

- indexing CAD parts, 1-67

Indexing

- file content, A-20
- overview, A-20
- policies, A-21
- publishing, A-21
- queue, 8-5

InfoEngine MBean, 1-27

Instance Based Attribute definitions, C-13

Internationalization, A-2

J

JAR

- import, 6-30
- top level, 6-30

JAR caching, 2-3

JAR file

- export and import, 6-2

JAR File Cache, 2-3

JAR files, 1-5, 2-2

- downloading, 2-8
- importing and exporting, 6-1
- security, 2-9

Java applets, A-4

Java database connectivity (JDBC), A-2

Java Management Extensions. See JMX

Java Mapping, C-12

Java Plug-in Cache, 2-3

Java property files. See Property files

Java system property settings

- bootstrap client, 2-6

JMX

- clients, 1-4, 1-8
- introduction, 1-7
- MBeans, 1-2, 1-7, 1-26

L

lightweight replica site, 4-7

Loader MBean, 1-27

Local Upload, 4-15

Log events

- storing, 1-59

Log files, 1-57

- e-mail, 1-24
- queues, 8-13
- usage, A-18

Logging

- file settings, 1-57
- managing, 1-56
- MBean, 1-27

M

Mapping rules, C-1

Context mapping files, C-13

Java mapping, C-12

METHOD Element, C-12

overview, C-4

special rules, C-4

XSL transformation, C-12

Master sites, 4-7

local cache, 4-19

Maximum File Size for Content Replication, 4-14

MBeans

accessing form JConsole, 1-19

DirContexts, 1-26

Dumper, 1-26

e-mail logs, 1-24

Emler, 1-26

GarbageCollection, 1-26

IeCalls, 1-27

IeContexts, 1-27

InfoEngine, 1-27

JMX client, 1-8

Loader, 1-27

log files, 1-23

Logging, 1-27

Memory, 1-28

MethodContexts, 1-28

MethodServer, 1-28

MethodServers, 1-28

NotificationHandlers, 1-29

ProcessCpuTime, 1-30

Properties, 1-30

PropertiesManager, 1-30

property values, 1-20

Queues, 1-30

Reporting System, 1-30

RMICalls, 1-30

Server Manager (monitor), 1-30

ServletRequests, 1-31

ServletSessions, 1-31

start and stop Windchill, 1-22

Summaries, 1-31

Support, 1-33

UserSpecified (monitor), 1-33

using, 1-7

VaultSites, 1-33

viewing log files, 1-24

WebAppContexts, 1-33

WindchillVersion, 1-33

WrappedSockets, 1-34

wt.logs.dir, 1-28

- Meeting Center, 1-46–1-48
- Memory MBean, 1-28
- menuObjects.xml files, E-1
- Metadata conflicts, C-23
- METHOD Element, C-12
- Method servers
 - component, A-14
 - configuring single, 1-61
 - external vaults, 3-3
- MethodContexts MBean, 1-28
- MethodServer MBean, 1-29
- MethodServers MBean, 1-29
- Microsoft Office
 - desktop preferences, 1-52
- Migration
 - vaulting, 3-29
- Mounts
 - creating, 3-14
 - folders, 3-14
 - root folders, 3-14
 - updating, 3-14
- moveItemsBetweenReplicaVaults, 4-10
- Multi-valued property
 - adding, 1-43
 - removing, 1-43

N

- NotificationHandlers MBean, 1-29
- Null value, setting, 1-42

O

- Object Relational Database Management System (OR-DBMS), A-18
- Oracle database
 - replication, 4-2
- Overridable conflicts, C-23

P

- Parallel replication
 - about, 4-27
 - process, 4-28
 - property controls, 4-29
- Policies for Import and Export, C-1
- Predictive replication, 4-23
- PREDICTVE, 4-24
- Preferences
 - Desktop Integration, 1-52
- Preferred Content Cache Site
 - setting, 4-19

- ProcessCpuTime MBean, 1-30
- Properties MBean, 1-2, 1-30
- PropertiesManager MBean, 1-2, 1-30
- Property, 1-43
- Property files
 - editing, 1-2
 - listing values, 1-43
- Proximity, 4-26
- Proxy servers
 - Meeting Center, 1-48
- Publishing agent. See CAD agent
- Publishing Pro/ENGINEER Wildfire Assemblies, 10-8

Q

- Queue Manager, 8-1, 8-2
- Queues
 - background, 8-1
 - configuring, 8-10
 - DeliveryStatusOnStartup, 4-12
 - entry states, 8-2
 - log files, 8-13
 - maintaining, 8-14
 - maximum values, 4-13
 - MBean, 1-30
 - out-of-the-box, 8-3
 - PurgeScheduleHistoryQueue, 4-13
 - replication, 4-12

R

- Reforming custom modeled attributes, C-26
- Remote machine, 1-15
- Replica site vaulting, 5-5
- Replica sites, 4-15
- ReplicaLoader, 5-1
 - commands, 5-4
 - tasks, 5-2
- Replication, 4-1
 - compression, 4-14
 - creating rules, 4-24
 - importing certificates, 4-8
 - master sites, 4-7
 - performance, 4-15
 - queues, 4-12
 - security, 4-7
 - summary, 3-2
 - Windchill Visualization Service, 4-14
- Reporting System MBean, 1-31
- Reports, 1-53
- Revaulting

- History window, 3-24
 - process, 3-22
 - schedule items, 3-23
- Revisions, changing, 1-65
- RMI, B-6
- RMICalls MBean, 1-31
- Root folder, 3-5
 - updating mounts, 3-15
- Rules
 - configuring, 5-4
 - listing, 5-7
 - removing, 5-7
 - replication, 4-24
- Runtime environment, A-1
- Runtime services chapter, 1-1

S

- secret.text, 1-61
- secret.text2, 1-61
- Security
 - Infrastructures, B-1
 - JAR files, 2-9
 - shared directory login, 9-10
- Server codebase property, B-4
- Server hostname property, B-6
- ServerManager (Monitor) MBean, 1-31
- ServerManager MBean, 1-31
- ServletRequests MBean, 1-31
- ServletSessions MBean, 1-32
- Site Proximity, 4-26
- site.xconf, 1-2, 1-4, 1-34
- Special rules for mapping, C-4
- Stopping Windchill, 1-22
- Storage Administrator window, 3-12
- Summaries MBean, 1-33
- Support MBean, 1-33
- Supported objects for import export, 6-8
- System account
 - workerdaemon, 9-12
- System Administrator's Guide
 - audience, xxi
 - organization, xxii
 - overview, xxi
 - related documentation, xxi
- System configuration
 - Windchill, 1-60

T

- TAN example, 9-16
- Technical Application Note, 9-16

- Technical support, xxiv
- Temporary files, 1-60
- Threshold detection, 1-61
- Traffic Light icon, 9-18
- Tutorials, E-1

U

- Updating
 - mounts, 3-14
 - root folder mounts, 3-15
- Upgrade, replication, 4-31
- URL generation, B-4
- URL security, 4-8
- Usage report
 - creating, 1-53
 - running, 1-53
 - utility, 1-53
- User initiated replication, 4-26
- Users
 - authenticating, 1-62–??, A-5
- UserSpecified (Monitors) MBean, 1-33

V

- Validating XCONF files, 1-44
- Vault Configuration, 3-4
- Vault Configuration window, 3-12, 3-13
- vault diagnostics, 3-29
- Vaulting Rule window, 3-20
- Vaulting rules. See File vaults.
- Vaults. See File vaults
- VaultSites MBean, 1-33
- Viewing
 - revaulting results, 3-24
 - sites in tree view, 3-4
- Visualization server, 9-1

W

- Web browsers, A-3
- WebAppContexts MBean, 1-33
- WebEx, 1-45, 1-48, 1-49
- Windchill adapter, 1-61
- Windchill clients
 - components, A-3
- Windchill environment
 - configuring, 1-60
 - starting and stopping, 1-22

- Windchill Index Search, 8-5
 - backup, 1-56
 - indexing CAD parts, 1-67
 - indexing components, A-20
- Windchill PDMLink
 - supported objects for import/export, 6-13
- Windchill ProjectLink
 - supported objects for import/export, 6-9
- Windchill servers
 - components, 1-60, A-5
- Windchill Service Pack, 1-73
- windchill shell, 1-34
- Windchill Visualization Service
 - replication, 4-14
- WindchillVersion MBean, 1-33
- WorkerDaemon.exe, 9-9
- workerwaemon.log, 9-13
- WrappedSockets MBean, 1-34
- wt.fv.replicationFileSizeThreshold, 4-14
- wt.logs.dir MBean, 1-28
- wt.properties file
 - content replication, 4-8
 - file vaulting, 3-7
 - import, 6-41
 - log files, 1-57
 - primary, 1-2
 - queues, 8-10

- wt.queue.queueGroup, 8-11
- wt.vc.ChangeFirstRevisionLabelUtility, 1-65
- wt.vc.views.ReassignSingleViewCleanser, 1-66
- WVS viewables, 4-14

X

- XCONF files, 1-2, 1-3
 - overview, 1-3
 - validating, 1-44
- xconf.dtd, 1-4
- xconfmanager, 1-34, 1-47
 - command syntax, 1-35
 - help, 1-40
 - introduction, 1-2
 - JMX, 1-2
 - syntax, 1-35
 - using, 1-34
 - utility, 1-3
 - validating XCONF files, 1-44
- XSL transformation, C-12