



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

**Windchill 9.0**

**Pro/INTRALINK® 9.0™**

**Arbortext® Content Manager™**

**Windchill® PDMLink™**

**Windchill® ProjectLink™**

**September 2007**

**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

<b>Change Record .....</b>	<b>xix</b>
<b>About This Guide.....</b>	<b>xlili</b>
Overview .....	xliv
Chapter Contents .....	xliv
Related Documentation .....	xlvi
Technical Support.....	xlvi
Documentation for PTC Products.....	xlvi
Comments .....	xlvi
<b>Getting Started.....</b>	<b>1-1</b>
Introduction .....	1-2
Regarding Arbortext Content Manager .....	1-2
Regarding Pro/INTRALINK 9.0 .....	1-3
Logging On as the Administrator .....	1-3
Windchill PDMLink Home Page .....	1-4
Windchill ProjectLink Home Page.....	1-6
Arbortext Content Manager Home Page .....	1-7
Pro/INTRALINK 9.0 Home Page .....	1-9
Working With the Initial Organization Context .....	1-10
Establishing Administrators .....	1-12
Organization Administrators .....	1-13
Windchill PDMLink Administrators .....	1-13
Windchill ProjectLink Administrators.....	1-14
Creating Users to Select as Administrators .....	1-15
Establishing End Users .....	1-15
Using an Enterprise Directory Service .....	1-16
Using the Principal Administrator Utility.....	1-16
The Next Steps.....	1-17
<b>Administration Overview .....</b>	<b>2-1</b>
Your Installed Windchill Runtime Architecture.....	2-2
Managing Your Windchill System.....	2-3
Your Installed Windchill Environment .....	2-3

Managing User Access to Data .....	2-5
Product and Library Hierarchy.....	2-7
Program and Project Hierarchy .....	2-7
Hierarchy for Integral Windchill Solutions.....	2-8
Managing Access to Data through Access Control Rules.....	2-8
Setting Up User Access to Data.....	2-14
Managing Access to Data through Team Memberships .....	2-15
Managing Users .....	2-16
Managing Data.....	2-18
Data Types .....	2-20
Soft Types .....	2-20
Visualization Data.....	2-20
CAD Data .....	2-21
Dynamic Document Data .....	2-21
Document Data .....	2-22
Part Data .....	2-22
Auditing .....	2-22
Managing Windchill Processes .....	2-23
Managing User Collaboration .....	2-23
Additional Administrative Groups.....	2-24
Post-Installation Activities .....	2-25
About the windchill Command .....	2-25
About the windchill shell.....	2-27
About the xconfmanager Utility .....	2-28
Formatting Property Value Guidelines .....	2-30
<b>Contexts .....</b>	<b>3-1</b>
Distributed and Hierarchical Administration .....	3-2
Context Administrative Items .....	3-4
Context Configuration.....	3-4
Context Structure .....	3-6
Context Participation .....	3-7
Context Policies.....	3-9
Context Data Types and Attributes .....	3-13
Templates.....	3-15
Object Initialization Rules .....	3-21
Context Preferences.....	3-23
Creating the Contexts from which Users Work .....	3-23
Using Out-of-the-box Context Templates.....	3-24

Administering Domains and Policies .....	3-25
Context and Domain Hierarchy Overview .....	3-26
Domains in the Site Context .....	3-29
Creating Domains .....	3-30
Defining Domain-based Policies .....	3-31
Using the Policy Administrator .....	3-31
Specifying Policy Rules in a Context Template .....	3-33
Assigning Domains to Folders in Solutions with Products and Libraries .....	3-34
Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects ..	3-35
Out-of-the-box Numbering Schemes .....	3-35
Changing Numbering Schemes .....	3-36
Administering the Versioning of Parts, Documents, and CAD Documents .....	3-36
Understanding the Use of Versioning Schemes .....	3-36
Initial Versioning Rules .....	3-37
Changing Versioning Schemes .....	3-37
Administering Preferences .....	3-38
Preference Hierarchy .....	3-38
Preference Manager .....	3-39
Administering Saved Searches .....	3-40
<b>Site .....</b>	<b>4-1</b>
Overview .....	4-2
Typical Duties of Site Administrators .....	4-2
Creating and Managing Organizations .....	4-3
Adding and Editing Members .....	4-4
Creating and Managing Site Folders and Documents .....	4-4
Changing Default Configuration Options .....	4-4
Managing Site-level Types and Type-specific Attributes .....	4-5
Configuring Numbering and Versioning Schemes and Units of Measure .....	4-6
Managing Site-level Templates .....	4-6
Managing Site-level Object Initialization Rules .....	4-6
Managing Workflow Security .....	4-7
Auditing System Information .....	4-7
Creating and Managing Access Control Policies .....	4-7
Creating and Managing Profiles .....	4-8
Configuring External Vaults or Replication Sites to Optimize Performance .....	4-8
Configuring and Managing CAD Publishing Utilities .....	4-9
Creating, Updating, and Managing Reports .....	4-9
Importing and Exporting Information Among Systems .....	4-9
Managing Calendar Settings .....	4-9

Monitoring Enterprise Systems Transactions Log.....	4-10
Purge, Archive, and Restore Jobs.....	4-10
Managing Searches .....	4-10
Viewing and Managing Access Control Rules for Objects .....	4-11
Managing Overall System Configuration.....	4-11
Business Object Uniqueness Considerations .....	4-11
Out-of-the-Box Site Configuration.....	4-15
About the Site Tab .....	4-15
Best Practices .....	4-16
For Windchill Solutions with Products and Libraries .....	4-16
<b>Organizations .....</b>	<b>5-1</b>
Overview .....	5-2
Typical Duties of Organization Administrators .....	5-3
Managing Organization Members, Groups, and Roles .....	5-4
Creating, Editing, and Managing Organization Folders and Documents .....	5-5
Managing Organization-level Types and Attributes.....	5-5
Managing Organization Templates .....	5-6
Managing Organization-level Object Initialization Rules .....	5-6
Auditing Activities Within the Organization.....	5-7
Creating and Managing Access Control Policies .....	5-7
Viewing and Managing Access Control Rules for Objects .....	5-7
Creating and Managing Profiles .....	5-8
Configuring Numbering and Versioning Schemes .....	5-8
Monitoring and Managing Viewable Publishing.....	5-8
Viewing Reports .....	5-8
Importing and Exporting Information .....	5-9
Purging, Archiving, and Restoring Jobs .....	5-9
Managing Preferences .....	5-9
Undoing a User Checkout .....	5-9
Out-of-the-box Organization Templates.....	5-10
Context Structure .....	5-10
Context Participation .....	5-11
Context Access Control Policies .....	5-11
Context Data .....	5-16
Creating an Organization Context.....	5-17
Owning Organization Principals.....	5-19
Setting Up Domains for Use with Owning Organization Principals .....	5-20
Creating and Updating Owning Organization Principals .....	5-20
Using the Organization Utilities Page .....	5-22

Changing an Established Internet Domain .....	5-22
Using the OrganizationSync Utility for User Organization Changes.....	5-23
Best Practices.....	5-25
For All Windchill Solutions .....	5-25
For Windchill Solutions with Products and Libraries .....	5-25
For Windchill Solutions with Projects and Programs .....	5-26
<b>Projects and Programs .....</b>	<b>6-1</b>
Overview.....	6-2
Typical Duties of Project and Program Managers .....	6-2
Creating and Editing Projects and Programs.....	6-3
Overriding User Profiles.....	6-3
Managing Team Members and Roles.....	6-4
Creating, Editing, and Managing Documents and Folders .....	6-5
Creating, Editing, and Managing Activities, Deliverables, Resources, and Action Items.....	6-5
Limiting Edit Privileges for All Action Items .....	6-6
Managing Templates .....	6-6
Importing and Exporting Information.....	6-6
Undoing a User Checkout.....	6-7
Viewing and Managing Access Policies .....	6-7
Managing Utilities .....	6-8
Out-of-the-box Project and Program Templates.....	6-10
Configuring Part and Document Relationships.....	6-13
Part to Document Relationships .....	6-13
Part to Part Relationships (Projects Only) .....	6-16
Document to Document Relationships .....	6-18
<b>Products and Libraries .....</b>	<b>7-1</b>
Overview.....	7-2
Typical Duties of Product and Library Administrators.....	7-2
Managing Team Members and Roles.....	7-3
Managing Folders .....	7-4
Managing Templates .....	7-5
Managing Object Initialization Rules.....	7-5
Viewing and Managing Access Policies .....	7-5
Configuring Numbering and Versioning Schemes.....	7-6
Managing the Life Cycle of Parts, Documents, CAD Documents, and Dynamic Documents.....	7-7
Managing Viewable Publishing.....	7-7
Managing Preferences.....	7-7
Undoing a User Checkout.....	7-8

Importing and Exporting Information .....	7-8
Configuring External Vaults or Replication Sites to Optimize Performance .....	7-8
Tracking Changes .....	7-9
Out-of-the-box Product and Library Context Templates .....	7-9
Product Design Template .....	7-11
General Product and General Library Templates .....	7-33
Updating Access Control Rules .....	7-42
Creating a Product .....	7-42
Creating a Library .....	7-43
Administering Teams .....	7-45
Using the Product and Library Utilities Page .....	7-46
Configuring Products or Libraries for Dynamic Documents .....	7-46
Configuring Part and Document Relationships .....	7-48
Part to Document Relationships .....	7-48
Part to Part Relationships .....	7-51
Document to Document Relationships .....	7-53
Best Practices For Object Initialization Rules .....	7-53
<b>Principals (Users, Groups, and Organizations) .....</b>	<b>8-1</b>
Overview .....	8-2
Windchill Users .....	8-3
Windchill Groups .....	8-3
Windchill Organizations .....	8-4
Working with LDAP Directory Services .....	8-5
Searching for Principals in Administrative Clients .....	8-8
Using the Principal Administrator .....	8-9
Best Practices for Windchill PDMLink and Windchill ProjectLink .....	8-11
Best Practices When Maintaining a Directory Service Outside of Windchill .....	8-11
Managing Users .....	8-12
Managing User-defined Groups .....	8-18
Managing Organizations .....	8-22
Understanding Principals (Users, Groups, and Organizations) .....	8-24
Best Practices for Assigning Domains to Principals .....	8-25
Receiving Administrative Notifications .....	8-26
Managing the Principal Cache .....	8-27
Automatically Purging Entries from the Principal Cache .....	8-27
Manually Purging Entries from the Principal Cache .....	8-28
Maintaining the Connections between Principal Objects and their Directory Service Entries ..	8-28



Profile Management .....	8-29
Creating Profiles .....	8-29
Profiles as a Visibility Control Mechanism .....	8-30
Default Profile Behavior for a New User .....	8-30
Global Default Settings .....	8-31
Overriding Profiles in an Application Context .....	8-31
Default Visibility for Application Context Managers .....	8-31
Out-of-the-Box Profiles .....	8-32
<b>Teams .....</b>	<b>9-1</b>
Overview .....	9-2
Context Teams .....	9-3
Local Teams .....	9-4
Shared Teams .....	9-6
System Groups Associated with Teams .....	9-9
Restricting the Visibility of Actions in the User Interface through Teams.....	9-10
Team Templates.....	9-12
Out-of-the-Box Team Templates .....	9-13
Team Templates and Object Types.....	9-15
Team Roles Resolution .....	9-17
Defining Team Properties and Roles.....	9-26
Predefined Roles .....	9-27
Assigning Participants to Team Roles .....	9-28
Manually Selecting and Resetting Team Templates .....	9-29
Team Best Practices .....	9-30
Using Local and Shared Teams .....	9-30
Using Team Templates.....	9-30
Refreshing User-Defined Groups .....	9-30
<b>Context Templates .....</b>	<b>10-1</b>
Creating Context Templates .....	10-2
Organization Templates.....	10-2
Product and Library Context Templates .....	10-3
Program and Project Context Templates.....	10-3
Methods of Creating Additional Context Templates .....	10-4
ZIP File Contents of an Exported Context Template .....	10-6
Required Contents of ZIP File Used for Importing a Context Template .....	10-7
Contents of Top-level XML File for Imported Templates .....	10-8
Creating Business XML Files for Context Templates .....	10-9

Managing Context Templates .....	10-10
Filtering Template Visibility.....	10-11
Enabling Templates.....	10-11
<b>Object Initialization Rules .....</b>	<b>11-1</b>
Overview .....	11-2
Loaded Object Initialization Rules.....	11-3
Adding and Changing Object Initialization Rules .....	11-3
Accessing the Object Initialization Rules Administrator .....	11-3
How Object Initialization Rules Work .....	11-4
Determining the Composite Rule .....	11-5
Specifying Rules in the Object Initialization Rules Administrator.....	11-7
Defining the Content of XML Documents used for Object Initialization Rules .....	11-8
Windchill Types in AttributeValues .....	11-9
Object Types and Attributes Used with AttrValue and AttrConstraint .....	11-9
Algorithms Used with AttrValue.....	11-10
Algorithms Used with AttrConstraint.....	11-13
Conditional Logic Algorithms.....	11-16
Specifying Object Initialization Rules in a Context Template .....	11-20
Customizing Rules .....	11-20
Custom Rule Algorithms.....	11-20
Additional Types for AttrValue.....	11-21
Additional Type and Attribute Pairs for AttrConstraint.....	11-21
Changing Numbering Schemes .....	11-21
CAD Document Numbering.....	11-22
Dynamic Document Numbering .....	11-23
Additional Information about Numbering .....	11-23
Changing Versioning Schemes.....	11-24
Harvard Series Versioning Schemes .....	11-25
Setting Up a File-based or State-based Versioning Scheme .....	11-34
Using Versioning Schemes that Conform to the U.S. Military Specification.....	11-36
Setting up the MilSpec Multicharacter Series.....	11-36
<b>Access Control .....</b>	<b>12-1</b>
Overview .....	12-2
About Access Control Policy Rules.....	12-3
Creating and Managing Access Control Rules .....	12-4
Setting Permissions .....	12-6
Setting Permissions through the Policy Administrator.....	12-8
Setting Permissions through the Access Table.....	12-10

About Access Control Lists.....	12-10
Deriving ACLs from Access Control Policies .....	12-11
How ACLs Work .....	12-12
Distributed Administration of Policy Rules .....	12-15
About Default Access Control Policy Rules.....	12-16
Managing Access to Enterprise Information.....	12-17
Domain Administered Information .....	12-18
Policy and Ad Hoc Access Controlled Information .....	12-19
Content Holder Information.....	12-19
Foldered Information.....	12-20
Life-Cycle Managed Information.....	12-24
Example Permissions Needed for Moving a Document .....	12-26
Example Permissions Needed for Creating a Part in a Shared Cabinet .....	12-27
Access to Cabinets .....	12-27
Access Control Strategies for Life-Cycle Managed Objects.....	12-29
Combining Access Control Strategies for Cabinets and Life-Cycle Managed Objects .....	12-29
<b>Type and Attribute Manager.....</b>	<b>13-1</b>
Overview of Types and the Runtime Typing Capability .....	13-2
Effects of Deploying a New Type.....	13-3
Using Typing in Conjunction with Classification .....	13-3
When to Use Typing and When to Use Modeling.....	13-5
Localizing New Type Definitions.....	13-6
Migrating Existing Type Instances to a New Type Definition.....	13-7
Overview of the Type and Attribute Manager Utility .....	13-7
Accessing the Type and Attribute Manager Utility .....	13-9
Type Manager Tab .....	13-10
Attribute Definition Manager Tab .....	13-12
Measurement System Manager Tab .....	13-15
Quantity of Measure Manager Tab .....	13-16
Supported Units .....	13-17
Administrator Roles .....	13-18
Using the Type and Attribute Manager.....	13-19
Creating a Soft Type.....	13-19
Creating a Soft Attribute .....	13-22
Adding an Attribute to a Type .....	13-23
Setting Constraints on Attributes .....	13-25
Editing a Type Definition.....	13-28
Deleting a Type.....	13-30
Out-of-the-Box Constraints .....	13-30

Specifying Windchill Types .....	13-35
Managing Types and Attributes for Parts, Documents, and Change Objects.....	13-36
Excluding Types from a Types List .....	13-37
Excluding Types from Simple and Advanced Search Pages .....	13-38
Understanding Soft Attribute Behavior on Object Create and Update Windows, and on Information Pages .....	13-38
Managing Types and Attributes for CAD Documents .....	13-39
Renaming the CAD Document Soft Type.....	13-40
Additional Soft Types Related to CAD Documents .....	13-41
Mapping Attributes to CAD Parameters .....	13-42
Setting Attribute Access in CAD Applications and Windchill .....	13-43
Managing Types and Attributes for Dynamic Documents .....	13-43
Renaming Dynamic Document and Burst Configuration Soft Types.....	13-44
Creating Additional Soft Types for Use with Arbortext Editor and the Arbortext Publishing Engine .....	13-44
Specifying the Logical Identifier for Types Used with Arbortext .....	13-46
Defining Windchill Soft Attributes for Attributes Used with Arbortext .....	13-46
Best Practices .....	13-47
Working with Multiple Organization Contexts.....	13-47
Associating Soft Types with the Site and Organization Contexts.....	13-48
Restricting the Use of Soft Types and Soft Attributes .....	13-49
For Searchable Types .....	13-51
Glossary Terms for the Type and Attribute Manager Utility .....	13-51
<b>Reporting .....</b>	<b>14-1</b>
Overview .....	14-2
Reports Page .....	14-2
Related Reports for Products .....	14-4
Report Manager .....	14-4
Change Monitor Reports .....	14-5
Auditing Administration Reports .....	14-6
Windchill Business Reporting .....	14-6
Viewing Reports with Windchill Business Reporting .....	14-7
Accessing Windchill Business Reporting .....	14-7
Configuring Windchill Business Reporting .....	14-8
Accessing Documentation within Windchill Business Reporting .....	14-8

Authoring Reports.....	14-9
Query Builder.....	14-9
Info*Engine Tasks.....	14-9
Windchill Business Report Author .....	14-10
Editing Reports .....	14-10
Searching for Reports.....	14-11
Monitoring Reports .....	14-11
<b>Life Cycles .....</b>	<b>15-1</b>
Overview.....	15-2
Accessing the Life Cycle Administrator .....	15-2
The Life Cycle Model.....	15-2
Windchill Solutions.....	15-3
Life Cycle States.....	15-3
Basic and Advanced Life Cycles .....	15-4
Basic Life Cycles .....	15-4
Advanced Life Cycles .....	15-5
Managing Life Cycle Processes .....	15-5
Promotion Process .....	15-7
Out-of-the-box Life Cycle Templates.....	15-13
Windchill PDMLink.....	15-14
Windchill ProjectLink.....	15-16
About Life Cycle Iteration .....	15-17
Viewing Iteration History .....	15-17
Creating a Life Cycle Template .....	15-17
Life Cycle Properties.....	15-19
Defining Life Cycle Phases and Gates .....	15-21
Importing and Exporting Life Cycle Templates .....	15-40
Overriding and Reassigning Life Cycle and Team Templates .....	15-42
Manual Selection of Life Cycle and Team Templates .....	15-43
Reassigning Life Cycles and Resetting Team Templates .....	15-43
Defining Additional Life Cycle States.....	15-43
Access Control for Life Cycle Administration.....	15-44
Best Practices.....	15-44
Life Cycle Support in Windchill ProjectLink .....	15-45
Managing Promotion Processes in Windchill PDMLink .....	15-46
Life Cycle Teams in ProjectLink .....	15-48
Restrictions on Moving Objects Between Contexts .....	15-48

<b>Workflow.....</b>	<b>16-1</b>
Overview .....	16-2
Accessing the Workflow Administrator Utility .....	16-3
Managing Workflow Security .....	16-3
Workflow Creators.....	16-4
Restricting Workflow-Embedded Java Code.....	16-5
Enabling Restricted Access to Workflow Administrators Group.....	16-7
Workflow Iteration .....	16-8
Testing an Edited Workflow Process Template.....	16-8
Using the Workflow Process Editor.....	16-8
Working with Workflow Templates .....	16-9
Navigating a Process Diagram.....	16-12
Placing Process Nodes .....	16-12
Declaring Variables .....	16-16
Defining an Assigned Activity .....	16-19
Defining a Subprocess .....	16-34
Defining Connectors.....	16-35
Defining Links.....	16-37
Importing and Exporting Workflow Templates .....	16-39
Importing and Exporting Workflow Templates.....	16-40
Process Manager Toolbar Access Control .....	16-41
Viewing Workflow History .....	16-42
Selecting Events.....	16-42
Using the Workflow History Viewer .....	16-44
Workflow Instance States .....	16-49
Domain-based Workflows .....	16-50
Out-of-the-Box Workflow Templates.....	16-51
Change Management Workflows .....	16-52
Out-of-the-Box Process Images .....	16-59
Workflow Template Execution Flags.....	16-60
Process Flags.....	16-61
Activity Flags .....	16-65
Both Process and Activity Flags.....	16-67
Modifying Execution Flags .....	16-68
Using Task Form Templates in a Workflow .....	16-71
Creating Task Form Templates with Adobe Forms Software.....	16-71
Electronic Signatures .....	16-72
Setting Up for Electronic Signatures .....	16-72
Requiring Electronic Signatures in a Workflow .....	16-73

Best Practices.....	16-75
Access Control.....	16-75
Using a Single Workflow in a Life Cycle Having Multiple States.....	16-76
Workflow Process Support in Windchill ProjectLink .....	16-76
<b>Views and View Associations .....</b>	<b>17-1</b>
Overview.....	17-2
Accessing the View Administrator .....	17-2
Views and View Associations .....	17-2
Managing Views and View Associations .....	17-3
<b>Indexing.....</b>	<b>18-1</b>
Overview.....	18-2
About Indexing Rules .....	18-2
Creating and Managing Indexing Rules .....	18-3
About Indexing Policy .....	18-4
Considerations for Establishing Indexing Rules .....	18-4
About Indexing Lists .....	18-4
Defining a Collection.....	18-5
<b>Notifications.....</b>	<b>19-1</b>
Overview.....	19-2
About Notification Rules .....	19-2
Creating and Managing Notification Rules .....	19-3
Notification Lists .....	19-4
Object-based and Context-based Subscriptions .....	19-5
About Notification .....	19-6
<b>Visualization Services.....</b>	<b>20-1</b>
Overview.....	20-2
File Types .....	20-2
Architecture .....	20-3
Troubleshooting.....	20-6
Windchill Visualization Service Loader .....	20-6
Publishing CAD Documents .....	20-9
Visualization Collaboration .....	20-12
Creating and Enabling Server-side Watermarks .....	20-13
Creating Server-side Watermarks .....	20-13
Enabling Server-side Watermarks on the Server .....	20-16
Exporting Watermarks for ProductView.....	20-18

Copy Forward Functionality .....	20-19
Copying Representations Forward .....	20-19
Copying Markups Forward .....	20-19
Copy Forward Properties .....	20-20
Using Publish Rules .....	20-23
About Publish Rules .....	20-23
Enabling Publish Rules .....	20-23
Evaluating Publish Rules .....	20-24
Post-Publishing .....	20-33
Examples of Publish Rules XML Files .....	20-37
Creating Publish Rules .....	20-40
Windchill Visualization Service Properties .....	20-45
<b>Auditing .....</b>	<b>21-1</b>
Overview of the Auditing Event Framework .....	21-2
Out-of-the-Box Configuration for Audit Event Recording .....	21-2
Supported Auditing Events .....	21-3
Service Events .....	21-3
Summary Events .....	21-6
Attributes Captured with Events .....	21-8
Accessing the Auditing Administration Utility .....	21-12
Audit Report Queries .....	21-12
Viewing and Generating Report Queries .....	21-13
Purging the Audit Log .....	21-13
License Usage Reporting .....	21-14
Organization Usage Reporting .....	21-15
Example Audit Report Query .....	21-16
Creating an Audit Report Query .....	21-16
Example of Exporting a Saved Query .....	21-19
<b>Creating XML Files for Templates and Shared Teams .....</b>	<b>A-1</b>
Overview .....	A-2
Creating Business XML Files for Organization Templates .....	A-2
OrgStructure Element .....	A-3
ContainerTemplate Element .....	A-4
loadXMLFile Element .....	A-5
TypeBasedRule Element .....	A-5
AccessControlRule Element .....	A-5
ProjectEnumType Element .....	A-6
DiscussionForumTemplate Element .....	A-6
NotebookTemplate Element .....	A-6



Creating Business XML Files for Product Templates .....	A-7
SharedTeamDef Element .....	A-8
OrgStructure Element .....	A-8
DiscussionForumTemplate Element .....	A-8
ExportedRoleMemberMap Element .....	A-8
ExportedGuestMembers Element .....	A-8
ExportedTemplateFiltering Element .....	A-8
AccessControlRule Element .....	A-8
TypeBasedRule Element .....	A-9
NotebookTemplate Element .....	A-9
AdHocACLEntrySet Element .....	A-9
PreferenceInstance Element .....	A-9
Creating Business XML Files for Library Templates .....	A-10
SharedTeamDef Element .....	A-11
OrgStructure Element .....	A-11
DiscussionForumTemplate Element .....	A-11
ExportedRoleMemberMap Element .....	A-11
ExportedGuestMembers Element .....	A-11
ExportedTemplateFiltering Element .....	A-11
AccessControlRule Element .....	A-11
TypeBasedRule Element .....	A-12
NotebookTemplate Element .....	A-12
AdHocACLEntrySet Element .....	A-12
PreferenceInstance Element .....	A-12
Creating Business XML Files for Program and Project Templates .....	A-13
SharedTeamDef Element .....	A-14
ExportedRoleMemberMap Element .....	A-14
ExportedGuestMembers Element .....	A-14
ExportedTemplateFiltering Element .....	A-14
OrgStructure Element .....	A-14
TypeBasedRule Element .....	A-15
AccessControlRule Element .....	A-15
ProductStructure Element .....	A-15
shareMapSet Element .....	A-23
projectPlan Element .....	A-25
ProjectEnumType Element .....	A-45
DiscussionForumTemplate Element .....	A-46
NotebookTemplate Element .....	A-47

AdHocACLEntrySet Element.....	A-47
PreferenceInstance Element .....	A-47
Creating Business XML Files for Shared Teams .....	A-48
ExportedRoleMemberMap Element .....	A-49
ExportedGuestMembers Element .....	A-49
AccessControlRule Element.....	A-49
AdHocACLEntrySet Element.....	A-49
Common XML Elements Used in Multiple Business XML Files .....	A-50
Common AccessControlRule Element .....	A-50
Common AdHocACLEntrySet Element .....	A-52
Common ExportedGuestMembers Element.....	A-54
Common ExportedRoleMemberMap Element.....	A-55
Common ExportedTemplateFiltering Element .....	A-57
Common loadXMLFile Element.....	A-58
Common NotebookTemplate Element .....	A-58
Common OrgStructure Element .....	A-59
Common PreferenceInstance Element .....	A-66
Common ProjectEnumType Element .....	A-67
Common SharedTeamDef Element .....	A-68
Common TypeBasedRule Element.....	A-69
Common WTPPrincipalReference Element.....	A-72

## Index

# Change Record

The following tables describe the major changes made in the guide for each update.

**Table 1 Changes for 9.0**

Chapter	Description
Entire guide	Removed references to Windchill Foundation & PDM solution as this solution is no longer supported.
<a href="#">Getting Started</a>	Updates to this chapter include: <ul style="list-style-type: none"><li>• Added Arbortext Content Manager and ProINTRALINK 9.0 information to <a href="#">Logging On as the Administrator</a>.</li><li>• Updated <a href="#">Working With the Initial Organization Context</a>.</li><li>• Added a new section <a href="#">Establishing End Users</a>.</li><li>• Updated <a href="#">The Next Steps</a> section.</li></ul>
<a href="#">Administration Overview</a>	Updates to this chapter include: <ul style="list-style-type: none"><li>• Added program context references.</li><li>• Updated <a href="#">Managing User Access to Data</a> with new shared team and access control information.</li></ul>

Chapter	Description
<a href="#">Contexts</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Updating what is installed in the site context throughout the chapter.</li> <li>• Moving context template and object initialization rules content to new chapters.</li> <li>• Added new sections: <a href="#">Administering Preferences</a> and <a href="#">Administering Saved Searches</a>.</li> </ul>
<a href="#">Site</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Added new sections: <ul style="list-style-type: none"> <li>– <a href="#">Managing Workflow Security</a></li> <li>– <a href="#">Creating and Managing Profiles</a></li> <li>– <a href="#">Auditing System Information</a></li> <li>– <a href="#">Managing Calendar Settings</a></li> <li>– <a href="#">Overriding and Reassigning Life Cycle and Team Templates</a></li> </ul> </li> <li>• Moved main discussion of creating organizations to the <a href="#">Organizations</a> chapter.</li> <li>• Added and updated information regarding preferences and the new Preference Manager.</li> </ul>

Chapter	Description
<a href="#">Organizations</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Updated the section: <a href="#">Auditing Activities Within the Organization</a></li> <li>• Added a new sections: <ul style="list-style-type: none"> <li>– <a href="#">Creating and Managing Profiles</a></li> <li>– <a href="#">Owning Organization Principals</a></li> </ul> </li> <li>• Updated organization context and organization principal information throughout.</li> <li>• Added information on preferences and the new Preference Manager.</li> <li>• Updated creating organization information.</li> </ul>
<a href="#">Projects and Programs</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Added new programs context.</li> <li>• Added new section: <a href="#">Undoing a User Checkout</a>.</li> <li>• Updated template information.</li> <li>• Added information preferences and new Preference Manager.</li> </ul>

Chapter	Description
<a href="#">Products and Libraries</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Updated template information</li> <li>• Added new section: <a href="#">Undoing a User Checkout</a></li> <li>• Added new section: <a href="#">Tracking Changes</a></li> <li>• Updated <a href="#">Configuring Part and Document Relationships</a> with new preference information.</li> <li>• Added information on new Preference Manager</li> </ul>
<a href="#">Principals (Users, Groups, and Organizations)</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Changed the chapter title from Administering Principals to <a href="#">Principals (Users, Groups, and Organizations)</a></li> <li>• Added new sections: <ul style="list-style-type: none"> <li>– <a href="#">Working with LDAP Directory Services</a></li> <li>– <a href="#">Associating Users with Profiles</a></li> <li>– <a href="#">Editing the Domain of a User</a></li> <li>– <a href="#">Profile Management</a></li> </ul> </li> <li>• Added information about editing the user's domain and the personal cabinet.</li> <li>• Changed the term "container" to "context" throughout the chapter.</li> <li>• Added a preference for turning off the automatic update for Synchronize Domains for User Organization Changes.</li> <li>• Updated the section: <a href="#">Refreshing Team Membership for Users and User-defined Groups</a></li> </ul>

Chapter	Description
<a href="#">Teams</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Reorganized and updated the chapter.</li> <li>• Added the information in the following sections for local and shared teams: <ul style="list-style-type: none"> <li>– <a href="#">Context Teams</a></li> <li>– <a href="#">Team Best Practices</a></li> </ul> </li> </ul>
<a href="#">Context Templates</a>	<p>New chapter includes context templates content that had been in <a href="#">Contexts</a> chapter as well as the following new section:</p> <p><a href="#">Managing Context Templates</a></p> <p>In the <a href="#">Creating Context Templates</a> section, added ability to export the following:</p> <ul style="list-style-type: none"> <li>• Template visibility</li> <li>• Preferences</li> </ul>

Chapter	Description
<a href="#">Object Initialization Rules</a>	<p>New chapter includes rules content that had been in the <a href="#">Contexts</a> chapter as well as major updates in the following sections:</p> <ul style="list-style-type: none"> <li>• <a href="#">Overview</a></li> <li>• <a href="#">Specifying Rules in the Object Initialization Rules Administrator</a></li> <li>• <a href="#">Defining the Content of XML Documents used for Object Initialization Rules</a></li> <li>• <a href="#">Customizing Rules</a></li> <li>• <a href="#">Changing Numbering Schemes</a></li> <li>• <a href="#">Changing Versioning Schemes</a></li> </ul> <p>New sections include:</p> <ul style="list-style-type: none"> <li>• <a href="#">Algorithms Used with AttrConstraint</a></li> <li>• <a href="#">Custom Rule Algorithms</a></li> <li>• <a href="#">Accommodating Legacy Values in Revision Labels</a></li> </ul>
<a href="#">Access Control</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Changes to all sections to support the new permissions</li> <li>• New ad hoc rules user interface described in <a href="#">Creating and Managing Access Control Rules</a></li> <li>• New permissions described in <a href="#">Setting Permissions</a></li> <li>• New permissions included in <a href="#">Managing Access to Enterprise Information</a></li> </ul>



Chapter	Description
<a href="#">Type and Attribute Manager</a>	<p>Updates to this chapter include:</p> <ul style="list-style-type: none"> <li>• Removed the section about Displaying Type Lists in a hierarchical format.</li> <li>• Updated the preferences in the following sections: <ul style="list-style-type: none"> <li>– <a href="#">Managing Types and Attributes for Parts, Documents, and Change Objects.</a></li> <li>– <a href="#">Renaming the CAD Document Soft Type.</a></li> <li>– <a href="#">For Searchable Types.</a></li> </ul> </li> <li>• New sections include: <ul style="list-style-type: none"> <li>– <a href="#">Overview of the Type and Attribute Manager Utility.</a> This section includes information about the: <ul style="list-style-type: none"> <li>• <a href="#">Type Manager Tab</a></li> <li>• <a href="#">Attribute Definition Manager Tab</a></li> <li>• <a href="#">Quantity of Measure Manager Tab</a></li> </ul> </li> <li>– <a href="#">Setting Attribute Access in CAD Applications and Windchill.</a></li> <li>– <a href="#">Excluding Types from Simple and Advanced Search Pages.</a></li> <li>– <a href="#">Working with Multiple Organization Contexts</a></li> </ul> </li> </ul>

Chapter	Description
<a href="#">Type and Attribute Manager</a> cont'd.	<ul style="list-style-type: none"> <li>• Changed the section previously named "Type Manager" to <a href="#">Using the Type and Attribute Manager</a>. All procedures about using the Type Manager have been updated.</li> <li>• Added new constraints (Hidden, Lower Case, and Value Hidden) to the constraints table and renamed the section from "Setting Constraints" to <a href="#">Out-of-the-Box Constraints</a>.</li> <li>• Updated the section: <a href="#">Restricting the Use of Soft Types and Soft Attributes</a>.</li> </ul>
<a href="#">Reporting</a>	<p>New chapter includes reporting content that had been in other chapters, and new content for the Windchill Business Reporting functionality.</p>

Chapter	Description
<a href="#">Life Cycles</a>	<p>Updates to the chapter include:</p> <ul style="list-style-type: none"> <li>• Added new sections: <ul style="list-style-type: none"> <li>– <a href="#">Overriding and Reassigning Life Cycle and Team Templates</a></li> <li>– <a href="#">Life Cycle Teams in ProjectLink</a></li> </ul> </li> <li>• Reorganized the Overview section, as well as the overall structure of the chapter.</li> <li>• Updated the transition preferences for the out-of-the-box workflow processes for promote transition.</li> <li>• Updated the out-of-the-box life cycle templates section.</li> <li>• Updated the description for the Roles phase property.</li> <li>• Changed life cycle history to maturity history.</li> </ul>

Chapter	Description
<a href="#">Workflow</a>	<p>Updates to the chapter include:</p> <ul style="list-style-type: none"> <li>Added a new section: <ul style="list-style-type: none"> <li><a href="#">Managing Workflow Security</a></li> <li><a href="#">Using Task Form Templates in a Workflow</a></li> </ul> </li> <li>Moved the existing content for Enabling Restricted Access to Workflow Administrators Group to <a href="#">Managing Workflow Security</a> section</li> <li>Added a new paragraph under <a href="#">Placing Process Nodes</a> for deleting workflow objects.</li> <li>Added content for Set Up Participants activity.</li> <li>Added content for the new Role Setup tab and the Resource Pool tab.</li> <li>Updated the <a href="#">Defining a Subprocess</a> section.</li> <li>Updated all out-of-the-box workflow template graphics and steps.</li> <li>Added a note to the Variables section and under <a href="#">Change Management Workflows</a> about how the activity variables display in the task details page.</li> </ul>
<a href="#">Views and View Associations</a>	<p>The entire chapter has been rewritten based on the new View Administrator utility.</p>
<a href="#">Visualization Services</a>	<p>The following sections have been added:</p> <ul style="list-style-type: none"> <li><a href="#">Creating and Enabling Server-side Watermarks</a></li> <li><a href="#">Using Publish Rules</a></li> </ul>

Chapter	Description
<a href="#">Auditing</a>	<p>The entire chapter has been rewritten based on the new functionality.</p> <p>The chapter title has been changed from Audit Reporting to Auditing.</p>
<a href="#">Creating XML Files for Templates and Shared Teams</a>	<p>Throughout the appendix, updated the XML elements to reflect the changes in standardX10.dtd for context templates. Notable changes include the following new elements:</p> <p style="padding-left: 40px;">SharedTeamDef ExportedGuestMembers ExportedTemplateFiltering AccessControlRule AdHocACLEntrySet PreferenceInstance</p> <p>AccessControlRule replaces ACLRule and AdHocACLEntrySet replaces AdHocAclEntrySet.</p> <p>Added the <a href="#">Creating Business XML Files for Shared Teams</a> section that describes the elements used for importing shared teams.</p>

**Table 2 Changes for Windchill 8.0, Maintenance Release M040**

Chapter	Description
Administering Containers	<ul style="list-style-type: none"> <li>Updated the File-based Scheme section to include versioning name rules.</li> <li>Updated the <a href="#">Creating Domains</a> section to include forward slash in domain names information.</li> </ul>

Chapter	Description
Administering the Site	<ul style="list-style-type: none"> <li>Updated the <a href="#">Business Object Uniqueness Considerations</a> section.</li> <li>Updated the <a href="#">Managing Searches</a> section with latest version searching information, and reference to instructions for adding new searchable types.</li> <li>Added caution to the <a href="#">Creating and Managing Organizations</a> section.</li> <li>Added a new section to <a href="#">Best Practices For Windchill Solutions with Products and Libraries</a>.</li> <li>Added a new <a href="#">Overriding and Reassigning Life Cycle and Team Templates</a> section to <a href="#">Best Practices For Windchill Solutions with Products and Libraries</a>.</li> </ul>
Administering Organizations	<ul style="list-style-type: none"> <li>Updated the <a href="#">Using the OrganizationSync Utility for User Organization Changes</a> section.</li> </ul>
Administering Products and Libraries	<ul style="list-style-type: none"> <li>Updated the <a href="#">Revised or Saved Part to Related Document</a> section of <a href="#">Part to Document Relationships</a> to clarify that removing copy rules does not affect revise actions.</li> <li>Added new Managing Related Reports section.</li> </ul>

Chapter	Description
Administering Principals	<ul style="list-style-type: none"> <li>Updated the <a href="#">Changing the Organization to which a User Belongs</a> section.</li> <li>Added a new <a href="#">Working with User-defined Groups that are Maintained in a Directory Service</a> section to <a href="#">Managing User-defined Groups</a>.</li> <li>Added a new section, <a href="#">Refreshing Team Membership for Users and User-defined Groups</a>.</li> </ul>
Using Types and the Type Manager	<ul style="list-style-type: none"> <li>Added statement about making attributes searchable to the "Adding Attributes" section.</li> <li>Updated the following sections to support the use of the logical identifier when specifying types in Windhchill and Arbortext: <ul style="list-style-type: none"> <li>– <a href="#">Specifying Windchill Types</a></li> <li>– <a href="#">Managing Types and Attributes for Parts, Documents, and Change Objects</a></li> <li>– <a href="#">Managing Types and Attributes for CAD Documents</a></li> <li>– <a href="#">Managing Types and Attributes for Dynamic Documents</a></li> </ul> </li> </ul>

Chapter	Description
Administering Life Cycles	<ul style="list-style-type: none"> <li>Added property file for displaying annotations during promotion approval to the <a href="#">Promotion Process</a> section.</li> <li>Changed the description of requirements for Roles in the phase properties table in the <a href="#">Defining Life Cycle Phases and Gates</a> section.</li> <li>Added preferences for manually selecting and reassigning life cycle and resetting team templates in the <a href="#">Overriding and Reassigning Life Cycle and Team Templates</a> section.</li> </ul>
Administering Visualization Services	<ul style="list-style-type: none"> <li>Updated <a href="#">Windchill Visualization Service Properties</a> section with new mark out of date properties.</li> <li>Added new <a href="#">Copy Forward Functionality</a> section.</li> </ul>

**Table 3 Changes for Windchill 8.0, Maintenance Release M030**

Chapter	Description
Getting Started	Added new section, <a href="#">Regarding Arbortext Content Manager</a> , with information on Arbortext Content Manager support.



Chapter	Description
Administering Containers	<p>Updated the Administering Object Initialization Rules section:</p> <ul style="list-style-type: none"> <li>• Windchill ProjectLink does not use the default values set for folder path.</li> <li>• Customization is not supported for rule algorithms.</li> <li>• Specify types in the objType attribute using the logical identifier of the type.</li> </ul> <p>Updated the <a href="#">Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects</a> section.</p> <ul style="list-style-type: none"> <li>• Do not use numeric values other than 0 for the pad character.</li> </ul> <p>Updated the <a href="#">Administering the Versioning of Parts, Documents, and CAD Documents</a> section.</p> <ul style="list-style-type: none"> <li>• Do not remove series or series values that have been used.</li> <li>• Revision labels must be unique within the entire versioning scheme.</li> <li>• Using versioning schemes that conform to the ASME Y 14.35M revision standard is described in the Updating Versioning Schemes to Conform to the ASME Y 14.35M Standard section.</li> </ul>
Administering the Site	<p>Added new section, <a href="#">Business Object Uniqueness Considerations</a>.</p>

Chapter	Description
Administering Organizations	<ul style="list-style-type: none"> <li>Added new section, <a href="#">Changing an Established Internet Domain</a>.</li> <li>Added multiorganization environment information to Creating Only One Organization Context in the <a href="#">Best Practices</a> section.</li> </ul>
Administering Projects	<ul style="list-style-type: none"> <li>Added Hiding the Guest Role section to <a href="#">Managing Team Members and Roles</a>.</li> <li>Added <a href="#">Limiting Edit Privileges for All Action Items</a> section to <a href="#">Typical Duties of Project and Program Managers</a>.</li> </ul>
Administering Principals	Removed creating similar organizations link information from <a href="#">Managing Organizations section</a> as this functionality is no longer supported.
Using Types and the Type Manager	<p>Added new <a href="#">For Searchable Types</a> section to <a href="#">Best Practices</a>.</p> <p>Added character requirements for creating types.</p>
Administering Audit Reports	Added note regarding use of the forward slash ( / ) as a directory separator to the Enabling Auditing section.

**Table 4 Changes for Windchill 8.0, Maintenance Release M020**

Chapter	Description
Administration Overview	Updated the <a href="#">Managing Data</a> section to include support for dynamic documents in Windchill PDMLink.

Chapter	Description
Administering Containers	<p>Updated the Windchill ProjectLink Context Templates section to include role access for teams.</p> <p>Updated the Access to Creating Additional Context Templates section to include the <b>Download</b> action.</p> <p>Added <a href="#">Understanding the Use of Versioning Schemes</a> section and added cautions to the <a href="#">Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects</a> and <a href="#">Administering the Versioning of Parts, Documents, and CAD Documents</a> sections about changing numbering and versioning schemes after objects have been created.</p> <p>Added name of Windchill PDMLink out-of-the-box state-based versioning file to the State-based Scheme section.</p>
Administering Products and Libraries	<p>Added wt.inf.team .displayContainerListForGuest property description to the <a href="#">Out-of-the-box Context Participation</a> section.</p> <p>Added <a href="#">Configuring Products or Libraries for Dynamic Documents</a> section.</p> <p>Added <a href="#">Configuring Part and Document Relationships</a> section.</p> <p>Added Allowing URLs as Content for Documents section.</p>
Administering Projects	<p>Added wt.inf.team .displayContainerListForGuest property description to the <a href="#">Managing Team Members and Roles</a> section.</p> <p>Added <a href="#">Controlling the Visibility of Actions</a> section to the <a href="#">Managing Team Members and Roles</a> section.</p>

Chapter	Description
Using Types and the Type Manager	<p>Updated the <a href="#">Overview of Types and the Runtime Typing Capability</a> section.</p> <p>Updated the <a href="#">Associating Soft Types with the Site and Organization Contexts</a> Best Practices section.</p> <p>Soft attributes can now be added to WTPartsUsageLink objects and soft type and soft attributes are supported for change objects as described in sections the Defining Additional Attributes and Restricting the Use of Soft Types and Soft Attributes.</p> <p>There are changes to how attributes for change objects are displayed. See section, Client Changes.</p> <p>Updated description of Immutable Constraint is in the Out-of-the-box Constraints section.</p> <p>Added following new sections:</p> <p><a href="#">Specifying Windchill Types</a></p> <p><a href="#">Managing Types and Attributes for Parts, Documents, and Change Objects</a></p> <p><a href="#">Managing Types and Attributes for CAD Documents</a></p> <p><a href="#">Managing Types and Attributes for Dynamic Documents</a></p>
Administering Notifications	<p>Added new <a href="#">Object-based and Context-based Subscriptions</a> section.</p>

Chapter	Description
Administering Life Cycles	<p>Added new Two Level Approval Routing advanced life cycle template to <a href="#">Windchill ProjectLink</a> section.</p> <p>Added location of resource files to the <a href="#">Defining Life Cycle Phases and Gates</a> section.</p> <p>Modified <a href="#">State-based Revision Sequences by Life Cycle State</a> section to update XML file information.</p>
Administering Teams and Roles	Updated <a href="#">Refreshing User-Defined Groups</a> section to clarify this functionality.
Administering Workflow Processes	Updated <a href="#">Access Control</a> section to remove limit restriction on ad-hoc access policies.
Creating Business XML Files for Templates	Updated the <a href="#">Common ExportedRoleMemberMap Element</a> section to add project template support for exporting and importing role access data.

**Table 5 Changes for Windchill 8.0, Maintenance Release M010**

Chapter	Description
Administering Containers	<p>In the <a href="#">Installed Site Context Policies</a> section, added the access control rule set for the RelationshipMap object.</p> <p>Updated the <a href="#">Administering Domains and Policies</a> section to better describe domains and the association of domains to folders. The ability to add top-level domains in application containers is now supported.</p> <p>In the <a href="#">Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects</a> section, added a description of the preference you can use to remove the ability to override the autogenerated part number.</p>
Administering Workflow Processes	<p>Added guidelines to the <a href="#">Declaring Variables</a> section.</p> <p>Updated the <a href="#">Process Flags</a> section to include the wt.workflow.engine.useLatestTemplateIteration property.</p>
Administering Views and View Associations	<p>Updated chapter to add the <b>[D]elete view</b> choice description and improve other choice descriptions.</p>
Administering Audit Reports	<p>Updated Enabling Auditing section to clarify the use of the configAudit.xml file.</p>

Chapter	Description
Creating Business XML Files for Templates	<p>Added support for the ProductStructure element in project template XML files in the <a href="#">ProductStructure Element</a> section.</p> <p>Updated the <a href="#">Common OrgStructure Element</a> section to support the parentDomain element, better describe the domainName element, and correct the examples provided.</p> <p>Updated the <a href="#">Common AccessControlRule Element</a> section to better describe the domainName element.</p>

**Table 6 Changes for Windchill 8.0**

Chapter	Description
Getting Started	<p>Updated the <a href="#">Logging On as the Administrator</a> and Creating the Initial Organization Context sections to reflect base data changes for Windchill PDMLink.</p> <p>Updated the <a href="#">Establishing Administrators</a> section with e-mail sender information.</p>
Administration Overview	<p>Updated domain diagrams for containers in the <a href="#">Managing User Access to Data</a> section.</p> <p>Updated the <a href="#">About the windchill Command</a> and <a href="#">About the xconfmanager Utility</a> sections.</p>

Chapter	Description
Administering Containers	<p>Added <a href="#">Assigning Domains to Folders in Solutions with Products and Libraries</a> section.</p> <p>Added <a href="#">Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects</a> section to provide an overview of numbering schemes.</p> <p>Added <a href="#">Administering the Versioning of Parts, Documents, and CAD Documents</a> section to document versioning schemes including the new state-based versioning.</p> <p>Added information about exporting and importing context templates in the Creating Context Templates section.</p> <p>Added information about the formatting required for object initialization rules in the Administering Object Initialization Rules section.</p> <p>Added information about the wt.org.client.applet.ActiveSearch property in the Searching for Principals section.</p>
Administering the Site	<p>Added information about using the Purge Manager in the <a href="#">Purge, Archive, and Restore Jobs</a> section.</p> <p>Added information about requiring primary content in Windchill PDMLink documents to the <a href="#">Creating and Managing Site Folders and Documents</a> section.</p> <p>Added best practices information in the <a href="#">Setting Object Initialization Rules</a> and <a href="#">Setting Up Enhanced Life Cycle Templates</a> sections.</p>



Chapter	Description
Administering Organizations	<p>Added information about using the Purge Manager in the <a href="#">Purging, Archiving, and Restoring Jobs</a> section.</p> <p>Updated the Best Practices section because tables are collapsed out-of-the-box.</p>
Administering Products and Libraries	<p>Updated <a href="#">Configuring Numbering and Versioning Schemes</a> section to add site-based versioning.</p> <p>Added <a href="#">Managing the Life Cycle of Parts, Documents, CAD Documents, and Dynamic Documents</a> section.</p> <p>Updated <a href="#">Out-of-the-box Product and Library Context Templates</a> section to document Product Design template.</p> <p>Updated <a href="#">Administering Teams</a> section to document reassigning tasks.</p> <p>Added Creating Custom Change Monitor Reports section.</p> <p>Added <a href="#">Best Practices For Object Initialization Rules</a> section for setting object initialization rules in individual product and library containers</p>
Administering Principals	<p>Added <a href="#">Searching for Users and Groups</a> section.</p> <p>Updated <a href="#">Managing Users</a> section.</p> <p>Updated chapter to reflect updated user interface.</p>
Using Types and the Type Manager	<p>Included information about mapping CAD documents.</p>
Administering Life Cycles	<p>Updated to include enhanced life cycle management.</p> <p>Added Basic and Advanced Life Cycles section.</p>
Administering Teams and Roles	<p>Added <a href="#">Refreshing User-Defined Groups</a> section.</p>

Chapter	Description
Administering Workflow Processes	<p>Added information about electronic signatures.</p> <p>Added information regarding two new promotion request workflows.</p> <p>Added <a href="#">Workflow Template Execution Flags</a> to document how to modify the workflow engine flags for an existing workflow.</p>
Administering Visualization Services	Updated table of Windchill Visualization Service properties and descriptions.
Creating Business XML Files for Templates	New appendix describing content of business XML files that can be used when creating an organization template or importing a product, library, or project template.

# About This Guide

The *Windchill Business Administrator's Guide* serves as a reference guide for Windchill business and application administrators.

The following table illustrates the responsibilities and skills of each type of administrator:

	<b>Business Administrator</b>	<b>Applications Administrator</b>
<b>Responsibilities</b>	Use Windchill to achieve specific business goals, such as setting up workflows and entering users, user-defined groups, ACLs, and so on.	Use Windchill applications to achieve goals appropriate for the application.
<b>Skills</b>	Understand the Windchill client and the business needs of enterprise users.	Understand the needs of particular application-user communities.

System administrators, who are responsible for keeping the system running, should refer to the *Windchill System Administrator's Guide*.

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.

## Overview

The *Windchill Business Administrator's Guide* describes responsibilities and roles of Windchill business and application administrators, providing conceptual and background information to help them understand the nature of administrative tasks.

## Chapter Contents

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

Chapter 1, Getting Started, provides a road map for getting your Windchill solution set up to be usable as a test system and provides some basic information about your Windchill environment.

Chapter 2, Administration Overview, provides a general overview of your installed Windchill runtime architecture and Windchill environment.

Chapter 3, Contexts, provides the overall details relating to working with contexts.

Chapter 4, Site, provides an overview for administering sites and describes the typical duties that a site administrator performs.

Chapter 5, Organizations, provides an overview for administering organizations and describes the typical duties that an organization administrator performs.

Chapter 6, Projects and Programs, provides an overview for administering projects and programs and describes the typical duties that an administrator performs in the project or program context.

Chapter 7, Products and Libraries, provides overview for administering products and libraries and describes the typical duties that an administrator performs in the product and library context.

Chapter 8, Principals (Users, Groups, and Organizations), describes the principals (user, group, and organization objects) that are used in your Windchill solution. It also provides details about managing the principals.

Chapter 9, Teams, describes the definition of teams and the mapping of team roles to principals.

Chapter 10, Context Templates, provides information on how to create and manage templates for contexts.

Chapter 11, Object Initialization Rules, introduces the concepts for object initialization rules and provides details on how to create and manage the rules, including rules for numbering and versioning.

Chapter 12, Access Control, describes access control policies, consisting of rules that govern access to objects. It includes the Windchill model for storage of enterprise information and strategies for creating access rules that best serve your site's security needs.

Chapter 13, Type and Attribute Manager, describes the definition of subtypes, attributes, and constraints.

Chapter 14, Reporting, describes the various uses for reporting within Windchill solutions, including the Windchill Business Reporting functionality.

Chapter 15, Life Cycles, describes life cycles, which define how an object matures and provides models for product commercialization.

Chapter 16, Workflow, describes workflow processes, which enable an organization to automate procedures in which information, tasks, and documents are passed among participants.

Chapter 17, Views and View Associations, describes view setups and view associations for your Windchill system.

Chapter 18, Indexing, describes indexing policies and specifies the indexes into which an object is to be entered when a specified event occurs, and it describes how to define and bulk-load index collections when setting up a new site or upgrading to a new release.

Chapter 19, Notifications, describes notification policies, which determine users and groups to be notified when specific events are applied to an object.

Chapter 20, Visualization Services, describes the configuration and troubleshooting of the visualization service.

Chapter 21, Auditing, provides information on how to enable and use the audit reporting feature.

Appendix A, Creating XML Files for Templates and Shared Teams, provides information to assist in creating business XML files that can be used when creating an organization template, or creating or importing a product, library, or project template.

## Related Documentation

The following documentation may also be helpful:

- *Windchill System Administrator's Guide*
- *Windchill Installation and Configuration Guide - Express*
- *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*
- properties.html file
- *Arbortext Content Manager User's Guide*
- *Pro/INTRALINK 9.0 User's Guide*
- *Getting Started with Pro/INTRALINK 9.0*
- *Getting Started Authoring Arbortext Documents*

If a guide you are interested in reading is not installed at your site, see [Documentation for PTC Products](#) below.

## 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](http://www.ptc.com/go/wc_pubs_feedback)





# 1

## Getting Started

This chapter provides a road map for getting your Windchill solution set up so that it is usable as a test system. The chapter also provides some basic information about the use of Windchill administrators and your Windchill environment.

PTC strongly recommends that all administrators read this chapter as well as [Administration Overview](#) and [Contexts](#) in their entirety. Later chapters assume that you are familiar with this information.

Topic	Page
Introduction .....	1-2
Logging On as the Administrator.....	1-3
Working With the Initial Organization Context.....	1-10
Establishing Administrators .....	1-12
Establishing End Users.....	1-15
The Next Steps .....	1-17

# Introduction

This guide provides business administration information for the following Windchill solutions:

Windchill PDMLink

Windchill ProjectLink

Pro/INTRALINK 9.0

Arbortext Content Manager

After a Windchill solution installation is complete, the following basic activities have been accomplished:

- A Web server and servlet engine are installed and configured.
- The Windchill database is installed and configured.
- Any administrative and enterprise directory services that are going to be used are configured.
- The Windchill solution is installed and has been started.

If you want more information about these activities, see the *Windchill Installation and Configuration Guide - Advanced*.

Before you can get started with administrative activities in your Windchill solution, you must log on as the administrator (defined during the installation). Additionally, you may also need to create a container for the installed default organization and establish additional administrators.

The next sections in this chapter describe how to log on, work with the initial organization context, and establish additional administrators. The last section provides a guide to which additional chapters you may want to read next.

## Regarding Arbortext Content Manager

Parts of this guide apply to Arbortext Content Manager. The general information and information specific to Windchill PDMLink (products and libraries) is relevant to Arbortext Content Manager administrators.

**Note:** As parts and CAD documents are not supported for Arbortext Content Manager, disregard references to those objects. References to "Arbortext" or "Arbortext products" refer to the Arbortext Editor and Arbortext Publishing Engine, not the Arbortext Content Manager solution.

For additional information on Arbortext Content Manager, see the *Arbortext Content Manager User's Guide*.

## Regarding Pro/INTRALINK 9.0

Parts of this guide apply to Pro/INTRALINK 9.0. The general information and information specific to Windchill PDMLink (products and libraries) is relevant to Pro/INTRALINK 9.0 administrators.

**Note:** As parts and dynamic documents are not supported for Pro/INTRALINK 9.0, disregard references to those objects.

For information about what is supported in Pro/INTRALINK 9.0, see *Getting Started with Pro/INTRALINK 9.0*.

## Logging On as the Administrator

You can access your Windchill solution using a URL from a Web browser.

**Note:** The server manager and method server must be running (as well as the Web server and servlet engine) before Windchill can be accessed.

The URL string is formatted as follows:

```
http://<hostname>:<port>/<webapp>
```

The required parameters were defined when Windchill Info\*Engine was installed. You only need to include the port number when the Web server being used is using a port number other than 80 (default). If the Web server is using the default port, then you can enter `http://<hostname>/<webapp>` in your Web browser **Address** (or **Location**) text box. For example, if you specified Windchill for the `<webapp>` parameter, then the following URL entered in the Web browser **Address** text box will open the Windchill home page:

```
http://<hostname>/Windchill
```

**Tip:** If you are logging on using the same system where your solution is installed, you can use **localhost** as `<hostname>`.

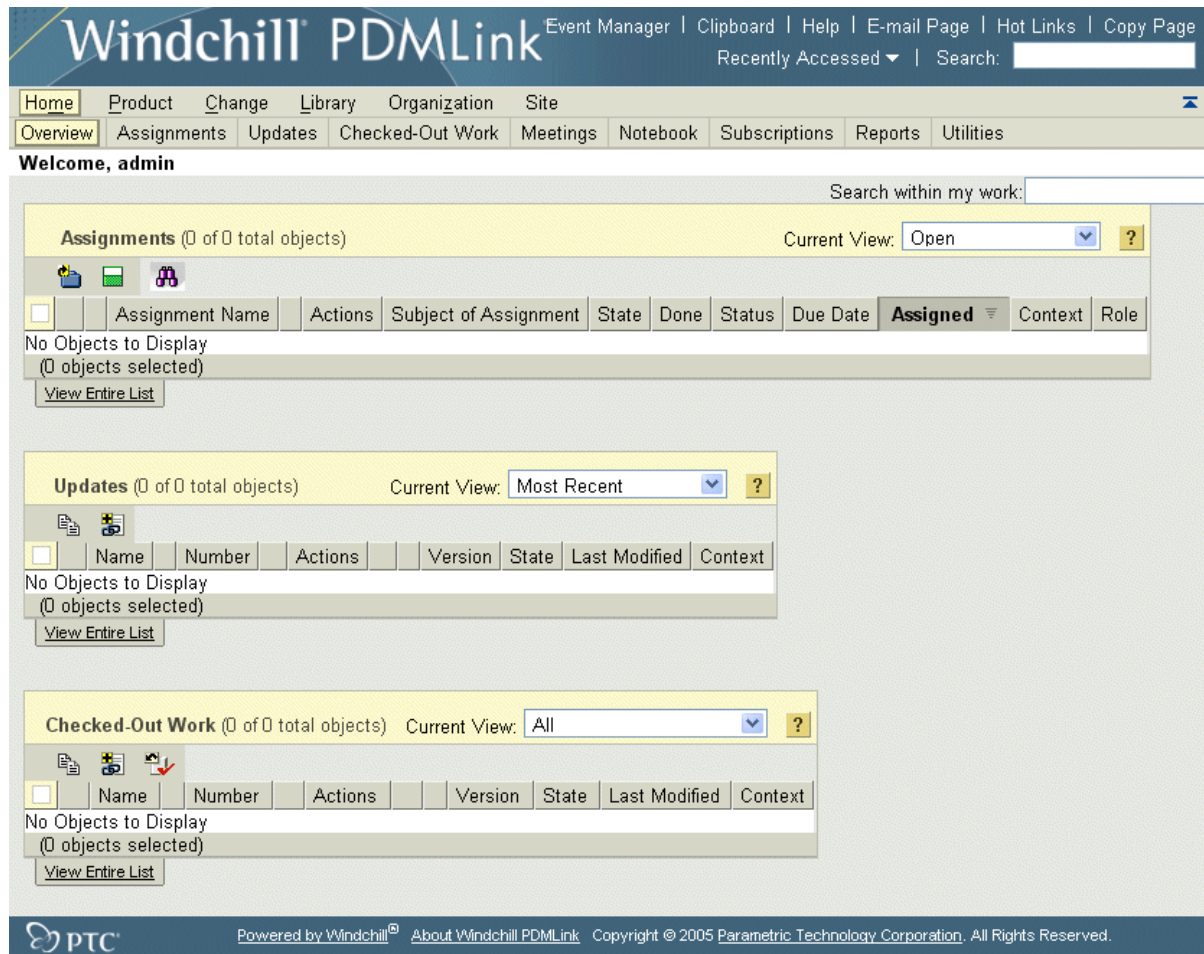
**Note:** If your system was configured with SSL, then the URL used to access Windchill will use the https prefix, rather than http.

Use the administrative user defined during the installation to log on. This user is a member of the Administrators group and has out-of-the box privileges that give you full control over all Windchill objects.

The home page that appears is unique to your Windchill solution. The following sections describe each home page.

## Windchill PDMLink Home Page

The Windchill PDMLink home page that appears is similar to the following:



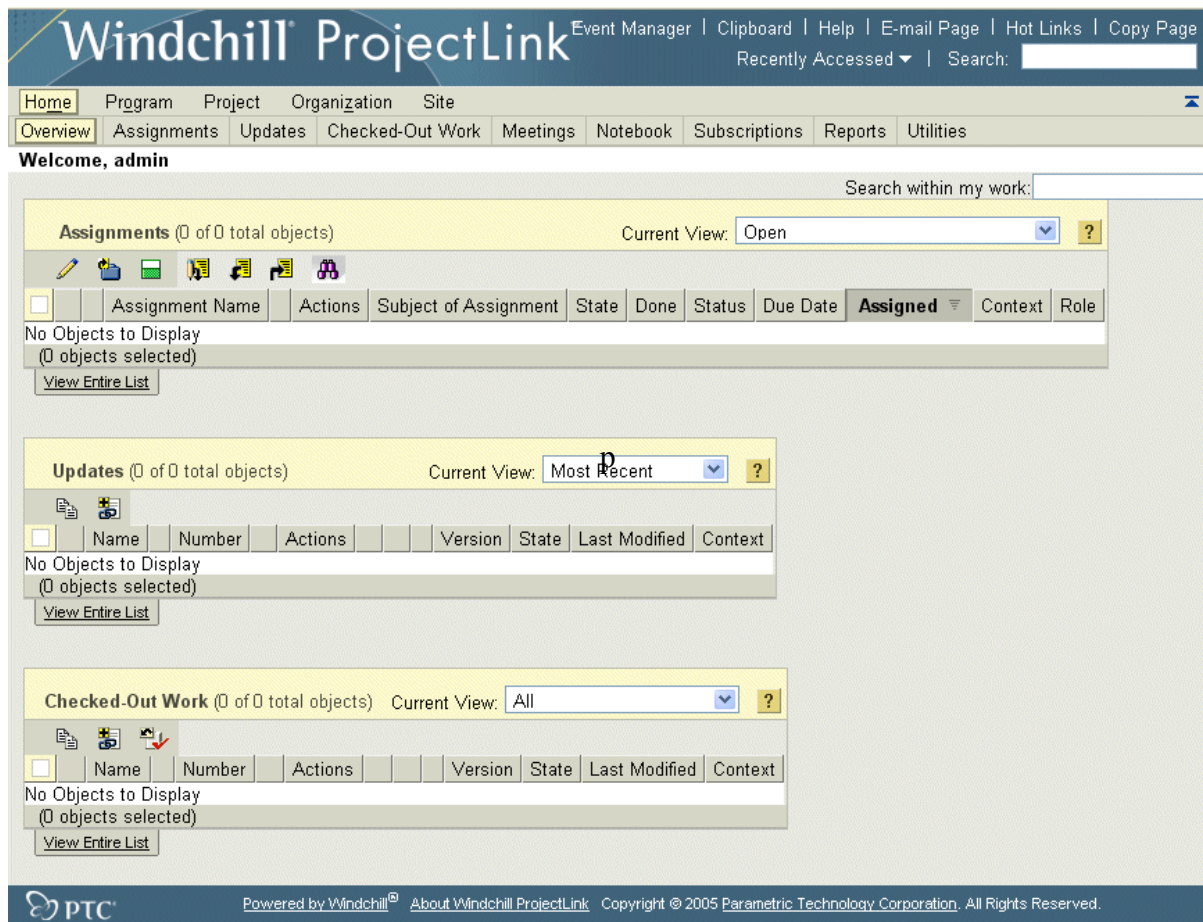
The tabs at the top of the window identify the major areas provided in Windchill PDMLink:

- From the **Home** tab, users manage their daily work.
- From the **Product** tab, users have access to all products to which they are a member. For each product, team members manage all of the information that is relevant to the design, manufacture, and support of a product. When only base data is installed, there are no out-of-the-box products.
- From the **Change** tab, users can view and work with change objects, such as problem reports, change requests, and change notices.

- From the **Library** tab, users have access to all libraries to which they are a member. In a library, team members can store and provide access to business information (such as in a document library) or can store and provide access to objects that are not related to a single product (such as in a common parts library). There are no out-of-the-box libraries.
- From the **Organization** tab, site and organization administrators manage organization contexts, including creating new organizations (site administrators only), managing groups and roles, access control, object initialization rules, and templates. This tab is only visible to site and organization administrators.
- From the **Site** tab, site administrators configure and manage the Windchill system as a whole. This tab is only visible to site administrators and is the tab from which the initial administration activities are done.

## Windchill ProjectLink Home Page

The Windchill ProjectLink home page that appears is similar to the following:



The tabs at the top of the window identify the major areas provided in Windchill ProjectLink:

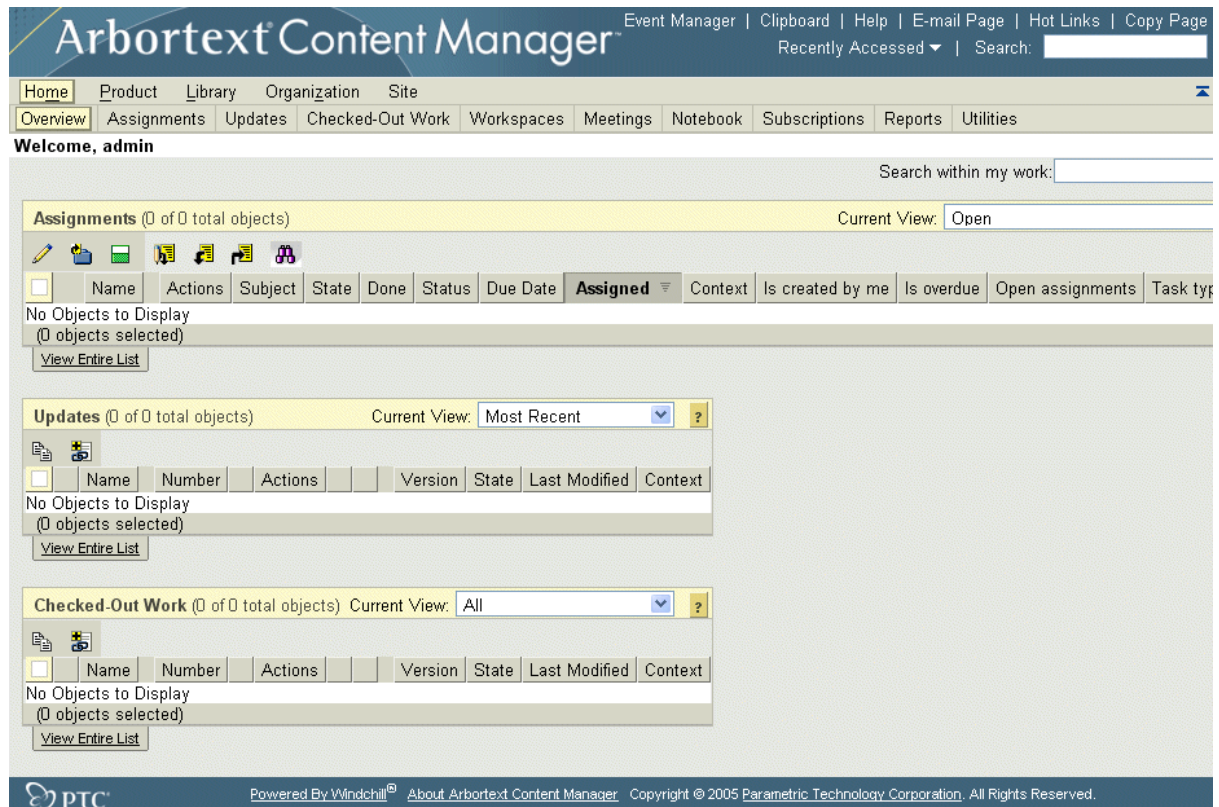
- From the **Home** tab, users manage their daily work.
- From the **Program** tab, users have access to all programs to which they are a member. A program is used to collect and manage the planning and execution of a set of projects. A program can be used solely as a means to organizing projects, or it can contain a plan, documents, and teams. There are no out-of-the-box programs.
- From the **Project** tab, users have access to all projects to which they are a member. For each project, team members have access to the project information, the project schedule, resources, and plan details. There are no out-of-the-box projects.



- From the **Organization** tab, site and organization administrators manage organization contexts, including creating new organizations (site administrators only), managing groups and roles, access control, object initialization rules, and templates. This tab is only visible to site and organization administrators.
- From the **Site** tab, site administrators configure and manage the Windchill system as a whole. This tab is only visible to site administrators and is the tab from which the initial administration activities, such as creating the organization container, are done.

## Arbortext Content Manager Home Page

The Arbortext Content Manager home page that appears is similar to the following:



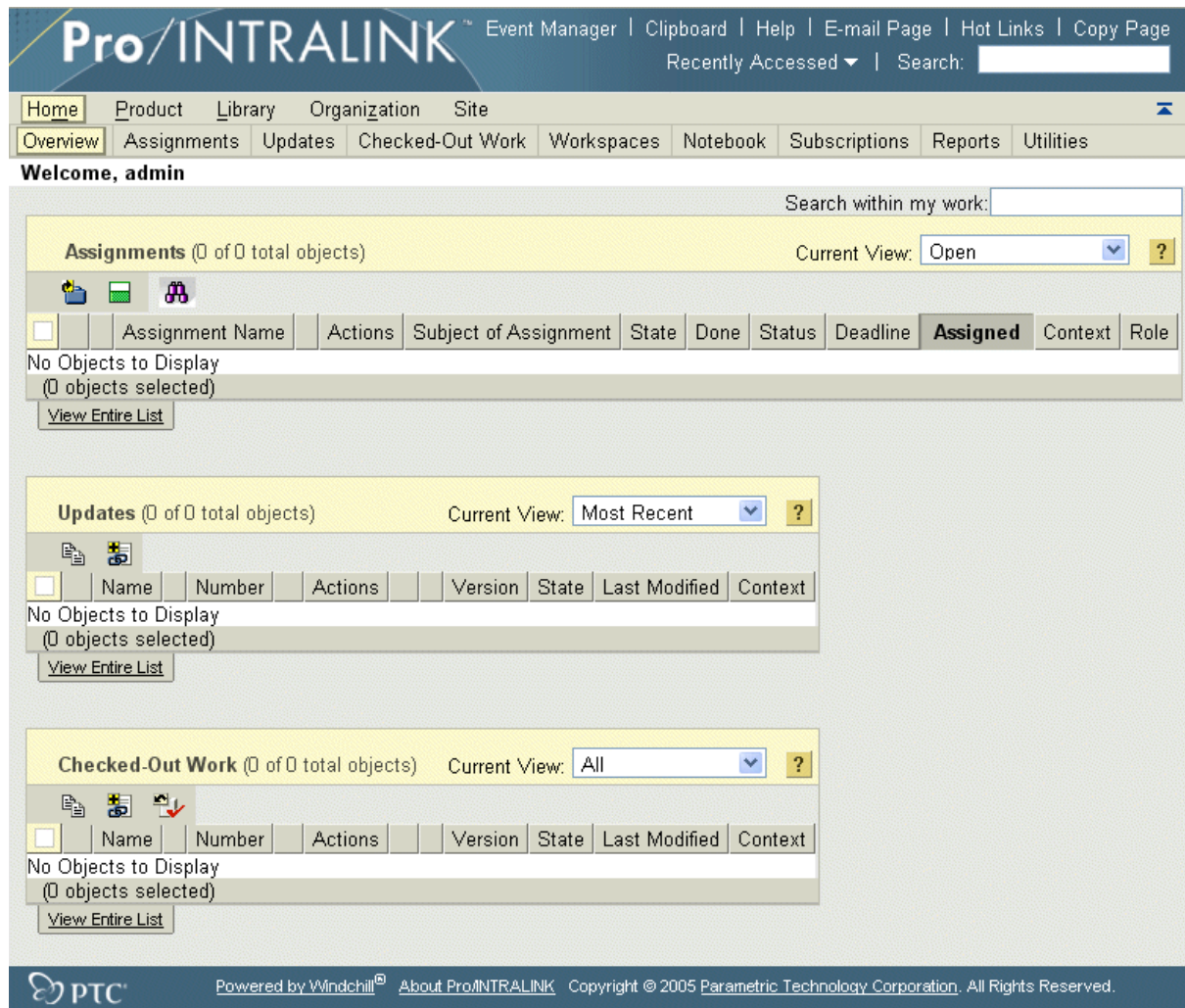
The tabs at the top of the window identify the major areas provided in Arbortext Content Manager:

- From the **Home** tab, users manage their daily work.
- From the **Product** tab, users have access to all products to which they are a member. For each product, team members manage all of the information that is relevant to the collaboration and support of a product. When only base data is installed, there are no out-of-the-box products.
- From the **Library** tab, users have access to all libraries to which they are a member. In a library, team members can store and provide access to business information (such as in a document library) or can store and provide access to objects that are not related to a single product (such as in a common parts library). There are no out-of-the-box libraries.
- From the **Organization** tab, site and organization administrators manage organization contexts, including creating new organizations (site administrators only), managing groups and roles, access control, object initialization rules, and templates. This tab is only visible to site and organization administrators.
- From the **Site** tab, site administrators configure and manage the Windchill system as a whole. This tab is only visible to site administrators and is the tab from which the initial administration activities are done.



## Pro/INTRALINK 9.0 Home Page

The Pro/INTRALINK 9.0 home page that appears is similar to the following:



The tabs at the top of the window identify the major areas provided in ProINTRALINK 9.0:

- From the **Home** tab, users manage their daily work.
- From the **Product** tab, users have access to all products to which they are a member. For each product, team members manage all of the information that is relevant to the design, manufacture, and support of a product. When only base data is installed, there are no out-of-the-box products.

- From the **Library** tab, users have access to all libraries to which they are a member. In a library, team members can store and provide access to business information (such as in a document library) or can store and provide access to objects that are not related to a single product (such as in a common parts library). There are no out-of-the-box libraries.
- From the **Organization** tab, site and organization administrators manage organization contexts, including creating new organizations (site administrators only), managing groups and roles, access control, object initialization rules, and templates. This tab is only visible to site and organization administrators.
- From the **Site** tab, site administrators configure and manage the Windchill system as a whole. This tab is only visible to site administrators and is the tab from which the initial administration activities are done.

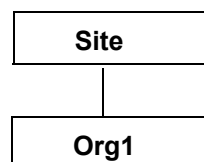
## Working With the Initial Organization Context

Your Windchill environment consists of a set of containers that hold all of the administrative areas (known as *domains*), rules, and data that make up the *context* from which Windchill users work. Although the underlying code refers to containers, the term **Context** is used throughout the user interface to identify where in the framework specific rules, domains, and data reside. Since **Context** is the label you see in the interface, this guide uses the term context (rather than container) so you can make the connection with the user interface. Unless you are working with the code, you can assume that context and container refer to the same thing.

The contexts are set up in a hierarchy so that the rules and data created in one context can be available to child contexts and the domains in one context can have child domains in a child context. The child domains inherit information from ancestor domains.

After installation, every Windchill solution has an installed top-level site context, and an initial organization context. Site context activities are performed from the **Site** tab; organization context activities are performed from the **Organization** tab.

The organization context is a child of the site context. The context name of the initial organization context is the organization name that was entered during installation. For example, assume that the organization name is Org1, then the following context hierarchy is established:



In this example, the Org1 organization context inherits rules and data from the site context.

Both the site and the initial organization contexts have the same owning organization *principal* (type WTOrganization). This becomes important if you want to segregate some of the objects created in an organization context from other contexts. The term principal is introduced here and used throughout Windchill as a way of categorizing Windchill objects that identify users, groups, and organizations. The organization principal created in the installation process becomes the owning organization for some of the objects created in both the site and the initial organization contexts. For example, soft types are associated with the owning organization. Therefore, the soft types created from within the initial organization context become site-level soft types because the owning organization is the same. For more information, see [Associating Soft Types with the Site and Organization Contexts](#) in the Type and Attribute Manager chapter. To have soft types created in an organization context that do not become available as site-level soft types, the context from which the soft types are created cannot be the initial organization context created during the installation.

**Note:** With Windchill PDMLink, the installer can install the golf cart demo data. As part of the golf cart installation, the Demo organization context is created and the Demo user is created as the organization administrator. As is the case for all organization contexts, the Demo organization context inherits rules and data from the site context, but the Demo organization does not have the same owning organization as the site context or the initial organization context.

You can create additional organization contexts. PTC recommends that if you are considering using multiple organization contexts in the future, you should create at least one additional organization context in which to create your products, libraries, projects, or programs, rather than creating them within the initial organization context. This allows you to add additional organizations (principals and contexts) without having to restructure your administrative data so that members of one organization cannot see data from another organization.

You must set the organization administrator for each organization context. How to do this is described in the next section.

For information about creating and using organization contexts, see the [Organizations](#) chapter.

## Establishing Administrators

Before you attempt to use your Windchill solution, you should establish a minimum set of administrators for the solution. The types of administrators that are available are determined by the Windchill solutions that are installed.

Every installed Windchill solution defines the Administrators group of which, by default, the administrative user entered during the installation (for example, wadmin) is initially the only member. Out of the box, the members of the Administrators group have full control over all Windchill objects and are commonly called the *system* or *site administrators*.

The additional types of administrators available are:

- An organization administrator manages a specific organization.
- A product manager manages a specific product (for Windchill PDMLink, Arbortext Content Manager, Pro/INTRALINK 9.0).
- A library manager manages a specific library (for Windchill PDMLink, Arbortext Content Manager, Pro/INTRALINK 9.0).
- A project manager manages a specific project (for Windchill ProjectLink).
- A program manager manages a specific program (for Windchill ProjectLink).
- A shared team manager manages a specific shared team and has management privileges for any context using the shared team.

In the out-of-the-box configuration, there are two LDAP directory service nodes. Administrative users reside in the Administrative LDAP node. For more information about the LDAP services, see the [Working with LDAP Directory Services](#) section in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

For e-mail messages to be sent, some mail servers require that a valid e-mail address be specified in the From header of the e-mail message. Windchill solutions use the wt.mail.from and wt.notify.notificationSenderEmail properties in the wt.properties file to populate the From header when sending e-mail messages. Setting these properties to a valid user should have been done as a manual step after installing your solution. If this was not done, see the *Windchill Installation and Configuration Guide - Advanced* for details on how to set these properties to the e-mail address of the Windchill administrator or some other authorized user.

The site administrator (as defined by the Administrators group) can use the links that are on the **Site** and **Organization** tabs. The site administrator is also known as the system administrator.

Clicking the **Administrators** link on the **Site** tab displays the **Site Administrators** table. From this table, you can add users to the Administrators group. The following sections provide additional information about the administrators in each Windchill solution and describe how to add the administrators.

## Organization Administrators

An organization administrator manages a specific organization. Initially, there is no organization administrator defined for an organization context.

Click the **Administrators** link on the **Organization** tab to add one or more users as organization administrators to the organization. For additional information, see [Creating Users to Select as Administrators](#).

## Windchill PDMLink Administrators

Additional Windchill PDMLink administrators include the following:

- A product manager for managing a specific product.

Initially, the product manager is the user who creates the product. Additional users can be added to the **Product Manager** role from the **Team** page of the product.

- A library manager for managing a specific library.

Initially, the library manager is the user who creates the library. Additional users can be added to the **Library Manager** role from the **Team** page of the library.

- A shared team manager for managing a specific shared team.

If a shared team is used in a product or library, then any users with the **Shared Team Manager** role act as product or library managers.

## Creating a Product or Library

To create a product or library, a user must be one of the following:

- The organization administrator.
- A member of the product creators group or the library creators group. These groups are maintained through the **Creators** link on the **Organization** tab. From the **Organization** tab, click the **Creators** link and then choose the type of creator from the **Current View** drop-down list on the **Creators** table.

## Windchill ProjectLink Administrators

Additional Windchill ProjectLink administrators include the following:

- A project manager for managing a specific project.

Initially, the project manager is the user who creates the project. Additional users can be added to the **Project Manager** role from the **Team** page of the project.

- A program manager for managing a specific program.

Initially, the program manager is the user who creates the program. Additional users can be added to the **Program Manager** role from the **Team** page of the program.

- A shared team manager for managing a specific shared team.

If a shared team is used in a project or program, then any users with the **Shared Team Manager** role act as project or program managers.

## Creating a Project or Program

To create a project or program, a user must be one of the following:

- The organization administrator.
- A member of the project creators group (to create a project).

Depending on the setup options selected when an organization is created, all members of the organization who have logged in may be automatically added to the project creators group. If this behavior is not enabled, both the site administrator and the organization administrator of the current organization can add users to the project creators group. From the **Organization** tab, click the **Creators** link, and select the **Project Creators** view from the **Creators** table **Current View** drop-down list. From this table you can add users.

- A member of the program creators group (to create a program).

Both the site administrator and the organization administrator of the current organization can add users to the program creators group. From the **Organization** tab, click the **Creators** link, and select the **Program Creators** view from the **Creators** table **Current View** drop-down list. From this table you can add users.

## Creating Users to Select as Administrators

**Note:** Only members of an organization can be organization administrators, product creators, library creators, project creators, or program creators.

In a production environment, users are usually defined in an enterprise directory that is set up during installation. If you are setting up a test system and do not have a set of users from which to select administrators, you can create a set of test users in the default directory service using the Principal Administrator. The Principal Administrator is available from the **Site** tab. Click the **Utilities** link and then click the **Principal Administrator** link to open the Principal Administrator.

The administrator user that is created during the installation (for example, wadmin) is not associated to a specific organization; this user does not have the organization attribute (usually “o”) set. Therefore, this user cannot be added as an organization administrator unless Administrator is updated to include the organization attribute. You can use the Principal Administrator to update the Administrator user.

**Note:** If your site does not use the organization attribute in the directory service entry, users can be assigned to an organization using the usersOrganizationName property. For more information on using this property, see adding enterprise directories in the *Windchill Installation and Configuration Guide - Advanced*.

At a minimum, you need to either update the Administrator user so the user is a member of the default organization or create another user who can be the organization administrator. This user can then administer the organization and create products, libraries, projects, and programs.

## Establishing End Users

Depending upon how your Windchill solution was set up during installation, *end users* (users that do not have administrative privileges) are managed in one of two ways:

- [Using an Enterprise Directory Service](#)
- [Using the Principal Administrator Utility](#)

Windchill uses both the Windchill database and a directory service when creating users. For each user, there is an entry in a directory service and a Windchill object stored in the database.

In the out-of-the-box configuration, there are two LDAP directory services: the Administrative LDAP service and the Enterprise LDAP service. End users can reside in either service; although typically, the Administrative LDAP service should be used for administrative users and enterprise LDAP service should be used for all end users who are not administrators. For more information about the LDAP services, see the [Working with LDAP Directory Services](#) section in the Principals (Users, Groups, and Organizations) chapter.

It is not required that end users be associated with an organization; however, categorizing users by organization provides an additional way in which you can identify a set of users. Windchill manages Windchill objects within each organization context and users affiliated with an organization can be given access to objects created from within an organization context.

**Note:** If your site does not use the organization attribute in the directory service entry, users can be assigned to an organization using the `usersOrganizationName` property. For more information on using this property, see adding enterprise directories in the *Windchill Installation and Configuration Guide - Advanced*.

## Using an Enterprise Directory Service

Instead of using the Aphelion Directory Service that is bundled with your Windchill solution for the end users in the Enterprise LDAP service, you can use an enterprise directory service such as the Microsoft Active Directory Service (ADS). Each enterprise directory service entry contains attributes for the user. For example, user entries can have attributes for the user's full name, e-mail address, and organization.

The installation process can set up the enterprise directory service as the Enterprise LDAP node. If this was not done, an additional directory service can be established that uses the enterprise directory service. For information about setting up enterprise directory services, see the *Windchill Installation and Configuration Guide - Advanced*.

Usually, users in an enterprise directory service are managed through a third party tool and the Windchill user database object is created automatically the first time the user logs into the Windchill solution.

## Using the Principal Administrator Utility

The Principal Administrator utility can be used to manage end users. It is part of the Windchill solution and can be accessed on the **Site** and **Organization** tabs. Both the LDAP entry and the Windchill user database object are created when a user is created from the Principal Administrator.

End users can be added to the Windchill solution by either a site or an organization administrator using the Principal Administrator. To create a new user, you must have write permission to the directory service in which you want the user directory entry to reside.

When you create a new user from the **Site** tab, you can associate that user with any existing organization within the Windchill solution.

When adding a new user in the Principal Administrator, you can associate that user with a group and a profile. Groups can be created from either the Principal Administrator or from the **Groups** link within the active organization context.



The **Profile** link available from the organization and site contexts allows you to set up the visibility of actions in the user interface for users, user-defined groups, and organizations.

## The Next Steps

After your administrators are established, individual administrators can log on and perform their administrative duties. For example, they can create products, libraries, projects, programs, and end users.

To understand what those duties might entail, refer to the following table:

Type of Administrator	Recommended Chapters to Read
All administrators	<a href="#">Administration Overview</a> - for general information. <a href="#">Contexts</a> - for major concepts about contexts.
Site administrator	<a href="#">Site</a> - for details using on the <b>Site</b> tab. <a href="#">Organizations</a> - for details on using the Organization tab and creating organization contexts. <a href="#">Principals (Users, Groups, and Organizations)</a> - for details about creating and managing users, groups, and organizations.
Organization administrator	<a href="#">Organizations</a> - for details on using the <b>Organization</b> tab and creating organization contexts. <a href="#">Principals (Users, Groups, and Organizations)</a> - for details about creating and managing users and groups within your organization.
Product manager	<a href="#">Products and Libraries</a> - for details on administering products and using the <b>Product</b> tab.
Library manager	<a href="#">Products and Libraries</a> - for details on administering libraries and using the <b>Library</b> tab.
Project manager	<a href="#">Projects and Programs</a> - for details on administering projects and using the <b>Project</b> tab.
Program manager	<a href="#">Projects and Programs</a> - for details on administering programs and using the <b>Program</b> tab.
Shared team manager	<a href="#">Teams</a> - for details on administering context teams and shared teams. As well as the chapter recommended for the context manager of any context the shared team is used in.



# 2

## Administration Overview

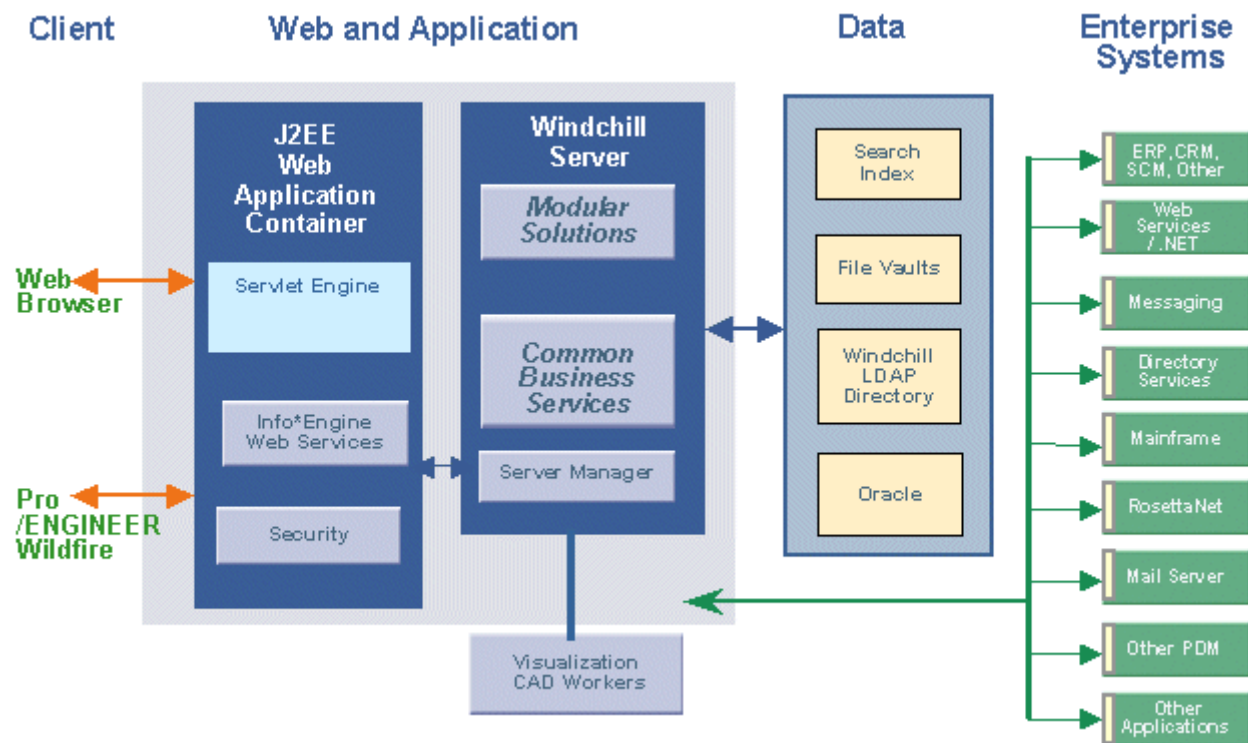
This chapter provides a general overview of your installed Windchill runtime architecture and Windchill environment. It also introduces you to the main Windchill administration areas and gives some basic information about how to manage your Windchill solution. Later chapters assume that you have read the information in this chapter.

Topic	Page
Your Installed Windchill Runtime Architecture .....	2-2
Managing Your Windchill System.....	2-3
Your Installed Windchill Environment .....	2-3
Managing User Access to Data .....	2-5
Managing Users.....	2-16
Managing Data .....	2-18
Managing Windchill Processes .....	2-23
Managing User Collaboration .....	2-23
Additional Administrative Groups .....	2-24
Post-Installation Activities .....	2-25
About the windchill Command .....	2-25
About the windchill shell .....	2-27
About the xconfmanager Utility.....	2-28

## Your Installed Windchill Runtime Architecture

After a base Windchill solution is installed, the Windchill runtime architecture consists of the following:

- Client applications that allow users access to Windchill. The clients can include the Windchill client pages, the visualization clients, the Pro/ENGINEER Wildfire client, and, possibly, workgroup manager clients.
- A Web server that includes a servlet engine, Windchill Info\*Engine Web services, and security modules.
- The Windchill server that includes the Windchill solutions and common business services.
- Data storage that includes the Aphelion LDAP directory service and an Oracle database.
- Possibly, connections to other enterprise systems, such as an enterprise directory service, ERP, CRM, SCM, or other PDM systems.



For details about what is installed, see the *Windchill Installation and Configuration Guide - Advanced*.

## Managing Your Windchill System

The *Windchill System Administrator's Guide* describes how to perform system operations that change and improve the out-of-the-box system that is installed. The following topics are covered in the *Windchill System Administrator's Guide*:

- Runtime Services, including starting and stopping Windchill, using the xconfmanager utility, setting up Meeting Center, Windchill Desktop Integration, and the Windchill Scheduler.
- The Bootstrap Client
- External File Vaults
- Content Replication
- Windchill Import and Export
- Background Queues
- Customizing Online Help
- Customizing Online Tutorials

The *Using Pro/ENGINEER Wildfire with Windchill* describes how to change and improve the out-of-the-box system that is installed when your site has Pro/ENGINEER Wildfire clients that are using the system.

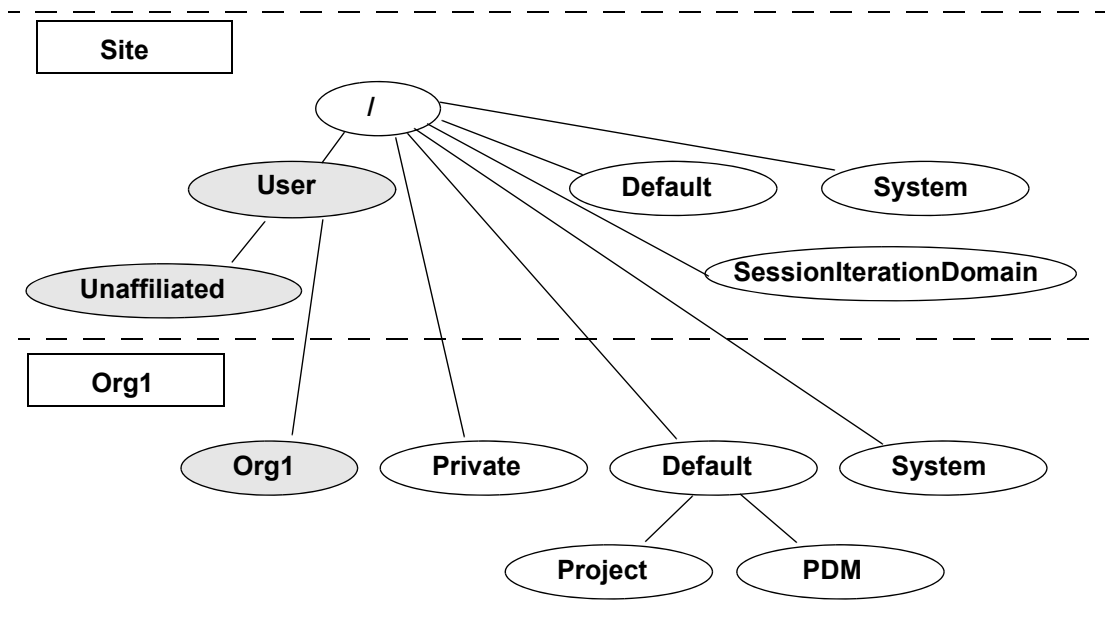
The *Windchill Advanced Deployment Guide* describes how to deploy a complex Windchill environment, including firewalls, multiple method servers, and clustered servers.

## Your Installed Windchill Environment

Your installed Windchill environment consists of a set of containers that hold all of the administrative areas (known as domains), rules, and data that make up the context from which Windchill users work.

The containers are set up in a hierarchy that is described in [Working With the Initial Organization Context](#) in the Getting Started chapter.

Within the installed containers, a set of domains are loaded during the installation process. For example, the site and Org1 contexts have the domains shown in following diagram:



In the diagram, the dashed line shows the container boundaries and the domain inheritance is shown by the lines connecting the domains. The top-level domain is labeled / (root) and is in the site context. The shaded domains are the domains associated with Windchill principals (users, user-defined groups, and organizations).

After containers are created and users become team members, the framework established is called the *context* from which the users work. In many instances, the context includes the contents of a specific container and the domains, rules, and data available from ancestor containers. For example, if a user entering Windchill ProjectLink navigates to a folder within the Bike Design project and creates a new document, that document is managed in the context of the Bike Design project. Persons with access to the Bike Design project may automatically have the right to see and modify the new document.

Depending on how container rules are set up, users may also be able to share data across containers. When this is the case, the user context can include data from multiple containers. You can think of the context as providing the framework from which user actions are executed. This framework is defined by a container, but can include data from multiple containers. For example, parts defined in one container can be used in an assembly structure that is saved in a different container.

Throughout the user interface, **Context** is used to identify where in the framework specific rules, domains, and data reside. Since **Context** is the label you see in the interface, the remainder of this guide uses the term context (rather than container) when referring to the framework that is in place. Each context provides the following:

- The context structure, which includes the default domains and folders, forum topics, reference notebook folders, and user notebook folders (if used).
- Context participation, which includes the available roles, teams, and system groups.
- Default access policies.
- Data types and object initialization rules.
- Preferences established for the specific context.
- Default life cycle, workflow, context, team, and report templates.

The base data that is loaded during the installation process creates the out-of-the-box templates for contexts, workflows, life cycles, teams, and reports, and associates them with the System domain that is in the site context. These templates are then available to descendent contexts where appropriate.

The Administrator user and the Administrators group are created during the base installation and are also associated with the System domain. Out of the box, these administrators have full access control over all Windchill objects.

One important type of data that is loaded is the context template data. Context template data files are XML files that define what is initially in a context when it is created. The file contains the types of items that are similar to the type of data, rules, and domains that are loaded during the Windchill solution installation. When creating additional contexts, the administrator selects the context template data file to use to establish the context. For more information on context templates, see [Creating the Contexts from which Users Work](#) in the Contexts chapter.

## Managing User Access to Data

All Windchill installations establish an initial organization principal and organization context for the initial organization (as described in [Working With the Initial Organization Context](#)). To belong to the initial organization that is established, each user must have an entry in the user directory service that was set up as part of the installation process and the organization attribute of the entry (usually “o”) must be set to the organization name. By being a member of the initial organization, users can be given access to the data stored in the organization context or its child contexts (depending on the access control rules that are in place). Users who are not members of the initial organization usually do not have access to the data stored in the initial organization context or its child contexts (unless they are invited to participate by someone in the organization). Again, this is dependent on the access control rules that are in place.

**Note:** If your site does not use the organization attribute in the directory service entries for users, all users can be assigned to an organization using the `usersOrganizationName` property. For more information on using this property, see the *Windchill Installation and Configuration Guide - Advanced*.

After analyzing the users who need access to data, the site administrator determines whether additional organization contexts are needed. The organization context can limit data access to members of an organization. For example, if your Windchill solution will be used by multiple companies where each company has a different set of data and rules that will be used from within the solution, then setting up an organization context for each company would be the best approach. However, if only one company will actively use the solution and other companies will just provide data that is managed by the initial organization, then one organization context is sufficient.

After determining the organization structure that is needed, the site administrator or organization administrator should determine if there are one or more sets of users who will need to have access to data in multiple application contexts that are created under one organization context. If there are sets of users who will be accessing multiple application contexts under one organization, shared teams can be created at the organization context level. A *shared team* identifies a set of users who will be working together in multiple application contexts and therefore, can be assigned together as a unit in the organization context rather than added individually to each application context.

Members of the appropriate application context creators groups can then create product, library, project, or program application contexts under an organization, depending on the contexts available from your Windchill solution. Using application contexts further separates the access of data. In each context, unique policy rules are set.

For example:

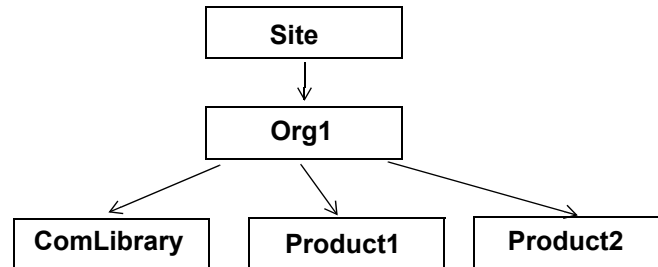
- The permissions for parts stored in a product context are usually set so only members of the product have write access to the parts.
- The permissions for documents stored in a library context are usually set so only members of the library have write access to the documents.
- The permissions for all objects stored in a project context are usually set so only members of the project have write access to the objects.
- The permissions for all objects stored in a program context are usually set so only members of the program have write access to the objects.

To use a shared team in an application context, it must be created before the application context is created.



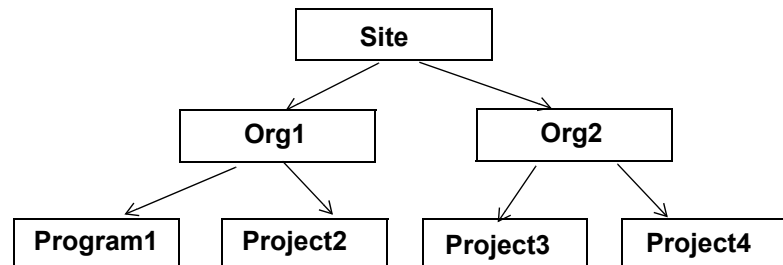
## Product and Library Hierarchy

The following diagram shows the context hierarchy of products and libraries where administrators have created one library context and two product contexts under the Org1 context:



## Program and Project Hierarchy

The following diagram shows the context hierarchy of programs and projects where there are two organization contexts (Org1 and Org2) and administrators have created a program and project under the Org1 context and two project contexts under Org2 context:

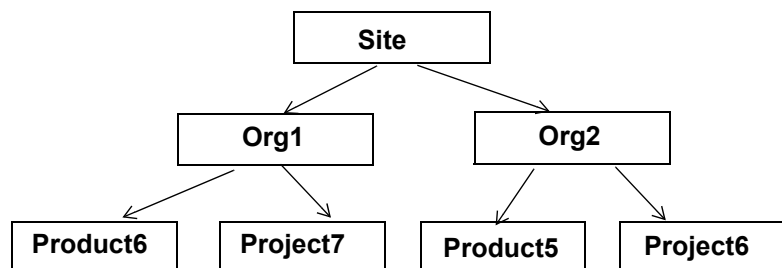


## Hierarchy for Integral Windchill Solutions

The following solutions can be installed in the same codebase:

- Windchill PDMLink and Windchill ProjectLink
- Arbortext Content Manager and Windchill ProjectLink

The following diagram shows the integral hierarchy where there are two organizations (Org1 and Org2) and administrators have created a product and project context under each organization:



## Managing Access to Data through Access Control Rules

In each context, a set of access control rules can be set when the context is created. These rules can be defined programmatically as well as defined in the template that is used to create the context. Additionally, an administrator can define access control rules for the data that is in the context and in its child contexts, thereby further controlling the access to data. Generally, each access control policy rule does the following:

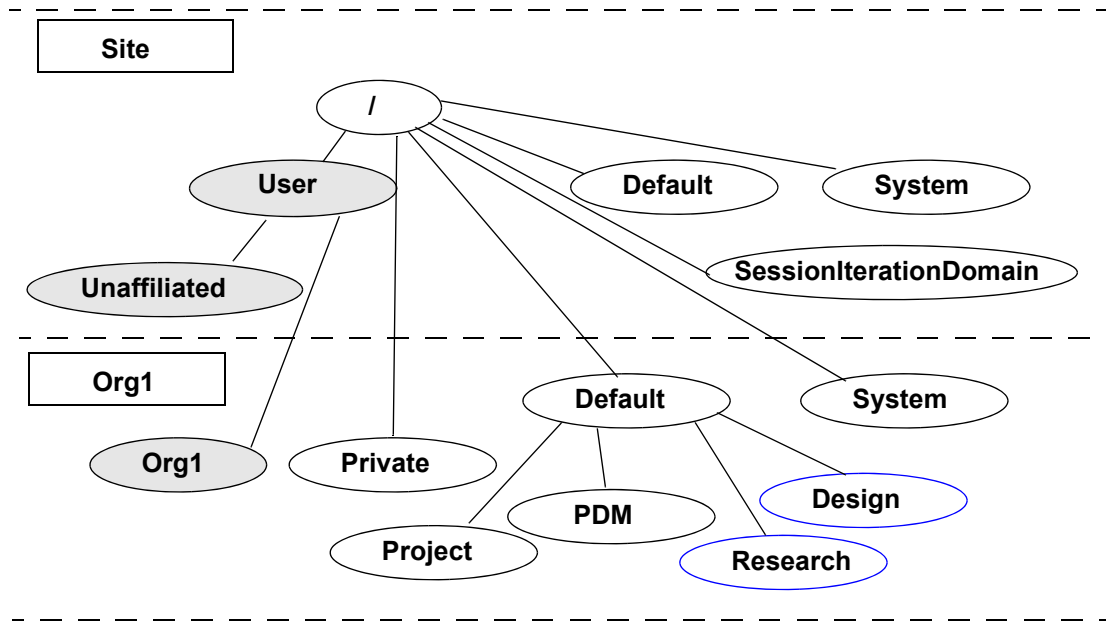
- Identifies a type of data stored in a specific administrative area (domain) to which access permissions are applied.
- Identifies the specific state (or all states) of an object to which access permissions are applied.
- Identifies users (either as individual users, user-defined groups of users, or entire organizations) for whom access is either granted or denied.
- Specifies the access permissions (such as read, write, and modify) given to the users for the data type in the specified domain.

Access control policy rules have a hierarchy based on the domain hierarchy. Dependant domains inherit rules from ancestor domains and rules granting permissions cannot override those rules that are inherited.

The following sections introduce the domains and context structure that are available.

## Shared Teams

All shared teams are created in an organization context and, as part of the creation process, domains with the same name as the shared teams are created as children of the Default domain in the organization context. For example, if the Research and Design shared teams are created in the Org1 context, then the additional domains created are shown in following diagram:



## Product, Library, Project, and Program Contexts

For each product, library, project, and program that is created, two domains are created in the context:

- A System domain.

This domain is a child of the Private domain that is in the organization context. Therefore, the System domain inherits from the /Private domain.

- A Default domain.

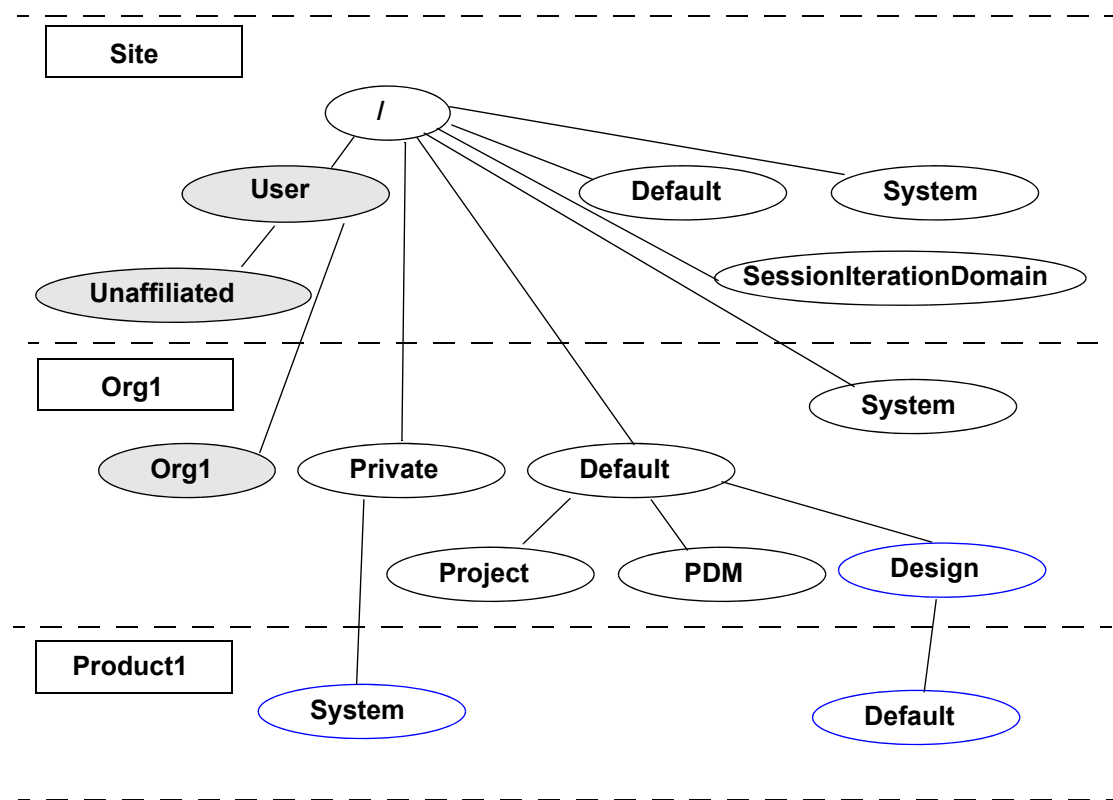
The placement of the Default domain is dependant on the following:

- Whether a shared team is used.
- If a shared team is not used, then whether **Private Access** is selected.

The following sections describe the possible placement of the Default domain and provide a diagram for each scenario.

## Contexts using Share Teams

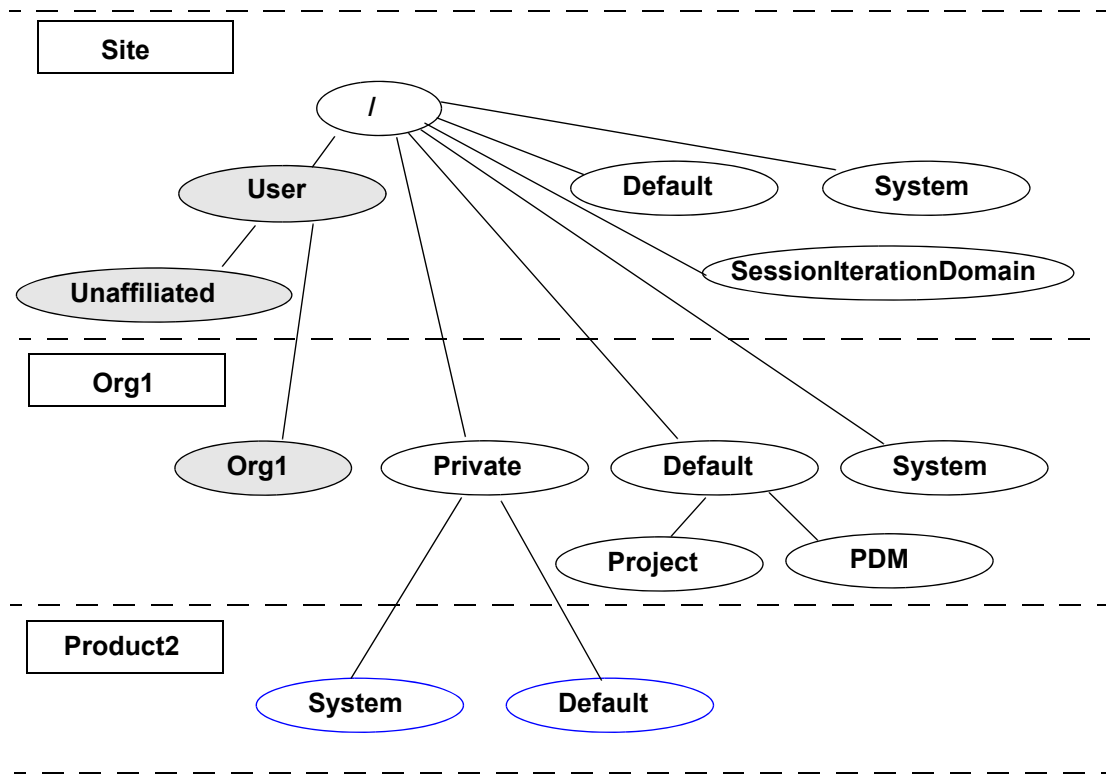
**Note:** When the creation of a product, library, project, or program specifies that a shared team is used, the Default domain that is created inherits from the shared team domain that is in the organization context, as shown in the following diagram. (Depending on the template you use, other domains can also be created in the context.) The diagram shows the Site, Org1, and Product1 contexts, the Design shared team domain in the Org1 context, and the System and Default domains in the Product1 context:



In this scenario, the domain hierarchy for the System and Default domains in libraries, programs, and projects is the same as the hierarchy for products. If multiple contexts use the same shared team, their Default domains inherit from the same shared team domain. In this case, they inherit from the /Default/Design domain.

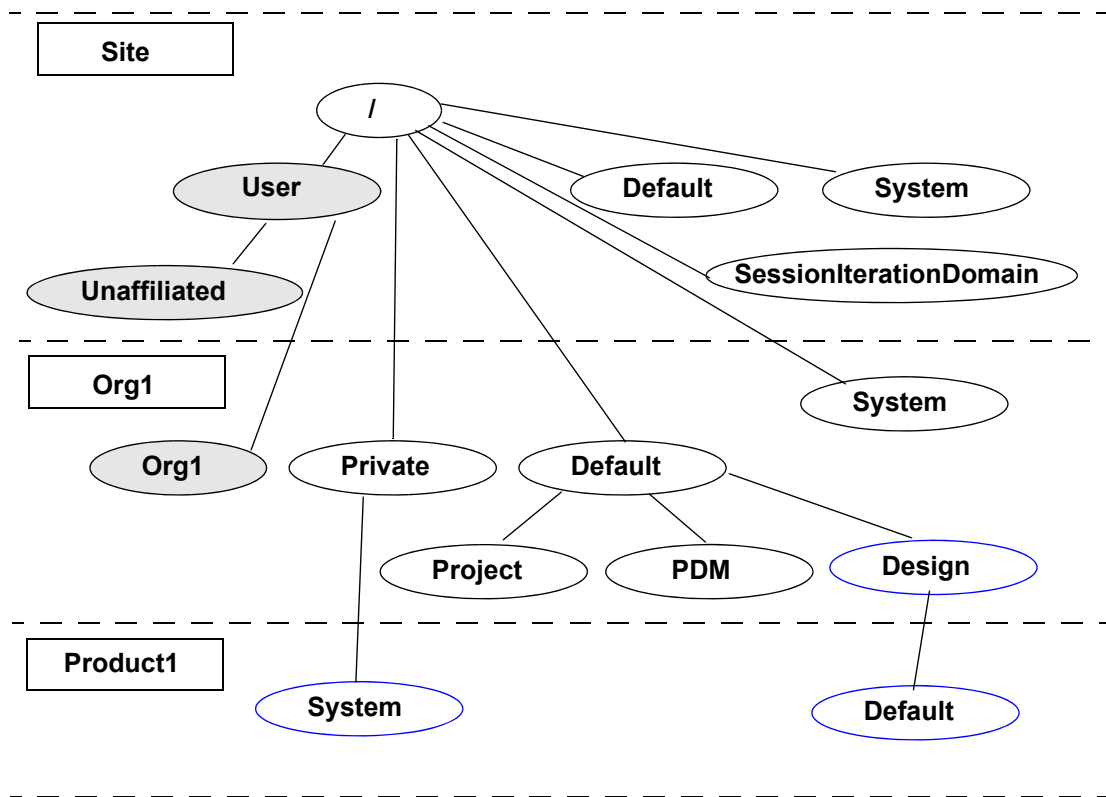
## Contexts with Private Access

When the creation of a product, library, program, or project does not specify a shared team but does specify **Private Access** (for product and libraries) or the Members Only **Access Group** option (for programs and projects), the Default domain that is created inherits from the /Private domain in the organization context instead of from either the /Default/PDM domain or the /Default/Project domain. The resulting hierarchy for a private Product2 context is as follows:



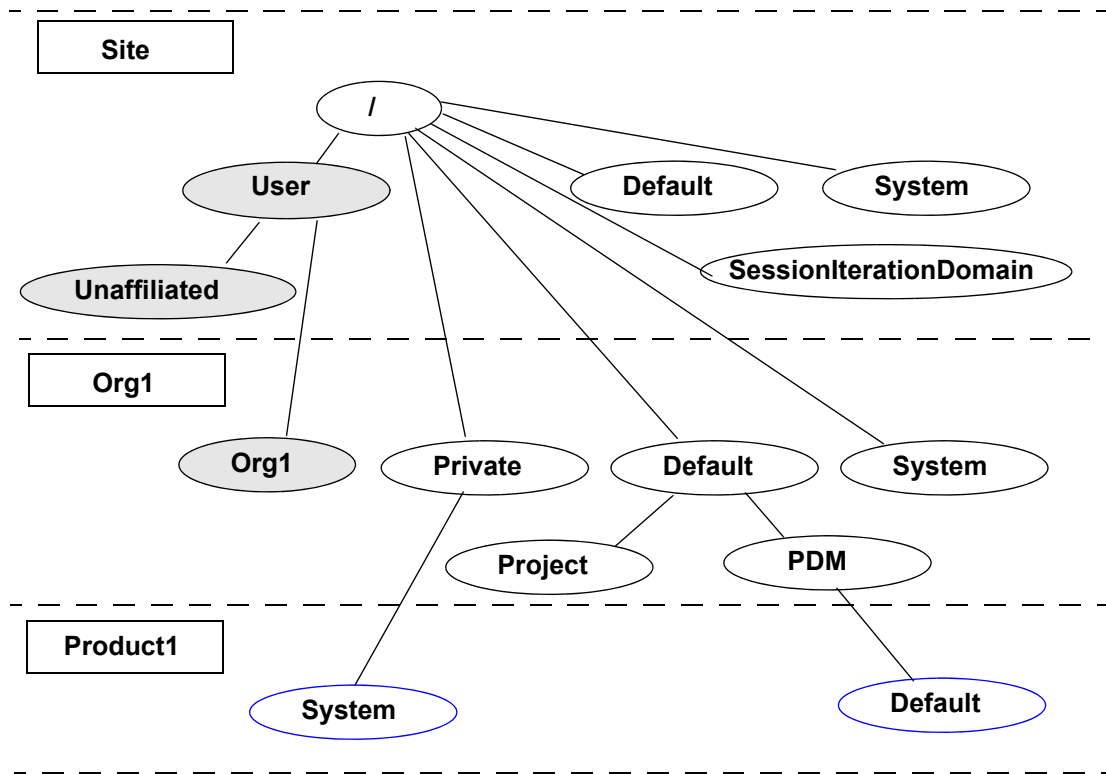
In this scenario, the domain hierarchy for the System and Default domains in libraries, programs, and projects is the same as the hierarchy for products.

**Note:** Specifying a shared team overrides the use of the /Private domain hierarchy. When a shared team is used, the Default domain always inherits from the shared team domain, as shown below:



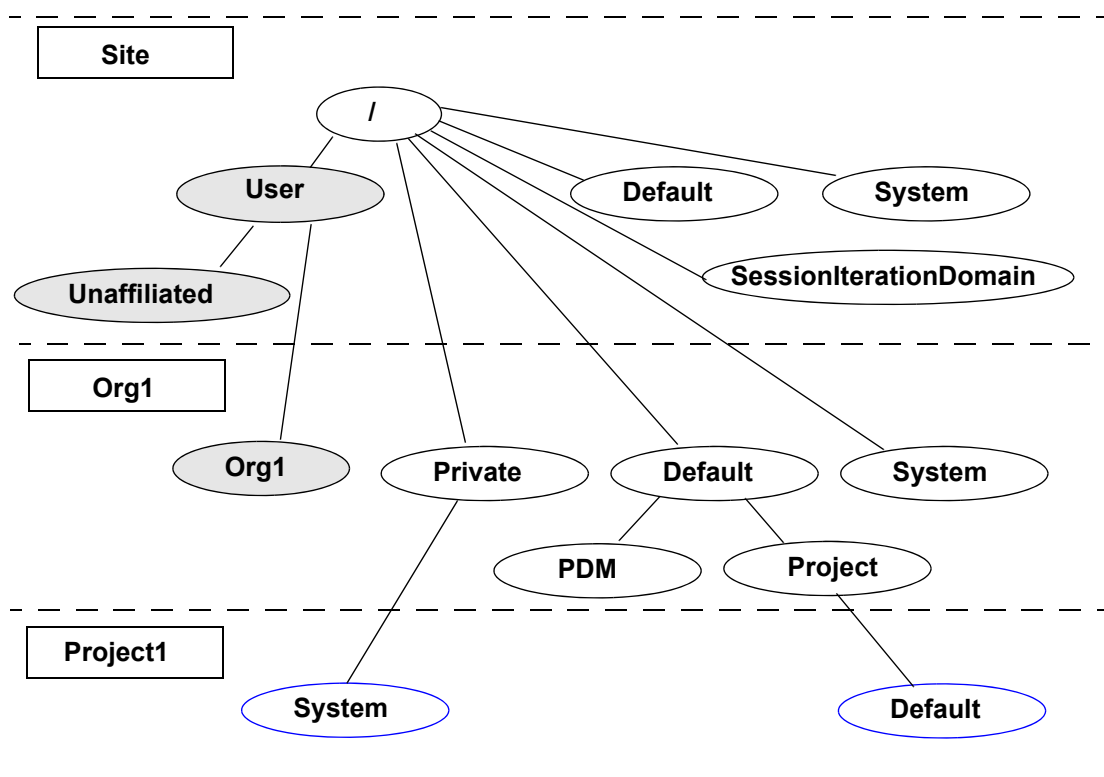
## Products and Libraries without Private Access

When the creation of a product or library does not specify a shared team and does not specify **Private Access**, the Default domain that is created inherits from the /Default/PDM domain in the organization context, as shown in the following diagram. (Depending on the template you use, other domains can also be created in the product or library context.) The diagram shows the Site, Org1, and Product1 contexts and domains:



## Programs and Projects without Private Access

When the creation of a program or project does not specify a shared team and does not specify the Default **Access Group** option, the Default domain that is created inherits from the /Default/Project domain in the organization context and a System domain inherits from the /Private domain in the organization context, as shown in the following diagram. (Depending on the template you use, other domains can also be created in the organization and project contexts.) The diagram shows the Site, Org1, and Project1 contexts and domains:



## Setting Up User Access to Data

You can determine which sets of users have access to the data in the context by setting access control rules in the domains associated with the context where the data resides. You can also establish the set of access control permissions users can view and edit when they manage the security of individual objects from within an application context.

Use the Policy Administrator to create policy rules. For details on domain inheritance and setting policy rules, see [Administering Domains and Policies](#) in the Contexts chapter.

Use the Preference Manager to establish the set of permissions users can view and edit. For details, see [Creating and Managing Access Control Rules](#) in the Access Control chapter.



The **Team** link that is available in each context allows you to set up the role and role memberships; these can be used as the system groups against which the access control rules are set, as described in [Managing Access to Data through Team Memberships](#).

Use the Principal Administrator to update users who have changed in your user directory service or create and update user-defined groups at the organization level that can be used as team members. For additional information about managing users, user-defined groups, and organizations, see the [Principals \(Users, Groups, and Organizations\)](#) chapter.

Additionally, you can limit the visibility of the actions in the user interface for users, user-defined groups, and organizations. For more information, see [Profile Management](#) in the Principals (Users, Groups, and Organizations) chapter.

## Managing Access to Data through Team Memberships

Another aspect of managing user access to data can be found in managing who becomes a member of an application context. The context team associated with a product, library, project, or program establishes which users are members of a specific application context. A context team can be made up of the following teams:

- A shared team that is established by the site or organization administrator (or by others given the rights to create shared teams) in the context of an organization and selected when creating the application context.
- A local team that is established by the application context manager as part of creating and managing the application context.
- Both a shared team and a local team.

Team members are added to a shared team according to their established role in multiple application contexts. For example, you could have a group of engineers who fill the Design Engineer role for many of the products or projects managed by your organization. This user-defined group could then be added to a shared team that is then selected when the application context is created.

Team members are added to a local team according to their role in a specific product, library, program, or project. For example, you could have an individual who will be in the Reviewer role for only a specific product or project. Then, this individual could be added to the local team for that product or project. The initial roles that are available for shared and local teams are determined when the teams are created; however, additional roles can be added.

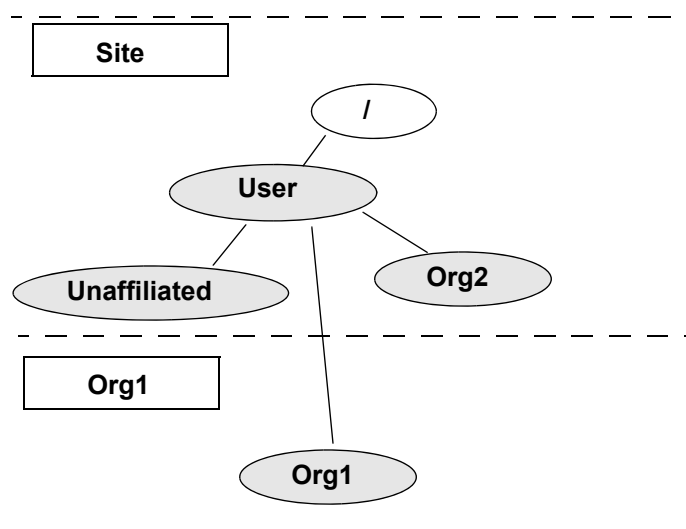
For each role used in a context team there is a corresponding system group created that administrators can use to create access control rules for the members assigned to the role. The **Team** link from the **Product**, **Library**, **Program**, or **Project** tab in your Windchill solution provides access to the interface for managing local teams. The **Team** link from the **Organization** tab in your Windchill solution provides access to the interface for managing shared teams. Use the Policy Administrator to create access control rules.

Your Windchill solution also uses roles and corresponding users defined in life cycle templates and team templates (if they are defined for an object). For additional information about teams, see the [Teams](#) chapter.

For additional information about access control rules, see [Administering Domains and Policies](#) in the Contexts chapter.

## Managing Users

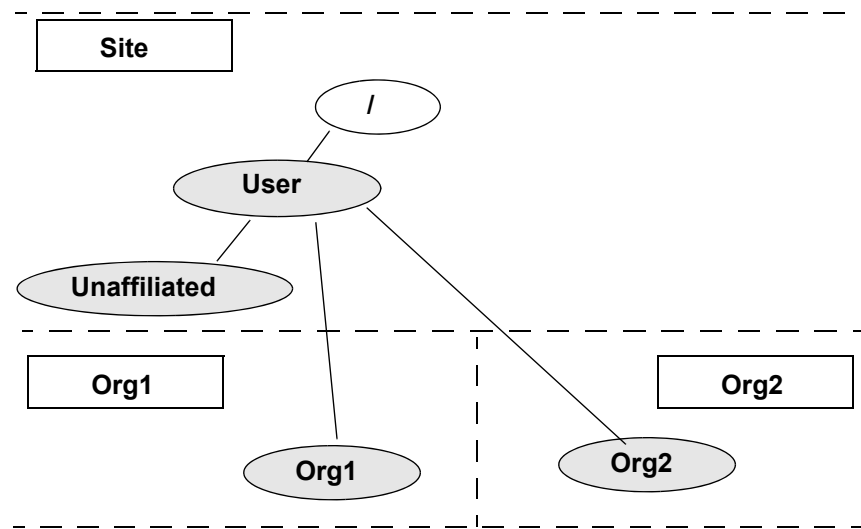
As mentioned in earlier sections, the default domains associated with users are the User domain in the Site context or one of its child domains. For example, assume that the Org1 organization context has been created and users from both the Org1 and Org2 organizations and users that have no organization affiliation (the organization attribute is not set for the user) have accessed your Windchill solution. The following domains are then automatically created:



In this example the user and domain associations are as follows:

- Users from the Org1 organization (their organization attribute is set to Org1) are associated with the Org1 domain by default. This domain is in the Org1 context.
- Users from the Org2 organization (their organization attribute is set to Org2) are associated with the Org2 domain by default. This domain is in the Site context.
- Users who have no organization affiliation are by default associated with the Unaffiliated domain. This domain is in the Site context.

In the previous example, assume that the Org2 organization context is now created in your Windchill solution. Then the Org2 domain moves from the Site context to the Org2 context, as show in the following diagram:



By using the default domains for users, users are automatically grouped by organization and access policy rules for each organization are initially set using the organization context template used to create the organization. Rules for users not affiliated with any organization can be set using the Unaffiliated domain.

When your Windchill solution creates user objects, a personal cabinet is also created for each user. By default, the personal cabinet for a user is put in the same domain as the user. Using this approach allows the access control rules for personal cabinets to be in the same domain as the access control rules for the users.

In the previous examples, the organizations have short names. If the organization names you are using are longer than 193 characters, then the names are truncated when creating corresponding domain names. For more information, see [Creating Domains](#) in the next chapter.

## Managing Data

The Windchill installation establishes a set of business object types that are available in the site context and then can be inherited by organization contexts and then by application contexts. The out-of-the-box business object types are described in the following table:

Object Type	Display Name	Description
wt.doc.WTDocument	Document	General documents such as text files or Microsoft Word documents
wt.change2.WTAnalysisActivity	Analysis Activity	Change data that assigns an analysis task to a user
wt.change2.WTChangeActivity2	Change Task	Change data that assigns product development work to a user
wt.change2.WTChangeInvestigation	Change Investigation	Change data that collects information about the root cause of a product problem
wt.change2.WTChangeIssue	Problem Report	Change data that reports a potential product problem
wt.change2.WTChangeOrder2	Change Notice	Change data that collects all tasks required to implement a product change
wt.change2.WTChangeProposal	Change Proposal	Change data that proposes a solution to a product problem
wt.change2.WTChangeRequest2	Change Request	Change data that collects all change impact data required for a decision

Object Type	Display Name	Description
wt.epm.EPMDocument	EPM Document	<p>This type is the parent type of soft types for the following business objects:</p> <ul style="list-style-type: none"> <li>workgroup manager CAD documents (type: &lt;prefix&gt;.DefaultEPMDocument)</li> <li>dynamic documents (type: &lt;prefix&gt;.DynamicDocument)</li> <li>burst configuration files (type: &lt;prefix&gt;.BurstConfiguration)</li> <li>notes (type: &lt;prefix&gt;.Note)</li> </ul> <p>For the details on the format of soft types, see <a href="#">Creating a Soft Type</a>.</p>
wt.part.WTPart	Part	A database object that represents a component of assembly in a product structure
wt.maturity.PromotionNotice	Promotion Request	A request to change the state of one or more objects to a new life cycle state with the selection of a review process.
wt.part.WTProductInstance2	Part Instance	A serialized copy of a product built according to a specific configuration
wt.change2.WTVariance	Variance	An authorization to depart from the as-designed configuration for a specific number of units or for a specified time period. Variances consist of deviation and waivers. A deviation is a planned departure from the as-designed configuration before the part is built. A waiver is a written acceptance of nonconformance.
com.ptc.wpcfg.doc.VariantSpec	Variant Specification	The collection of options specified for a particular generic part structure in preparation for creating a variant. The variant specification, in essence, defines the characteristics of the variant that should be created from the generic part structure.

**Note:** Not all Windchill solutions use all business object types and many additional object types are used by your Windchill solution to help manage administrative processes, such as updating user preferences, life cycles, and workflows. Additional information about the use of specific object types can be found throughout this guide.

## Data Types

All data stored in a Windchill solution are stored as objects of specific types. The type is identified when the object is created or imported. By typing data, you establish patterns for handling the data within the Windchill solution. For example, you can decide if part data is numbered automatically or manually and decide who has access to part data. Separate decisions can be made for each type of data by setting object initialization rules. A set of these rules is established when each context is created through the context template that is used. Additional object initialization rules can be set using the Object Initialization Rules administrator, accessed from the **Utilities** page of the context where you want the rule stored.

## Soft Types

In addition to the modeled object types that are provided out of the box, object types can have subtypes (known as *soft types*). Your Windchill solution provides some document soft types as part of its installation. Additionally, Windchill PDMLink and Arbortext Content Manager provide soft types for dynamic documents and bursting configuration files that are used when authoring document content through the Windchill connection to Arbortext Editor.

If your site needs additional types, you can create specific soft types from within the site or an organization context using the **Types** link.

## Visualization Data

Windchill Visualization Services integrates Windchill with ProductView, a PTC data visualization tool that allows you to view, annotate, analyze, measure, and animate CAD documents and parts. You may also view, mark up, and print the content of documents in ProductView. All information viewed or annotated in ProductView is saved and stored back into the Windchill database.

ProductView supports watermarking of 3D, drawings, images, and documents. Watermarks are defined in INI files created and edited using the ProductView watermark editor. The administrator that manages watermarks manually transfers the INI files from the watermarks directory into the Windchill server.

Windchill Visualization Services can also be used to integrate Windchill with the Arbortext Publishing Engine for creating representations from dynamic documents. Supported formats include PDF and HTML. This is done through the WVS publisher.

For more information about administering visualization data, see the [Visualization Services](#) chapter.

## CAD Data

Data files created by CAD applications and saved to Windchill using a workgroup manager or Pro/ENGINEER Wildfire are managed in Windchill as CAD documents.

CAD documents are associated with a soft type of the `wt.epm.EPMDocument` modeled object type and can be used to manage many different types of CAD files including parts, assemblies, drawings, layouts, and manufacturing files. In addition, a CAD document can have secondary content. The secondary content is typically used for additional supporting information or derived files such as STEP, IGES, JPEG, and others.

The Workgroup Manager CAD Document soft type defined as `<prefix>.DefaultEPMDocument` is established during the installation process, where `<prefix>` is generated by reversing the internet domain specified as part of the organization properties during the installation. For example, if the internet domain specified is `acme.com`, then the `<prefix>` used in the soft type name is `com.acme`.

Administrators can create and update CAD document templates. These templates can be used to create CAD documents.

## Dynamic Document Data

Windchill supports data files that are created by the Arbortext Editor. These files are saved to Windchill using the Windchill connection for the editor and are managed in Windchill as dynamic documents. The data files can then be published using the WVS publisher if the publisher has been configured for the Arbortext Authoring Application.

The Arbortext Editor and the Arbortext Publishing Engine are optional Windchill products that can be purchased through PTC.

Dynamic documents are associated with the `<prefix>.DynamicDocument` soft type or are possibly associated with a soft type that is a subtype of `<prefix>.DynamicDocument` (where `<prefix>` is generated by reversing the internet domain specified as part of the organization properties during the installation). For example, if the internet domain specified is `acme.com`, then the soft type for dynamic documents is named `com.acme.DynamicDocument`.

Using additional soft types helps categorize the types of dynamic documents and is often required when deploying a complex content authoring system that connects to the Arbortext Editor.

Understanding how Windchill and Arbortext Editor work together and the basics on setting up your Windchill PDMLink system to work with Arbortext Editor are covered in the [Getting Started with Windchill PDMLink and Arbortext Editor](#) guide.

For general guidelines on creating product and library contexts to use with dynamic documents, see [Configuring Products or Libraries for Dynamic Documents](#).

For information on creating soft type and soft attributes for use with Arbortext Editor and the WVS publisher, see [Managing Types and Attributes for Dynamic Documents](#).

For information on setting up the WVS publisher for publishing dynamic documents, see [Using Publish Rules](#) in the Visualization Services chapter.

## Document Data

General documents (such as Microsoft Word files and text files) stored in the Windchill database have the wt.doc.WTDocument object type or are possibly associated with a soft type that is a subtype of wt.doc.WTDocument. Using soft types helps categorize the types of documents.

Administrators can create and update document templates. These templates can be used to create documents.

Administrators can also change the preference settings related to documents (such as file name uniqueness and template file name length). These settings can be found under the **Documents** category in the Preference Manager. For information about the Preference Manager, see [Administering Preferences](#) in the Contexts chapter.

## Part Data

Parts stored in the Windchill database have the wt.part.WTPart object type.

Parts created in Windchill can be associated with a view. A view identifies what the part is used for. The Windchill installation provides two out-of-the-box views: design and manufacturing. Before allowing users to create parts, you can rename these views and add other views that make sense at your site by using the View Administrator utility. For more information, see [Views and View Associations](#).

## Auditing

The auditing utilities are designed to record and report user and system events for auditing and traceability purposes.



## Managing Windchill Processes

Windchill provides you with the following types of Windchill data processes that can be used with the business objects stored in Windchill:

- A life cycle process defines a set of phases and gates that manage the life of an object as it progresses from conception to obsolescence.
- A workflow process defines rules which allow workflow participants to view and pass along information, tasks, and documents.
- A change process establishes how to get changes made to parts and many other data types including documents, document soft types, and product instances.

These three processes work together to help you manage data. Workflows are often used to drive the life cycle. That is, the workflow process is used to transition from one life cycle state to another. In most cases, a workflow process is initiated from a life cycle. In any case, life cycle-managed objects obtain a life cycle when they are created. In obtaining a life cycle, an object can have a workflow process instance created as well. Change objects are life cycle-managed objects. Each change object starts a change process when it obtains its life cycle.

Use the Life Cycle Administrator to manage the life cycle templates that can be used. For details on managing life cycle templates, see the [Life Cycles](#) chapter.

Use the Workflow Administrator to manage the workflow templates that can be used. For details on managing workflow templates, see the [Workflow](#) chapter.

## Managing User Collaboration

User collaboration can be done using a specific Windchill solution and other PTC or third party products or using multiple co-installed Windchill solutions.

Windchill provides the following tools for collaboration:

- Windchill projects that are created using Windchill ProjectLink provide participants with a place in which they can share information. This information can include data that resides in either a Windchill PDMLink or Arbortext Content Manager solution.
- Pro/ENGINEER Wildfire provides users with the ability to share CAD drawings and other design-related information. For administration information, see the *Using Pro/ENGINEER Wildfire with Windchill* guide.

## Additional Administrative Groups

The following additional administrative groups are automatically created for all Windchill solutions to help define users for specific administrative activities in your solution:

- Attribute Administrators
- LifeCycleAdministrators
- Replication Managers
- Type Administrators
- WorkflowAdministrators
- Workflow Authors

For example, those users in the Attribute Administrators group can manage the metadata for attributes. Those users in the LifeCycleAdministrators group become participants in the Default life cycle template, when that template is used.

If Windchill Supplier Management is installed, the installation creates the following administration group:

- Supplier Administrators

For information on Supplier Management activities, see the *Windchill Supplier Management Administrator's and User's Guide*.

By using the Principal Administrator, you can add users to any of the administration groups.

Additional types of administrators are available as appropriate for the application contexts within your Windchill solution:

- An organization administrator manages a specific organization.
- A product manager manages a specific product.
- A library manager manages a specific library.
- A project manager manages a specific project.
- A program manager manages a specific program.
- A shared team manager manages a specific shared team, and acts as a context manager for any contexts that use the shared team.

For additional information about administrators, see [Establishing Administrators](#).

## Post-Installation Activities

Before allowing users to access the Windchill solution, be sure to do the following activities:

- Complete the activities described in the [Getting Started](#) chapter.
- Determine if there are additional organization contexts that you want to create and create them. See the [Organizations](#) chapter.
- Determine if additional organization principals are needed and whether or not to allow these organizations to own parts and documents.

By default, all parts and documents are owned by the organization from which they are created. For information on how to set up the ability to choose which organization owns a part or document, see the [Organizations](#) chapter.

- Determine whether you want audit reports enabled. See the [Auditing](#) chapter.

## 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.  <b>Note:</b> On UNIX systems where you have multiple instances of Windchill installed under the same user account, settings made to WT_HOME and SQLPATH environment variables by using this -w option are overridden by any settings to these same variables in the user's .cshrc, .login, and .profile shell initialization files.
--java=JAVA_EXE	The Java executable. Default is the wt.java.cmd variable value specified in the \$WT_HOME/code-base/wt.properties file.
-cp, --classpath=PATH	Java classpath. Default is the wt.java.classpath variable value specified in the \$WT_HOME/code-base/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 — with no arguments generates the help report</p>
CLASS [CLASS_ARGS]	<p>Run a Windchill class with optional class arguments. For example:</p> <p>windchill wt.load.Developer -UAOps</p>

## About the windchill shell

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

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

```
windchill shell
```

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

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

## About the xconfmanager Utility

The xconfmanager is a command-line utility that is used to add, remove, and modify the properties in the Windchill property files. You should use the xconfmanager (or the System Configurator) to manipulate properties; you should not manually edit property files.

There are property files that should not be modified using the xconfmanager. The following registry files are managed by Windchill Information Modeler and they also should not be edited manually or using the xconfmanager:

- associationRegistry.properties
- classRegistry.properties
- descendentRegistry.properties
- modelRegistry.properties
- moduleRegistry.properties
- moduleDir.properties
- debug.properties

The xconfmanager utility saves your changes in the site.xconf file and provides an option to generate updated property files using those changes. The site.xconf file contains changes made to Windchill property files, starting with installation and continuing with each use of the xconfmanager utility or the System Configurator. The xconfmanager utility is located in the <Windchill>/bin directory.

This chapter describes only the information and instructions necessary to modify specific Windchill properties. A full description of the xconfmanager utility and management of the Windchill property files is documented in the *Windchill System Administrator's Guide* in the Administering Runtime Services chapter.

Anyone with write access to the XCONF files and the property files under the Windchill installation directory can successfully run the xconfmanager utility. The xconfmanager is executed from the command line from within a windchill shell. See the [About the windchill shell](#) section for more information about the windchill shell.

The syntax of xconfmanager command with only commonly used parameters is as follows:

```
xconfmanager {-h} {-r <product_root>} {-s <property_pair>}
{-t <property_file>} {--add <property_pair>}
{--remove <property_pair>} {--reset <property_names>}
{--undefine <property_names>} {-d <property_names>} {-p}
```

For the purposes of modifying Windchill properties, you will primarily use the -s, -t, and -p parameters as follows:

- Use the **-s** (--set) parameter to identify the relevant property and specify the new property value. See the [Formatting Property Value Guidelines](#) section (below) for information about formatting the *<property\_pair>* value.
- Use the **-t** (--targetfile) parameter to specify the directory location of the property file. If the file name or path contains spaces, you must enclose the *<property\_file>* value in double quotes (" "). It is recommended to use a fully qualified file name to ensure an accurate reference to the file is made.
- Use the **-p** (--propagate) parameter to propagate the changes made to the XCONF files into the property files being modified in order to keep the XCONF and the property files in sync with one another.
- Use the **-h** (--help) parameter to view the help for xconfmanager. The help describes all xconfmanager parameters.

Additionally, you can add and remove property values from properties that are declared as a multi-valued properties using the following parameters:

- Use the **--add** parameter to add the value specified at the end of the set of ordered values already defined in the property.
- Use the **--remove** parameter to remove the value specified from the set of ordered values already defined in the property.

**Tip:** If you are unsure as to whether 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.

Some examples of using the xconfmanager utility are as follows:

- xconfmanager is run from the windchill shell. To open a windchill shell, execute the following command at a command prompt:

```
windchill shell
```

- To display xconfmanager help, execute the following command from the windchill shell:

```
xconfmanager -h
```

- To display the current settings for a property, execute the following command from the windchill shell:

```
xconfmanager -d <property_names>
```

*<property\_names>* is a comma-separated list of property names. This means that you can display the current settings for multiple properties by executing one command.

- To change a property value, execute the following command from the windchill shell:

```
xconfmanager -s <property_name>=<property_value>
-t <property_file> -p
```

**Tip:** Use the fully qualified name of the property file to ensure an accurate reference. If you are sure that there is only one property file that is known to xconfmanager containing the property, you can omit the -t parameter. When setting a value for a new property not in a property file, you must include the -t parameter to name the property file to which the property is added.

- To add a new classpath entry to the Windchill 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.

**Tip:** The previous example command did not include the target file in the -t parameter since the property is known to be in only wt.properties.

## Formatting Property Value Guidelines

The following guidelines will help ensure that you set properties correctly on the command line when using xconfmanager:

- To specify a property whose value contains characters that might be interpreted by your shell (such as spaces and special characters), escape them using the appropriate technique for the shell you are using.

On a Windows system, you can include spaces in a value by enclosing the argument with doubles quotes or you can escape the space character with ^. For example, use either of the following:

```
-s "wt.inf.container.SiteOrganization.name=ACME Corporation"
-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 \. For example, use either of the following:

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

- In many UNIX shells, the use of a backward slash (\) escapes the following character as a literal. In most cases, using forward slashes (/) in file paths is a simple way to specify a path without having to know the intricacies of your shell's command line escaping rules.



- 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 the shell does not interpret it 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, you should not need to escape other values to be compatible with XML or property file syntaxes. The xconfmanager escapes property names and values automatically if necessary.



# 3

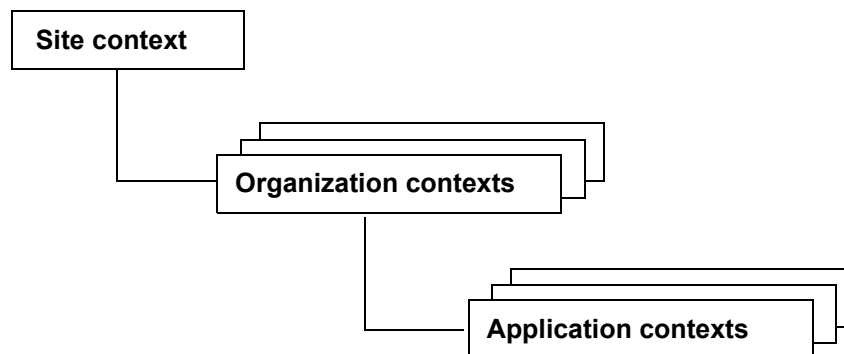
## Contexts

This chapter provides the overall details relating to working with contexts. Later chapters assume that you have read the information in this chapter.

Topic	Page
Distributed and Hierarchical Administration .....	3-2
Context Administrative Items .....	3-4
Creating the Contexts from which Users Work .....	3-23
Administering Domains and Policies .....	3-25
Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects.....	3-35
Administering the Versioning of Parts, Documents, and CAD Documents .....	3-36
Administering Preferences .....	3-38
Administering Saved Searches.....	3-40

## Distributed and Hierarchical Administration

Windchill contexts provide the framework for collecting and finding related information. The set of contexts in your Windchill solution have a hierarchical relationship. The following depicts the basic context hierarchy:



The site context can have one or more child organization contexts. An organization context can have one or more child application contexts.

Application contexts include:

- Products
- Libraries
- Projects
- Programs

Data, such as template files, can be distributed among the contexts. For example, you can define general document templates, such as those used for presentations or memos, at the top level of the hierarchy (in the site context). The document templates are available to all contexts. Then you can define progressively more specific templates at each layer in the hierarchy, such as in an organization context or a library context. In a child context, you can also define templates with the same name as those templates in a parent context so that the templates in the child context can override and be used in place of templates in a parent context.

With distributed administration, application context administrators are responsible for their own administrative tasks. So, for example, each product, library, project, and program can have its own administrators (called product, library, project, and program managers). Additionally, if you are using shared teams in context teams, the shared team manager (established in the shared team) becomes the context manager for each application context that uses the shared team. This allows you to easily assign administrative duties for multiple contexts to one or more individuals.

To support distributed administration, the administrative utilities are context aware. For example, opening the Policy Administrator in the context of a library initially displays the domains that are in the library context and the domains that are ancestors of the library context domains. Having context aware utilities allows the delegation of administrative duties to users who are recognized as application context managers.

With hierarchical administration, contexts inherit administrative items from parent contexts (or, in the case of policies, from parent domains). Administration performed at the level of a parent context is applicable to all of its child contexts. Except for policies, a child context can choose to override the administrative items of its parent contexts. For policies, rules from all parent domains are merged with the rules in a current domain to form the policy for the current domain.

In general, always think of performing administration duties from within a context, as follows:

- Site administrators can create, modify, delete, and view administrative items in the site context.
- Organization administrators can create, modify, delete, and view administrative items in the given organization context. They can view and override administrative items defined in the site context.
- Application context managers can create, modify, delete, and view administrative items in the given application context. They can view and override administrative items defined in the parent organization context and the site context.
- Site and organization administrators can administer child contexts; however, they do so by going into the child context and performing their administrative duties there.

**Note:** Windchill users generally perform their activities within an application context, rather than in an organization or site context.

Non-administrative Windchill users generally do not create, modify, or delete administrative items; however, they have visibility to, and are affected by, the following administrative items:

- The administrative items that are defined in a given application context.
- The administrative items that are defined in the parent organization and site contexts, but not overridden in the application context.

## Context Administrative Items

Each context can be populated with the following types of administrative items:

- Context configuration
- Context structure
- Context participation
- Context policies
- Context data types and attributes
- Templates, including context templates, process templates, and data templates
- Context object initialization rules
- Context preferences

After you have created a context, you can update many of the administrative items that are associated with the context. The contexts that can be updated include the site context, as well as organization and application contexts.

Use the administrative utilities and specific user actions to edit the administrative items in a context. These utilities and actions are usually available to the administrator of the context. For the details on Windchill administrators, see [Additional Administrative Groups](#) in the Administration Overview chapter.

Each type of context has a slightly different set of administrative items that can be updated. For example, you can edit the set of life cycle and workflow templates provided in organization, product, and library contexts. Also, you can only edit the set of product, library, project, and program templates that are provided in an organization or the site context. This is because product, library, project, and program templates are not included in application contexts.

The following sections provide descriptions of administrative items, information on what is installed in the site context for each item, and how to edit the items.

### Context Configuration

Configuration items identify the type of context and other miscellaneous information about the context.

There are three general types of contexts:

- Site – The site context is the top-level context. There can be only one site context.
- Organization – Organization contexts are always children of the site context. There is always at least one organization context required to have an operational Windchill solution.

- Application – Application contexts are always children of an organization context. There are four types of application contexts:
  - Product
  - Library
  - Project
  - Program

The context configuration can include the following additional items:

- You decide whether the context is public or private. This option is only available when you are not using a shared team.

Private contexts, by default, inherit access control rules from the /Private domain of the organization.

Public contexts, by default, inherit access control rules from a solution-dependent public domain within the organization context. For Windchill solutions with projects and programs, the public domain is the /Default/Project domain. For Windchill solutions including products and libraries, the public domain is the /Default/PDM domain. This configuration allows an administrator to:

- Create policies in the organization context /Default domain that apply to all public child contexts.
- Create policies in the /Default/Project and /Default/PDM domains that apply to solution-specific child context.
- Create policies in the /Private domain that apply to all private child context.

For contexts created with shared teams, the contexts inherit access control rules from the shared team domain in the organization context where the shared team was created. The domain has the same name as the shared team. This configuration allows an administrator to:

- Create policies in the organization context shared team domain that apply to all child contexts that use that shared team.
- Create policies in the /Default domain that apply to solution-specific child context.

- For project contexts, you decide whether or not data can be shared to other contexts.
- For an organization context in Windchill solutions including projects and programs, you decide whether the project creators group is auto-populated with members of the organization. The project and program creators groups determine who can create projects and programs.

The configuration of the context is set when the context is created based on the options chosen through the user interface or through data loading, and are not set in a template.

## Editing the Context Configuration

The configuration of an organization or application context is set when the context is created based on the options chosen through the user interface. Only a few of the options can be updated:

- For an organization context, update the context using the **Edit Organization** icon or the **Edit** action on the organization information page. Site administrators can navigate to the **Organizations** page on the **Organization** tab and select the **Edit** action for the organization they want to update.

For details on how to edit and what can be edited, view the help available from the **Organization** tab.

- For an application context, edit the context using the **Edit** icon or the **Edit** action from the context information page. First, navigate to the context by selecting the context tab and then selecting the specific product, library, project, or program that you want to edit.

For details on how to edit and what can be edited, view the help available from the context tab.

## Context Structure

Structure items identify the domains, cabinets, folders, notebook folders, discussion topics, and reference folders for notebooks that are in the context and the domain inheritance scheme that is in place. Contexts define a structure in which related information is organized. This structure can be represented by a folder hierarchy, product structure hierarchy, collection of discussion topics, or predefined milestones in a schedule. The structure defined by the context enforces consistency and improves efficiency.

The use of the context structure is very apparent when you look at how domains can help organize rules for users. For example, the Administrators group and Administrator user are associated with the /System domain and, therefore, are segregated from other users by default. The other users are associated with child domains of the /User domain. Both the /System and /User domains are in the site context, and the child domains of the /User domain can be in the site context or in an organization context.

## Installed Site Context Structure

The site container has the top-level / (root) domain, the /System, /Default, /User, and /SessionIterationDomain domains that are children of the root domain, and the /Unaffiliated domain that is a child of the /User domain. (For an explanation of these domains, see [Administering Domains and Policies](#).)



For the Windchill PDMLink, Arbortext Content Manager, and ProINTRALINK 9.0, there are no folders installed in the site context.

For Windchill ProjectLink, the following folders, which are associated with /Default domain in the site context, are installed:

- Change Log
- General
- Policies

## Editing Context Structure

The context structure is set either by the load files used (in the case of the site context) or by the template selected when the context is created. The template (and its underlying code) sets the domains, cabinets, folders, notebook folders, forum topics, and reference folders that are in the context and the domain inheritance scheme, which identifies parent and child domain relationships.

Although you can modify the set of domains in a context using the Policy Administrator, you should refrain from doing so unless the purpose of the domain is clearly defined. For example, a new domain may be needed to provide a unique set of access control policy rules for a specific folder.

Windchill solutions do not expose the creation of cabinets through their user interface. Use the default cabinets defined. Administrators can update the set of folders available in a context by navigating to the specific product, library, project, program, or organization and then clicking the **Folders** link. Similarly, forum topics can be created in a product, library, project, or program by clicking the **Forum** link that is available from the product, library, project, or program.

## Context Participation

Participation items establish the following:

- Roles that are automatically available in a context.
- System groups that are automatically created in a context.

Product, library, project, and program contexts are associated with teams of users and user-defined groups. Any user can display all products, libraries, projects, or programs in which the user is identified as a team member. The users and user-defined groups in a team are associated with roles that identify the responsibilities and permissions of the team members. For example, the Product Manager role establishes who is in charge of the product; similarly, the Project Manager role establishes who is in charge of a project.

The set of roles and system groups that are automatically available in a context consist of the roles and system groups established in parent contexts, as well as those defined specifically for the context.

If the team established in a context includes a shared team, the shared team roles and system groups are defined and managed in the organization context where the shared team was created.

## Installed Site Context Participation

Through the site context, roles and system groups that are useful throughout your Windchill solution are made available to the site.

### Roles

The roles that are available to the entire site are defined in the `<Windchill>\src\wt\project\RoleRB.rbin` resource bundle, which is translated to all supported languages and made available through the application clients.

All of these role names are available to all child organizations and then to the product, library, project, and program contexts created within those organizations. To see the roles, you can click the **Roles** link from the **Organization** tab. For products, libraries, projects, and programs, click the **Team** link from the corresponding tab.

### Groups

The following group is defined in the site context /System domain:

Administrators (known as the system or site administrators)

**Note:** This group defines the administrators for your entire site as described in [Distributed and Hierarchical Administration](#).

The following groups are defined in the Site container /User/Unaffiliated domain:

Attribute Administrators  
LifeCycleAdministrators  
Type Administrators  
Unrestricted Organizations  
WorkflowAdministrators

These groups are used during the normal operation of your Windchill solution and should not be removed. Domain-based access control rules are automatically loaded in the site container /System domain granting permissions to members of these groups, except the Unrestricted Organizations group. The rules for the members of the Unrestricted Organizations group are loaded in the site context /User domain.

## Editing Context Participation

Context participation is set by the template selected when the context is created. The roles and groups defined in the template can be modified in the organization context as well as in an application context. Those roles and groups defined in the organization are inherited by the child contexts of the organization.

From within an organization, use the **Groups** link to add or update user-defined groups and use the **Roles** link to add or remove roles from the organization.

From within a product, library, project, or program, use the **Team** link to add or remove roles. From the **Members** table, you also add and remove members from the roles. For additional information, see the [Teams](#) chapter.

## Context Policies

Context policy items can include the following:

- Domain-based access control rules that establish the access control against specific principal, object type, life cycle state, and domain combinations.

For example, there can be an access control rule for objects with the wt.doc.WTDocument data type in the /Default domain that gives read permission to the Engineers system group.

- Indexing rules that define which collections an object is included in when the object of a specified object type and domain combination moves to a specific life cycle state. Collections are used to create indexing lists which help improve performance when searching for objects.
- Notification rules that define which principals get notified when a specified event occurs for an object type and domain combination.

Contexts provide a means for controlling access to the contained information. In application contexts, access is controlled through context team membership, policy rules, and ad hoc rules applied to folders and roles. Context contents can be restricted so that access is limited to the members of a context team, or so that the context information can be made more broadly available to the enterprise through policies that grant specified user-defined groups in an organization access to specific object types.

The policies that are in effect in a context are determined by the policies set in the domains that are in the current context, as well as those set in the parent domains. For details, see [Administering Domains and Policies](#).

## Installed Site Context Policies

The site context policies that are set consist of the domain-based access control rules and one indexing rule. No notification rules are set in the site context domains, and no policy rules are set in the site context /SessionIterationDomain domain.

**Note:** Your solution may vary from the following description, as the name of the Administrators group, and the names of the some initial domains are configurable from the wt.properties.xconf file prior to the installation.

The following sections describe the rules that are set in the site context.

## Access Control Rules for / (Root) Domain

The following domain-based access control rules are set when the data is loaded during the installation. The rules are in the site container / (root) domain for all life cycle states.

**Note:** These rules ensure that users can operate within the Windchill solution and should not be changed without fully understanding the reason for the change.

Object Type	Principal	Permissions	Comment
AccessPolicyRule	ALL	Read	Allows organization and application context administrators to see inherited access rules.
AdministrativeDomain	ALL	Read	Allows all users to view domains.
DeliverableDefinition	ALL	Read and Create	Allows all users to create and read deliverable definitions.
EngineState	ALL	Read	Allows all users read access.
EPMDocConfigSpec	ALL	Full Control (All)	Allows all users to perform configuration management of business objects within their workspace when using Pro/ENGINEER or other workgroup managers.
ExchangeContainer	ALL	Read	Allows all users to complete a variety of general actions.
Meeting	OWNER	Full Control (All)	Grants owners full access to meetings that they own. For information about setting up meetings with WebEx, see the <i>Windchill System Administrator's Guide</i> .
NotificationSubscription	Administrators	Full Control (All)	Grants administrators full access.
NotificationSubscription	ALL	Read and Create	Grants all users read and create access.
NotificationSubscription	OWNER	Full Control (All)	Grants the owner full access.
Team	OWNER	Read	Grants the team owner read access.

<b>Object Type</b>	<b>Principal</b>	<b>Permissions</b>	<b>Comment</b>
WTDocumentConfigSpec	ALL	Full Control (All)	Grants full control to all users.
WTMarkup	ALL	Read, Download, and Create	Allows all users to create and read markups. These permissions are required because view markups are not life-cycle managed.
WTMarkup	OWNER	Modify, Modify Content, and Delete	Allows the owner of a markup the ability to modify and delete it.
WTObject	Administrators	Full Control (All)	Grants full control to all site administrators.
WTPartConfigSpec	ALL	Full Control (All)	Allows all users to perform configuration management of business objects within their workspace when using Pro/ENGINEER or other workgroup managers.

### Access Control Rules for /User Domain

The following domain-based access control rules are set in the site context /User domain for all life cycle states:

Object Type	Principal	Permissions	Comment
WTGroup	Unrestricted Organizations	Read	Allows read access to groups for the organizations that are in this group.
WTOBJECT	OWNER	Full Control (All)	Grants full control to the owner of an object.
WTUser	Unrestricted Organizations	Read	Allows read access to users for the organizations that are in this group.

### Access Control Rule for /User/Unaffiliated Domain

The following domain-based access control rule is set in the site context /Unaffiliated domain (which is a child of the /User domain) for all life cycle states:

Object Type	Principal	Permissions	Comment
WTPrincipal	ALL	Read	Allows read access to principals for all users.

### Access Control Rules for /Default Domain

The following domain-based access control rules are set in the site context /Default domain for all life cycle states:

Object Type	Principal	Permissions	Comment
Meeting	ALL	Read	Allows read access to meetings for all users. This rule is used to provide visibility of WebEx meetings for all users. For information about setting up meetings with WebEx, see the <i>Windchill System Administrator's Guide</i> .
RelationshipMap	ALL	Read	This object type is used for collecting dependent objects when archiving objects. For details on archiving, see the <i>Windchill Archive Administrator's Guide</i> .
Report	ALL	Read	Allows all users read access to report objects.
ReportTemplate	ALL	Read	Allows all users read access to report templates.

## Access Control Rules for /System Domain

The domain-based access control rules set in the site context /System domain are used to control access to administrative objects. For normal operations, you should not modify these rules.

To view the rules, access the Policy Administrator from the site context. For information about accessing the Policy Administrator, see [Using the Policy Administrator](#).

## Indexing Rule for / (Root) Domain

The following indexing rule is set in the site context / (root) domain for all life cycle states:

Object Type	State	Collections	Comment
WTOBJECT	All	WbLib	Indexes all objects in all states and puts the indexes into the WbLib collection.

## Updating Context Policies

Use the Policy Administrator to update context policies. For information about using the Policy Administrator, see [Administering Domains and Policies](#).

## Context Data Types and Attributes

Data type and attribute items identify data types (defined as soft types) and attributes that are in addition to the Windchill modeled classes.

The data types available in a context are determined by the data types defined in the organization context, as well as those defined in the site context.

For example, if your organization context has defined Plan and Report document soft types, then these soft types are available in any project, program, product, or library context created from your organization context.

## Installed Site Context Data Types and Attributes

All data types that are modeled in your Windchill solution are available for administrator use from the site context.

If Windchill ProjectLink is installed, then the following document soft types are also available from the site context:

- Agenda
- General
- Minutes
- Plan
- Presentation
- Reference Document

## Editing Context Data Types and Attributes

You can use the **Types** link from the **Site** and **Organization** tabs to view and edit data types:

- To create and edit site-wide types, navigate to the **Site** tab and click the **Types** link.
- For organization-specific types, access an organization and click the **Types** link.

Attributes can be associated with types by first defining the attributes using the **Attribute Definition Manager** tab which is accessed from the **Type and Attribute Manager** utility on either the site or organization tabs.

Attributes can be created or modified in the site or organization context by a site administrator. An organization administrator can create and modify attributes within the organization context, once the appropriate permissions have been set. Add the organization administrator to the **Attribute Administrators** group using the **Principal Administrator** utility to grant permission to create attributes.

Once the attributes are defined, use the **Type Manager** to add attributes to data types. For more information, see the Type and Attribute Manager online help.

**Note:** The internet domain defined on an organization principal is important when creating new soft types. The internet domain is used to distinguish which organization owns the type.

During the installation process, an organization principal (the organization object) is created and associated with the site context. Any types owned by this organization principal are available to all organizations.

Each organization principal has an internet domain that is used to identify the organization contexts for soft types and soft attributes. When soft types and soft attributes are created, Windchill provides a suggested prefix for the name of the new type or attribute. The suggested prefix is based on the context from which the **Type Manager** tab, or **Attribute Definition Manager** tab is launched. The suggested prefix is the reverse of the internet domain for the owning organization principal of the organization context. For example, an organization with the internet domain, acme.com would suggest the prefix, com.acme.

**Note:** An error will occur if a type or an attribute is created within an organization context and it has an internet domain that is not associated with a known organization context.

For additional information about updating data types and attributes, see [Overview of Types and the Runtime Typing Capability](#).



## Templates

Template items identify the templates that provide users with the required information used when creating and processing Windchill objects.

Windchill provides the following types of templates (not all types of templates are available from all Windchill solutions):

- Context templates provide the required and optional administrative items that are used to create context. There are five types of context templates:
  - Product templates define default values and other information, such as team roles and access policies, which are used when an administrator creates a product.
  - Library templates define default values and other information, such as team roles and access policies, which are used when an administrator creates a library.
  - Project templates define default values and other information, such as folder structure and team roles, which are used when an administrator creates a project.
  - Program templates define default values and other information, such as folder structure and team roles, which are used when an administrator creates a program.
  - Organization templates define default values and other information, such as folder structure and user-defined groups, which are used when an administrator creates an organization context. Each Windchill solution provides a unique set of organization templates.
- Document templates provide content files and default values, which are used when users create different types of documents. For example, content might include skeleton documents for memos or reports.
- CAD document templates define the content and default values, which are used when users create CAD documents.
- Life cycle templates define the phases and gates associated with various business objects when the objects are initialized.
- Report templates define default values, which are used when users run reports.
- Team templates define default roles, which are used when users populate teams associated with life cycles.
- Workflow templates define default values, which are used when users initiate a workflow.

- Note templates provide a starting point for creating a note from standardized text that can be associated to parts. Note templates can be modified and attachments can be associated to communicate important information about a part.
- Task form templates contain various form fields, such as labels, text fields, and radio buttons. These form fields are place holders to display the attributes of a task like variables, process name and type to the task participant. For more information about [Using Task Form Templates in a Workflow](#), see the [Workflow](#) chapter.

## Installed Site Templates

The templates loaded in the site context are associated with the /System domain.

The following sections describe the templates that are loaded and how to remove or hide templates.

**Note:** **CAD Document Templates** and **Note Templates** appear in the list for Windchill PDMLink and Windchill ProjectLink, but do not have installed templates.

## Organization Context Templates

When Windchill PDMLink is installed, the following organization context templates are loaded in the site context:

- General (PDM)
- DemoTemplate (not enabled)

When Windchill ProjectLink is installed, the following organization context templates are loaded in the site context:

- General
- Enterprise
- Supplier
- Customer

For additional information about organization context templates, see [Out-of-the-box Organization Templates](#) in the Administering Organizations chapter. For specific information about creating organization context templates, see [Creating Context Templates](#), in the Context Templates chapter.

## Workflow Templates

The following workflow templates are loaded in the site context for all Windchill solutions:

- Submit
- Review

For Windchill PDMLink, the following workflow templates are loaded in the site context (these templates are not used by Arbortext Content Manager or ProINTRALINK 9.0):

- Problem Report Workflow
- Change Request Workflow
- Change Notice Workflow
- Change Activity Workflow
- Promotion Request Approval Process
- Promotion Request Review Process
- Variance Workflow

For Windchill ProjectLink, the following workflow templates are loaded in the site context:

- Approval Process
- Notify Process
- Release Process
- Review Process
- Two Level Approval Process

## Life Cycle Templates

For the details on out-of-the-box life cycle templates, see [Out-of-the-box Life Cycle Templates](#) section in the Administering Life Cycles chapter.

## Team Templates

Team templates are used with life cycle templates.

For Windchill PDMLink, the following team templates are loaded in the site context (these templates are not used by Arbortext Content Manager or ProINTRALINK 9.0):

- Default
- Change Activity Team
- Change Notice Team
- Change Request Team
- Promotion Request Team
- Variance Team
- Problem Report Team

For Windchill ProjectLink, there are no team templates loaded.

## **Document Templates**

For Windchill ProjectLink, the following document templates are loaded in the site context:

- Agenda Template
- Memo Template
- Minutes Template
- MS Project Plan Template
- Presentation Template
- Requirements Template

No document templates are loaded for Windchill PDMLink.

## **Project Templates**

For Windchill ProjectLink, the following project templates are loaded in the site context:

- Custom
- Design
- General
- Manufacturing

No project templates are loaded for Windchill PDMLink.

## **Program Templates**

For Windchill ProjectLink, the following program template is loaded in the site context:

- General

No program templates are loaded for Windchill PDMLink.

## **Product Templates**

For Windchill PDMLink, the following product templates are loaded in the site context:

- Demo Product Template (not enabled)
- General Product
- Product Design

No product templates are loaded for Windchill ProjectLink.

## **Library Templates**

For Windchill PDMLink, the following library templates are loaded in the site context:

- Document Library
- General Library
- Part Library

No library templates are loaded for Windchill ProjectLink.

## Report Templates

For Windchill PDMLink, the following report templates are loaded in the site context (these templates are not used by Arbortext Content Manager or ProINTRALINK 9.0):

- All Average Change Notice Completion Time
- All Average Change Request Completion Time
- All Average Problem Report Completion Time
- All Change Notice Influx and Resolution
- All Change Request Influx and Resolution
- All Open Change Notices
- All Open Change Requests
- All Open Problem Reports
- All Problem Report Influx and Resolution
- Average Change Notice Completion Time
- Average Change Request Completion Time
- Average Problem Report Completion Time
- Change Notice Influx and Resolution
- Change Request Influx and Resolution
- Change Request Track Breakdown
- ChangeNotice\_ChangeAdmin\_Given
- ChangeNotice\_ChangeAdmin\_NotGiven
- ChangeRequest\_ChangeAdmin\_Given
- ChangeRequest\_ChangeAdmin\_NotGiven
- Domain Reports
- Open Change Notices
- Open Change Requests
- Open Problem Reports
- Overdue Running Processes
- Problem Report Influx and Resolution
- ProblemReport\_ChangeAdmin\_Given
- ProblemReport\_ChangeAdmin\_NotGiven
- Process Initiated by Current User
- Variance\_ChangeAdmin\_Given
- Variance\_ChangeAdmin\_NotGiven

For Windchill ProjectLink, the following report templates are loaded in the site context:

- Project Action Items
- Project Activities
- Project Deliverables
- Project Milestones
- Project Summary Activities
- Project Tasks
- Project-Program Status Report Query
- Projects Created Query
- Projects Owned Query

## Task Form Templates

The following task form templates are loaded in the site context for all Windchill solutions:

PBO Class	Type	Name
wt.fc.WTObject	XDP	default
wt.fc.WTObject	JSP	default
wt.Change2.WTChangeActivity2	JSP	default
wt.Change2.WTChangeIssue	JSP	default
wt.Change2.WTChangeOrder2	JSP	default
wt.Change2.WTChangeRequest2	JSP	default
wt.maturity.PromotionNotice	JSP	default

## Editing Templates

To edit templates, navigate to the **Templates** table by clicking the **Templates** link within a specific context and then select the type of template from the **Current View** drop-down list. The list of templates that are available from the context appears in the table. Each type of template has its own method of editing the templates. Generally, editing can be initiated either by clicking an icon at the top of the **Templates** table or selecting an action from a row in the table.

For details on templates and how to edit them, see the help that is available from the **Templates** table. For specific information about editing context templates, see the [Context Templates](#) chapter.

## Removing, Hiding, or Disabling Templates

You can remove, hide, or disable the templates that have been loaded. If you do not want a template to be used by anyone, PTC recommends that disable the use of the template instead of removing it. By disabling it, the template is not available for use in any context, but remains in the system in case it is needed in the future. If you do not want a template used in a specific context, but want to allow its use in other contexts, you can hide the template in the specific context and show the template in other contexts.

To view the templates that have been loaded, click the **Site** tab and then click the **Templates** link to display templates. Select the type of template from the **Current View** drop-down list to display the templates in the table. Then you can use the disable row action to make individual templates unavailable throughout the user interface or use the hide action to hide the visibility of individual templates in a specific context. For additional help on template actions, click the help icon on the **Templates** table to display the templates help.

## Managing Document Template Preferences

When your Windchill solution is installed, default preference settings for document templates are established. For example, the default name of a document that is created from a template is set in the **Document from template filename** preference. You can change the defaults to better meet the needs of users by opening the Preference Manager from within the context in which you want the settings to take effect. Document template preferences are listed under the **Documents** category. For information about the Preference Manager, see [Administering Preferences](#).

## Object Initialization Rules

Object initialization rule items consist of documents that contain XML elements that are formatted according to the object initialization rules DTD and are used in the following activities:

- Initializing objects -- Rules can define specific default attribute values that are used when instances of objects of a specific type are created.
- Constraining the display characteristics of attribute values -- Rules can set the following types of constraints on attributes:

Hidden -- the user interface does not display a value or label for the attribute.

Immutable -- The user cannot change the attribute value that is being displayed.

Server assigned -- The user interface does not display a value for the attribute; the value is generated and assigned when the instance of the object is saved.

Pregenerated -- The user interface displays a value for the attribute. The value is generated before the attribute is presented in the user interface.

Discrete set -- The user interface is constrained to picking from a discrete set of values.

Additionally, you can qualify when default values are set or when constraints are used for Windchill attributes by integrating the use of conditional logic algorithms in object initialization rules.

For details on the use of object initialization rules, see the [Object Initialization Rules](#) chapter.

## Installed Site Object Initialization Rules

The object initialization rules established for each Windchill solution vary slightly, but include the following types of rule definitions:

- Out-of-the-box numbering and versioning schemes for parts, documents, CAD documents, dynamic documents, and change objects. These schemes are simple Oracle sequences that have been loaded into the Windchill database. Each starts at 1 and increments by 1.
- Default attribute values for folder paths, life cycles, and team templates for many object types.
- Display constraints on attribute values such as numbering, life cycles, team templates, and organization identifier.
- Conditional logic that allows for variations when setting constraints, numbering, versioning, and default attribute values.

For the complete set of object initialization rules established during the installation process, log in as the site administrator and navigate to the **Utilities** page on the **Site** tab. Open the **Object Initialization Rules Administrator** to view the rules.

**Note:** The rules set in the site context work with the installed interface for each Windchill solution. PTC recommends that you do not modify the out-of-the-box object initialization rules that are set in the site context without understanding what the change affects.

Windchill PDMLink provides a demonstration template named Product Design that uses a different versioning scheme and two life cycle templates that you can use in evaluating what object initialization rules should be set for your site. For more information on the Product Design template, see [Out-of-the-box Product and Library Context Templates](#) in the Products and Libraries chapter.

For additional information about the out-of-the-box numbering and versioning schemes, see [Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects](#) and [Administering the Versioning of Parts, Documents, and CAD Documents](#).

For information about using the Object Initialization Rules Administrator, see the [Object Initialization Rules](#) chapter.

## Adding and Changing Object Initialization Rules

You can add and change object initialization rules by using the Object Initialization Rules Administrator or by including the rules in a context template.

For more information, see the [Object Initialization Rules](#) chapter.



## Context Preferences

Preferences are used enable, specify, or otherwise affect functionality within a particular context. For more information, see [Administering Preferences](#).

## Creating the Contexts from which Users Work

All Windchill solutions provide the capabilities for creating organization and application contexts.

There is only one site context, which is created when your Windchill solution is installed. No additional site contexts can be created.

**Note:** The base data installation can create an organization context. If one was not created by loading the base data, the site administrator creates the organization context from the **Organization** tab.

Application context managers can create application contexts. You become an application context manager for products, libraries, projects, or programs by being in the creators group for products, libraries, projects, or programs. The creators groups are maintained from the **Creators** link on the **Organization** tab. When you create an organization from Windchill ProjectLink, you can choose to automatically add those organization members who log on to the project creators group; you manually add users to the program creators group. In all other Windchill solutions, you manually add users to the product and library creators groups.

As part of the process of creating a context, you select a context template. Context templates provide the administrative items (as described in [Context Administrative Items](#)) that you want set in the context, thus establishing the initial context framework for users. For the details on creating context templates, see the [Context Templates](#) chapter.

For information about creating organization contexts, see [Creating an Organization Context](#) in the Organizations chapter.

For information about creating product and library contexts, see [Products and Libraries](#).

For information about creating project and program contexts, see [Projects and Programs](#).

## Using Out-of-the-box Context Templates

Your Windchill solution provides a set of out-of-the-box context templates that you can use when creating contexts.

The text in the templates has been translated, and translated files are provided in the loadFiles and loadXMLFiles directories where your Windchill solution is installed. As part of the installation process, the installer can change the load set so that the base data loaded includes files translated into the language of choice at your site. The base data that is loaded determines which language set of context templates is available.

For details on setting which language files are used when loading base data, see database initializing and data loading in the *Windchill Installation and Configuration Guide - Advanced*.

Use one of the out-of-the-box organization templates when creating organization contexts, or you can create a customized organization template, as described in the next section.

For application contexts, you can use one of the out-of-the-box product, library, project, or program templates, or you can create additional application context templates, as described in the next section.

The details of what is in each out-of-the-box context template can be found in the following chapters:

- Organization templates are described in the [Organizations](#) chapter.
- Product and library templates are described in the [Products and Libraries](#) chapter.
- Project and program templates are described in the [Projects and Programs](#) chapter.

For information on creating context templates, see [Creating Context Templates](#).

**Note:** The role names used in Windchill solution context templates are included as references to the <Windchill>\src\wt\project\RoleRB.rbinf resource bundle so that when they are used, they appear in the language specified by the browser language. This means that the translated templates do not contain translated role names.

## Administering Domains and Policies

This section describes how Windchill domains and policies are defined. A *domain* is an administrative area that defines a set of administrative policies, such as access control, indexing, and notification. Objects associated with a Windchill domain are subject to its policies.

**Note:** Windchill domains are not the same as internet domains.

A *policy* is a collection of rules designed for various types of objects associated with a domain. For example, an indexing policy consists of rules that determine the types of objects for which metadata should be entered into specified collections, when the objects belong to the domain.

Before creating a context in your Windchill solution, you should determine which domains are needed and use a template that creates the domains and the access control policies for those domains when you create the context. For details on what domains and policies are included in the out-of-the-box templates, see the following chapters:

- Organization templates are described in the [Organizations](#) chapter.
- Product and library templates are described in the [Products and Libraries](#) chapter.
- Project and program templates are described in the [Projects and Programs](#) chapter

For the details on creating additional context templates, see the [Context Templates](#) chapter.

After a context is created, you can use the Policy Administrator to administer the domains in the context.

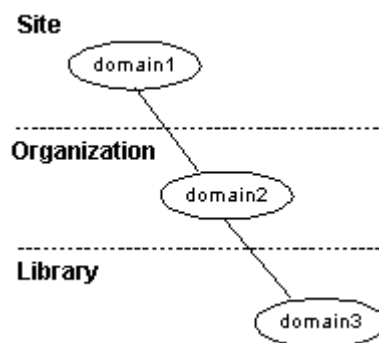
The following sections provide information about installed domains and how to use the Policy Administrator to administer the domains.

## Context and Domain Hierarchy Overview

Context types have the following established hierarchy:



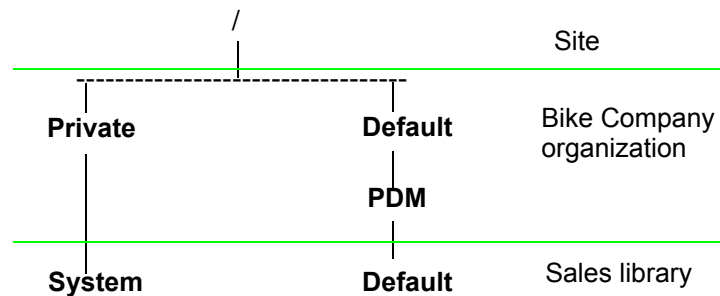
When each context is created, a set of domains is also created for use within the context. Generally, the domain hierarchy is established using the context hierarchy. A parent domain is either in the same context as its child domain or in the parent context of its child domain's context. In the following diagram, the site, organization, and library contexts are shown with one domain defined in each context. The domain hierarchy shows domain3 (in the library context) as a child of domain2 (in the organization context) and domain2 as a child of domain1 (in the site context):



To illustrate this rule, consider the following examples:

- Assume that the Windchill installation creates the site context and has a child organization context that is named Bike Company, and that an administrator (using an out-of-the-box library template) creates a public library context using the following:
  - The team is a local team.
  - The context is named Sales and is a child of the Bike Company context.

Then the domain hierarchy for the Sales library is as follows:



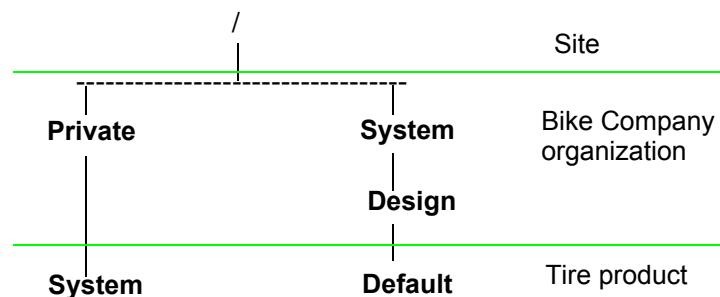
From within the Policy Administrator, the hierarchy is shown as follows:

```

/ (Site)
|-Default (Organization Bike Company)
  |-PDM (Organization Bike Company)
    |-Default (Library Sales)
  |-Private (Organization Bike Company)
    |-System (Library Sales)
  
```

- Assume that the Windchill installation creates the site context and has a child organization context that is named Bike Company, and that an administrator (using an out-of-the-box product template) creates a product context using the following:
  - The team is a shared team named Design.
  - The context is named Tire and is a child of the Bike Company context.

Then the domain hierarchy for the Tire product is as follows:



From within the Policy Administrator, the hierarchy is shown as follows:

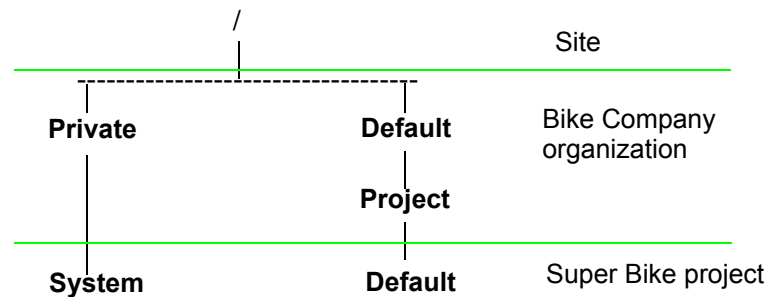
```

/ (Site)
|-System (Organization Bike Company)
  |-Design (Organization Bike Company)
    |-Default (Product Tire)
  |-Private (Organization Bike Company)
    |-System (Product Tire)

```

- Assume that the Windchill installation creates the site context and has a child organization context that is named Bike Company, and that an administrator (using an out-of-the-box project template) creates a public project context using the following:
  - The team is a local team.
  - The context is named Super Bike and is a child of the Bike Company context.

Then the domain hierarchy for the Windchill ProjectLink Super Bike project is as follows:



From within the Policy Administrator, the hierarchy is shown as follows:

```
/ (Site)
|-Default (Organization Bike Company)
  |-Project (Organization Bike Company)
    |-Default (Project Super Bike)
  |-Private (Organization Bike Company)
    |-System (Project Super Bike)
```

## Domains in the Site Context

When a Windchill solution is installed, the following domains are initially defined in the site context:

Domain Names	Description
/ (root )	Sets rules that govern the entire enterprise.
/System	Serves as the default domain for most system administration objects, including queues and life cycle definitions. /System is the default domain of the Administrators group.  The /System domain is not intended for general user business objects.
/User	Serves as the default domain for organization objects and as the parent domain for the /Unaffiliated domain and for organization context domains that hold the users (and their personal cabinets) for that organization.
/Default	Serves as the default domain for the initialization of business information objects; however, this domain is not usually the domain set when an object is persisted.
/SessionIterationDomain	Serves as the default domain for internal session processes. Do not modify the policies set for this domain.
/Unaffiliated	Serves as the default domain for users who are not associated with any organization. These users do not have the organization attribute set in their directory service entry. The /Unaffiliated domain is a child domain of the /User domain.

The root, /System, /User, and /Default domains cannot be moved or deleted. Also the domain names of these domains and of the Unaffiliated domain are configured through the wt.properties file.

The /System, /User, /Default, and /SessionIterationDomain domains belong to the / (root) domain. Additional domains can be defined and associated with the root domain, or domains can be nested by defining them as children of another domain. This allows you to define a hierarchy of domains and then define policies at each level in the hierarchy.

## Creating Domains

Windchill automatically creates domains as follows:

- During installation, as described earlier in this chapter.
- When you create contexts. Each context template defines a set of domains that are created.
- Each time Windchill identifies a user that is affiliated with an organization that does not have an associated domain, Windchill creates the corresponding domain for the organization as a child domain of the User domain.

In addition, administrators can manually create domains.

The name of a domain can be a maximum of 200 characters. When Windchill automatically creates the domains associated with organizations, the domain name is usually the same as the organization name. However, organization names can be up to 200 characters. Therefore to name the domain, Windchill truncates the organization name to 193 characters. This allows Windchill to add a maximum of seven more characters to the domain name to identify a unique name when two or more organizations have the same beginning 193 characters in their names. When the truncated domain name is not unique, Windchill appends [1] to the truncated name. If appending [1] to the name does not make it unique, then Windchill appends [2] instead of [1]. If appending [2] does not make the name unique, then [3] is appended and so on until a unique name is found or until the maximum number of attempts to find a unique name is achieved. The maximum number of attempts is defined in the following property in the wt.properties file:

`wt.inf.container.WTContainerServerHelper.maxDomainCreationAttempts`

The default value for this property is 25.

If the organization name contains a forward slash ( / ), Windchill automatically converts the forward slash to a dash ( - ) in the domain name. For example, if the organization name is "ABC/Main", the domain name created would be "ABC-Main".

**Note:** After Windchill creates a domain that is associated with an organization, an administrator can change the domain name to something more meaningful; Windchill does not rely on the organization domain name matching the organization name.



## Defining Domain-based Policies

Using the Policy Administrator, you can create additional domains and define or change the rules set for each domain. For example, you can define the following:

- An *access control policy*, which determines user, group, and organization permissions to access objects associated with the domain.
- An *indexing policy*, which determines the collections into which metadata for an object is entered when the object is in a specific life cycle state.
- A *notification policy*, which determines who is informed when an event of interest occurs to an object in the domain.

Load files can include sets of domain-based policies for access control and indexing. Templates that are used to create contexts can only include sets of domain-based policies for access control.

**Note:** Policy rules apply to object types. Access to an instance of an object type can be governed by ad hoc access control rules in addition to policy rules. For more information, see the [Access Control](#) chapter.

For objects to be subject to policies, they must reside in a domain. The policies that apply to an object are those policies defined for the domain to which the object belongs, as well as parent domains. To exist within a domain, the object must implement the `wt.admin.DomainAdminister` interface. Not all objects exist within a domain. For example, objects of type `WTPartMaster`, `WTDocumentMaster`, and `WorkItem` do not exist within a domain. For details on the `wt.admin.DomainAdminister` interface, see the associated javadoc.

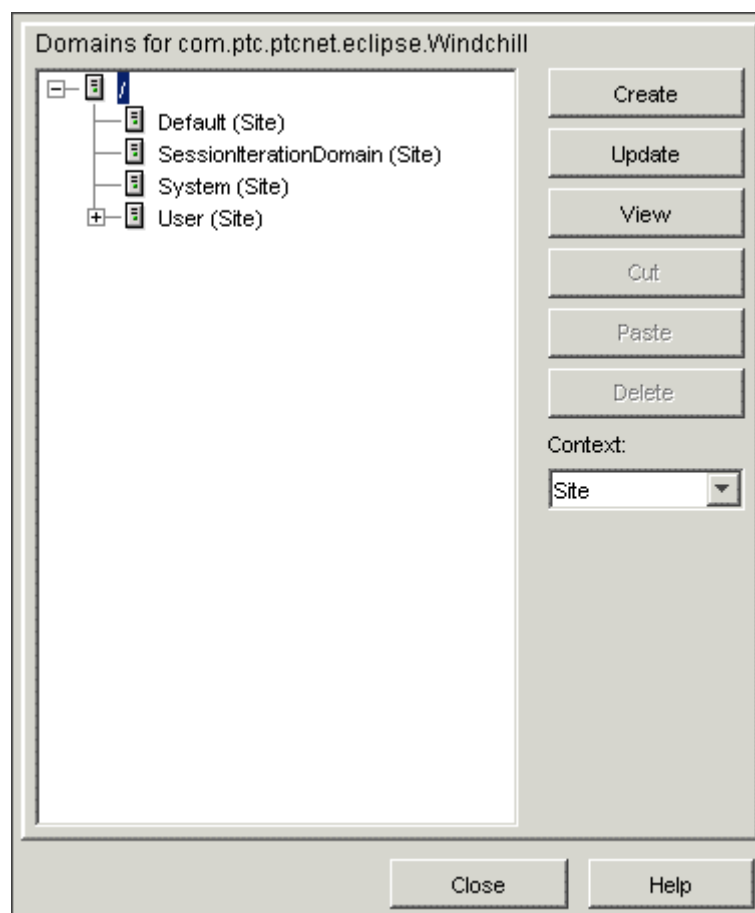
## Using the Policy Administrator

The Policy Administrator operates within the confines of the context from which it is started. You can access the Policy Administrator from the **Utilities** pages that are under the **Site**, **Organization**, **Product**, **Library**, **Project**, and **Program** tabs.

You begin policy administration on the Policy Administrator window. From this window, you have access to the following domains:

- The domains that are in the current context.
- The parent domains of the domains in the current context.

When accessing the **Policy Administrator** from the **Site** tab **Utilities** page, the initial window is similar to the following:



The domains display in a tree view which is open to the domains at the top of the domain hierarchy within the current context. The current context appears selected in the **Context** drop-down list.

The format of each domain in the list includes the domain name followed by the following information in parentheses:

- The type of context (for organization and application contexts)
- The name of the context where the domain resides

For example, if a domain in the list is shown as **Default (Library Common Parts)**, then the domain name is Default and the domain resides in the library context named Common Parts. If the Default domain is in the Site context, then you will see **Default (Site)**. In this case, the context type is not displayed.

You can select any of the domains listed in the tree, or you can expand the tree to show child domains within the current context and then select a domain. You can also change the contents of the domain tree by using the **Context** drop-down list, as follows:

- Selecting a parent context from the **Context** drop-down list updates the domain tree so that the domains in the selected context and direct parent domains are available.

The contexts listed in the **Context** drop-down list are formatted as the context type followed by the context name.

- Selecting **All Contexts** from the **Context** drop-down list updates the domain tree so that all domains from all contexts are available.

**Note:** The buttons that are enabled when you select a domain are determined by the Policy Administrator. Buttons are disabled if the action cannot be performed on the domain based on current selections or if the action can never be performed on the domain; buttons are not disabled based on user permissions. For example, if you select a domain that can never be deleted (such as the Default (Site) domain), the **Delete** button is disabled. If a button is enabled and a user who does not have permission to execute the action clicks the button, the user receives an error.

## Specifying Policy Rules in a Context Template

For Windchill PDMLink, the recommended method for including a specific set of policy rules in a context template is to set the rules in an existing container using the Policy Administrator. Then, either save the current context as a template or export the context to a ZIP file on your system. Be sure to select the **Access Policy Rules** option when you save or export an existing template. Using this method provides access control policy rules for domains within a context and allows only user-defined groups and organizations as principals. Also the principal names must be unique within the container search scope. For information about the search scope, see [Searching for Principals in Administrative Clients](#).

After you have a ZIP file containing the access control policy rules (along with any other administrative items you want in the template), import the template using the ZIP file as input.

For Windchill ProjectLink, you cannot save the policy rules set in the current context when creating a template from the current context or when exporting the context.

For the details on creating context templates, see the [Context Templates](#) chapter.

## Assigning Domains to Folders in Solutions with Products and Libraries

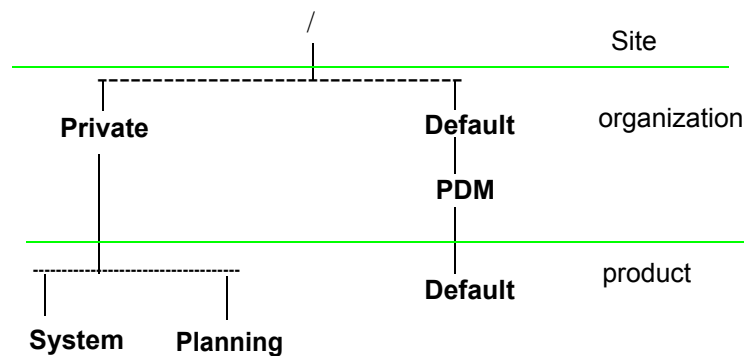
When creating folders, the default behavior is that the domain associated with the new folder is the same domain that is associated with the parent cabinet or folder.

The ability to create folders is controlled through the access control policy rules that are set for folders. Out-of-the-box, only product or library managers and organization administrators (in Windchill solutions with products and libraries) and the site administrators (in Windchill solutions with products and libraries) can create folders (or update the domains in existing folders). To allow others to create folders, you can add access control policy rules that grant Create and Modify permission for the SubFolder object type to those principals who you want to have this capability.

To allow those who can create folders the ability to associate a domain other than the domain that is associated with the parent folder, you must set the **Display Folder Domains** preference.

Setting this preference to true reveals domain information on the dialogs used when creating or updating folders and on folder information pages. Providing the domain information allows users who have the appropriate access control rights to assign different domains to folders and to see which domain is currently assigned to a folder. In Windchill solutions with products and libraries, the interface allows a user to select a domain from the domains established in the container other than the /System domain. You can set this preference at any context level through the Preference Manager from the product, library, organization, and site **Utilities** pages.

You can use the ability to create folders that are associated with a domain other than the domain that is associated with the parent folder in conjunction with the ability to create top-level domains in an application context. For example, assume you have created the /Planning domain in following domain structure for a product:



The access control rules in the /Planning domain only inherit the rules from the Site / (root) domain and the organization /Private domain and not the rules from the organization /Default and /Default/PDM domains. When creating folders in a product, the domain associated, by default, is the /Default domain in the product. By allowing the user to select a different domain, the user could select the /Planning domain instead of the /Default domain. Creating a folder associated with the /Planning domain provides a different access policy for objects in the folder since the access control rules for objects associated with the /Planning domain can be different than the rules associated with the product /Default domain and a different set of rules are inherited from the organization.

## Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects

Numbering schemes control how numbers are generated for parts, documents, CAD documents, and change objects. Each object class has its own numbering scheme.

For all Windchill solutions, a rules load file sets the initial numbering rules for documents, CAD documents, parts, and change objects. These rules affect the following:

- Documents, CAD documents, parts, and change objects created from within a product, library, organization, or site context (if Windchill PDMLink is installed)

**Note:** Autonumbering must be turned on for the Dynamic Document soft type and its subtypes. See [Configuring Products or Libraries for Dynamic Documents](#).

- Documents, CAD documents, and parts created from within a project, organization, or site context (if Windchill ProjectLink is installed)

### Out-of-the-box Numbering Schemes

The loaded numbering schemes for parts, documents, CAD documents, and change objects are simple Oracle sequences that have been loaded into the Windchill database. Each starts at 1 and increments by 1. This means that the numbers for all parts, documents, CAD documents, and change objects are autogenerated according to the numbering scheme that is in place.

## Changing Numbering Schemes

To change an out-of-the-box numbering scheme, you can change to a scheme that is defined for you or you can define your own scheme. You can view the rules that are being used for numbering from the Object Initialization Rules table using the Object Initialization Rules Administrator.

Object initialization rules can be set in context templates or by using the Object Initialization Rule Administrator from within a specific context.

For details on how to change a numbering scheme, see [Changing Numbering Schemes](#).

For the details on creating context templates, see the [Context Templates](#) chapter.

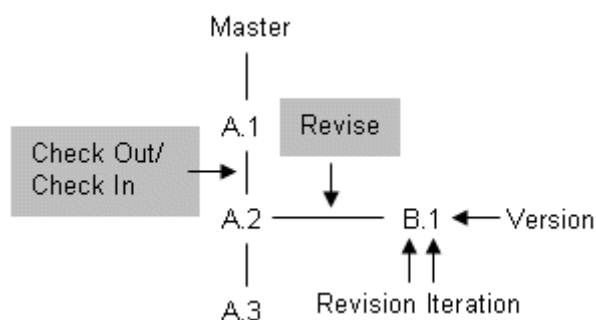
For details on assessing the Object Initialization Rules Administrator, see the [Object Initialization Rules](#) chapter.

## Administering the Versioning of Parts, Documents, and CAD Documents

Versioning schemes define the labels, or identifiers, that are automatically applied as object versions are created in a Windchill solution.

### Understanding the Use of Versioning Schemes

The following diagram and the sections that follow provide some context on how versioning schemes are used:



### Master

When users create a document or part, your Windchill solution creates a master as well as an initial version, which is identified in the previous diagram as revision A, iteration 1 (A.1). The master consists of common attributes for all versions, such as part number or document name.

## Version

The version of a part or document is identified by the combination of revision and iteration.

## Revision

When users select the **New Revision** action, your Windchill solution creates a new version as the start of a new revision of the part or document. For example, if the previous version was A.2, the new revised version would be B.1.

## Iteration

When users select the **Check Out** action, modify the part or document, and select the **Check In** action, your Windchill solution creates a new version of the part or document as the next iteration of the current revision upon check in. For example, if the previous version was A.2, the new iteration would be version A.3.

## Initial Versioning Rules

For all Windchill solutions, load files set the initial versioning rules for parts, documents, and CAD documents in the site context. These rules affect the parts, documents, and CAD documents created from within a product or library.

The default versioning scheme used is set when site context rules are established unless you choose to set them in an organization, product, or library context. The out-of-the-box demonstration template named Product Design sets versioning for a product context. For details on the Product Design template, see [Out-of-the-box Product and Library Context Templates](#) in the Products and Libraries chapter.

**Note:** PTC recommends that you establish one versioning scheme for all of your contexts and that you set the versioning scheme in the site context. However, you can override the default versioning scheme for a specific organization, product, or library context.

Documents, CAD documents, and parts that are created from within a project or program context are not versioned objects.

## Changing Versioning Schemes

To change an out-of-the-box versioning scheme, you can change to a scheme that is defined for you or you can define your own scheme. You specify the versioning scheme in an object initialization rule for a particular object type, such as wt.doc.WTDocument. You can view the rules that are being used for versioning from the Object Initialization Rules table using the Object Initialization Rules Administrator.

Object initialization rules can be set in context templates or by using the Object Initialization Rule Administrator from within a specific context.

For details on how to change a versioning scheme, see [Changing Versioning Schemes](#) in the Object Initialization Rules chapter.

For the details on creating context templates, see the [Context Templates](#) chapter.

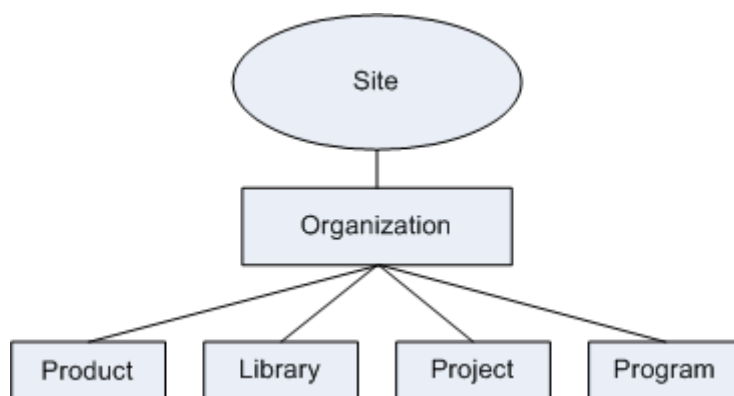
For details on accessing the Object Initialization Rules Administrator, see the [Object Initialization Rules](#) chapter.

## Administering Preferences

The Preference Manager allows administrators for the site, organizations and other application contexts (products, libraries, projects, and programs) to set preferences for the contexts that they administer.

### Preference Hierarchy

Preference values set at one level are inherited by the application contexts below. For example, preference values set at the site level are inherited by the organization; application contexts within the organization inherit preference values from the organization.



**Note:** User preferences may be inherited directly from the organization-level, from individual application contexts, or may be unique (able to be set only at the user level).

Preference values can be set for each application context, or for an individual user, unless the preference value is locked at a higher level. For example, if a preference is not locked at the organization level, then it may have one value at the organization level, but individual products or projects within that organization could each specify a different value. Similarly, if a preference is not locked at an organization level, but is available at the user level, individual users could each set different values for that preference.



Preferences can be locked at any level above the individual user. If a preference is locked, then the value cannot be changed for those lower levels. A site or organization administrator can choose to delete all values for a particular preference that may be set at levels below the current context, usually prior to locking the preference.

If a preference is not explicitly set for a particular application context, then the value for that preference is inherited from the level above. For example:

- If a preference is not set for a particular product or project, then the value set for that preference at the organization level is used.
- If the preference is not set at the organization level, then the value set for that preference at the site level is used.
- If no preference is set at any level, then the default value defined for that preference is used.

## Preference Manager



To access the preferences for a particular application context, click the **Preference Manager** link on the **Utilities** page of the application context tab. The preferences displayed are determined by the context from which the Preference Manager is launched. (Individual users can set their personal user preferences through the **Preference Manager** link on the **Utilities** page of the **Home** tab. The Preferences link from the search pages also launches the Preference Manager at the user level.)













When the Preference Manager is launched from a particular application context, the complete list of preferences for that context is displayed. For example, if the Preference Manager is launched from the **Utilities** page for an organization, all preferences for that organization are displayed. The preferences are grouped by category and subcategory according to the type of functionality they impact. Within each category, the preferences are named, described, and the current preference value is displayed. Additional information about each preference, such as a longer description and default value, is displayed on the **Set Preference** window.

Context:

Search within preferences:

Preference Manager Current View:



Name		Value	Description
[-] Attachments			
Default File Path	 		Specifies a default directory for uploading and do... <a href="#">More</a>
File Download Behavior	 	Always ask whether to ope...	Determines whether downloaded files are opened or ... <a href="#">More</a>
File Download Mechanism	 	Use basic browser functio...	Determines how to download files from Windchill.
File Upload Mechanism	 	Use Java applet functiona...	Determines how to upload files to Windchill.
[-] Attribute Handling			
Local Time Zone	 		The local time-zone of the user. The date and time... <a href="#">More</a>
Measurement system	 		Measurement system to use when displaying attribut <a href="#">More</a>

**Note:** If you want to view the internal name of the preferences, you can customize the table view to display the **Internal Name** column. For more information, see the help available from the **New View** window.

For more information, see the help available from the Preference Manager.

## Administering Saved Searches

When saving searches, you can select the **Mark as Global Search** check box to make the saved search available to all members of a particular context. Select the context from the **Set Scope of Global Search** field; this field will be populated with the contexts which you created or for which you are an administrator. Site administrators can also delete saved searches created by other users.

Context administrators (site, organization, product, library, project, and program) can save searches, and assign them to groups of users. You can also select the **Mark as Global Search** check box to make the saved search available to all members of a particular context. Select the context from the **Set Scope of Global Search** field; this field will be populated with the contexts for which you are an administrator. Administrators can also delete saved searches created by other users.

For more information, see the help available from the search pages.

# 4

## Site

This chapter provides an overview for administering the site and describes the typical duties that a site administrator performs. It also provides additional information about some of the main administrative tasks for sites.

Topic	Page
Overview .....	4-2
Typical Duties of Site Administrators.....	4-2
Business Object Uniqueness Considerations .....	4-11
Out-of-the-Box Site Configuration .....	4-15
About the Site Tab.....	4-15
Best Practices .....	4-16

## Overview

The site administrator manages the organizations within the site and is responsible for the common information and rules inherited by all contexts within the site.

The site administrator (for example, wadmin) created at installation time may be a temporary administrator specified only to get the system up and running for the customer. If that is true, create a user who is associated with the hosting organization and add that person to the Administrators group. That person will then take over as the site administrator once the initial installation and setup is completed.

For general information about context contents and how to create contexts, see [Contexts](#).

The general information in this chapter pertains to all Windchill solutions. The site administrator performs most of these actions from the **Site** tab of the Windchill solution user interface.

## Typical Duties of Site Administrators

Site administrators are responsible for the configuration and management of the Windchill system as a whole. They create organizations representing business units of a hosting company and organizations representing partners and suppliers. Site administrators control how authorized users are added to the system. They also define the information that is common across all organizations and their products, libraries, projects, and programs.

Typical duties of the site administrator include the following:

- Create and update organizations participating in the site.
- Determine if new organizations are subscribers (able to host product, library, project, or program contexts).
- Manage a group of users with site administration privileges.
- Manage site-level folders, documents, and links.
- Manage site-level access policies and rules.
- Manage site-level types and type-specific attributes that are inherited by all contexts.
- Manage the rules governing object creation.
- Manage site-level templates that are inherited by child contexts.
- Manage site-level object initialization rules that are inherited by child contexts.
- Create and manage security auditing reports for audit records of who did what causing changes in the Windchill database.

- Manage site configuration (such as vaulting, replication, calendar, property options, and preferences).
- Create profiles to define which parts of the user interface (e.g., actions, tabs, and attributes) should be visible to users, groups, or organization members. Site administrators determine whether application context managers can override profiles that are created at the site context.
- Manage processes (such as workflow, life cycles, CAD viewable publishing, and replication).
- Manage security access for workflow expressions.
- Export and import site-level information.
- Define and manage reports.
- Determine searchable object types.
- Manage saved searches.
- View and manage access control rules for objects.

## Creating and Managing Organizations

When a Windchill solution is created, an organization principal is created and associated with the site context. The default data loading process creates an initial organization context, also associated with the organization principal. Both the site and this initial organization have the same name. If the data loading process was modified so that the organization context is not created, you must create an organization context before users access the solution.

Users who have an organization attribute (the "o" attribute, by default) on their directory service entry that matches the organization principal's name automatically become members of that organization. If your site does not use the organization attribute in the directory service entry, users can be assigned to an organization using the `usersOrganizationName` property. For more information on using this property, see adding enterprise directories in the *Windchill Installation and Configuration Guide - Advanced*.

Organization contexts are created by site administrators from the **Organizations** page of the **Organization** tab. For more information, see [Creating an Organization Context](#).

**Note:** Site administrator users, such as `wcadmin`, should not be used to create contexts (products, libraries, projects, or programs) within organization contexts, nor should they create objects within those contexts.

The site administrator can review the list of organizations in the site and navigate to update each of the organizations from the **Organizations** page on the **Organization** tab. (The **Organization** page is visible only to site administrators.) A company will probably choose to administer the organizations representing their partners or customers. In this way, you become the organization administrator on behalf of any number of organizations defined in the site.

For more information on organizations, see the [Organizations](#) chapter.

## Adding and Editing Members

You control how users are added to the site to enable them to log on and view and author information in the site. Users are added and edited using the Principal Administrator utility. You can also define the users who have site administration privileges so that several individuals in the company can perform this duty.

For more information about using the Principal Administrator utility, see [Using the Principal Administrator](#) in the Principals (Users, Groups, and Organizations) chapter.

## Creating and Managing Site Folders and Documents

You can define documents, folders, and links within the site context. The site folders are intended to hold any documents that are important for administering the site.

Types of documents that administrators might define at the site include the following:

- Site configuration documentation.
- Site change log that captures a record of changes made to the site.
- Operation rules and procedures (such as shutdown, backup, and restart procedures).
- Site administrator responsibilities document.
- Key contact list for site administrators.
- Deployment schedule and plans (this might also be defined in a project but referenced by providing a link in the site folder).
- Documents describing site-level document, life cycle, and workflow templates.

## Changing Default Configuration Options

You can use properties and preferences to change the configuration options that are set by default. Preferences are accessed using the Preference Manager on the **Utility** page for a tab. Properties are contained in properties files in your codebase.

## Properties

You can find the details for properties that you may want to change for performance reasons in the `properties.html` file located in your codebase.

For example, you can change the folder size used when displaying content in a Windchill solution through the following property:

`wt.folder.ResultsLimit`

- Limits the number of objects that can be displayed in the **Folders** table. If the limit is exceeded, a message informs the user to use search or go to a folder details page to access the object they are looking for. A maximum number of objects that can be displayed is set so that the server does not become unresponsive when a user tries to retrieve a large product, library, project, or program. When the limit is reached, the user should pick a subfolder to reduce the size of the page.
- Default: 15000 objects

**Note:** When changing properties, always use the `xconfmanager` utility so that your site changes are maintained over maintenance releases.

## Preferences

The Preference Manager helps you maintain configuration preferences as well as many other types of preferences. The preference values you set at the site level affect all application contexts and users within the site. Access the site level Preference Manager by clicking the **Preference Manager** link on the **Site** tab **Utilities** page.

For more information on the Preference Manager, see [Administering Preferences](#).

## Managing Site-level Types and Type-specific Attributes

Using the Type and Attribute Manager utility from the **Site** tab **Utilities** page you can define object types and type-specific attributes to make available to all organizations in the system. For example, a company might define a change impact report document type with specific attributes for several categories, including replacement cost, production tooling cost, and so forth.

You can define types and attributes within a context of an organization. When soft types and soft attributes are created within an organization context, they are kept separate from all other organizations and remain private to the organization they are created in. For more information about types and attributes, see the [Type and Attribute Manager](#) chapter.

You can associate document types with life cycles which identify the various states of maturity of the document. For more information about life cycles, see the [Life Cycles](#) chapter.

## Configuring Numbering and Versioning Schemes and Units of Measure

You can configure the number and version scheme used to uniquely identify parts, documents, and other items in the system. The site-level schemes and units of measure are inherited by the organizations, but each organization can optionally define its own schemes or override the site-level schemes. In general, when a company is hosting Windchill for its use, the numbering, versioning, and units of measure should all be defined at the site context and should not be overridden by individual organizations. This approach ensures consistency of basic identification and units across the company.

For additional information on numbering [Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects](#) in the [Contexts](#) chapter

For additional information on versioning [Administering the Versioning of Parts, Documents, and CAD Documents](#) in the [Contexts](#) chapter.

For more information on units of measure, see the online help for the Type and Attribute Manager that is accessible on the **Utilities** page of the **Site** tab.

## Managing Site-level Templates

A company may choose to define a number of document, life cycle, and workflow templates that they want all their business units and partners to use. For example, the site may want to define a document template and associated life cycle and workflow process for capturing enhancement requests for the Windchill system. Or a company may want to make a corporate presentation template available to all its business units.

The site can define product, library, project, program, and organization templates. Organization templates are required to create organizations. For more information, see [Creating Context Templates](#).

You can restrict which templates are made available to the applications within the site by choosing to hide certain templates at the site level. For more information, see the help from the **Templates** table on the **Site** tab.

## Managing Site-level Object Initialization Rules

You can configure how attributes (such as number) are displayed and generated using the Object Initialization Rules administrator. This utility defines how the initial attribute values for the object are established, how the attributes are displayed, and can determine basic relationships, such as life cycle association, when an object is created. The rules established by the parent site are inherited by each organization by default, but they can be overridden at the organization level. The rules are defined in an XML format, which the site administrator can view and edit. For more information, see [Defining the Content of XML Documents used for Object Initialization Rules](#).



## Managing Workflow Security

Workflow authors are permitted to write workflow-embedded Java code to facilitate the execution of the workflow process. This embedded Java code is executed on the server, and there are no restrictions on the APIs available for use.

Considering this capability, a user with permissions to create workflow templates (for example, a project or program manager) could add malicious code in one of the workflow expressions, causing a possible security threat. For this reason, workflow templates that contain Java expressions must be written, reviewed, and thoroughly tested by individuals that are trusted by the organization. The site administrator has an additional level of control to prevent a user who is not a member of at least one of three specific site context groups (Administrators, Workflow Administrators, or Workflow Authors) from embedding Java code in workflows.

For more information about workflow security, see the [Workflow](#) chapter.

## Auditing System Information

The auditing framework provides a mechanism to enable event-based logging that provides a historical record of who did what that caused changes in the Windchill database. The security auditing feature also provides a mechanism to retain historical records of security related events, such as privilege changes, team changes, and denial of access. Security audit reporting is helpful to customers in highly regulated industries that need to identify events that may have resulted in a potential security breach.

The only events that are enabled out-of-the-box are **License Usage Reporting** and **Organization Usage**. Any other specific events that you want recorded in the audit log must be enabled on an individual basis. For more information, see the [Auditing](#) chapter.

## Creating and Managing Access Control Policies

Using the Policy Administrator, you can define policies that control the level of access to information in the system. For example, as site administrator, you may want to give an engineering group read access to all documents of type Engineering Specification. You need to first define an Engineering Group and populate it with the appropriate members, then define a document type of Engineering Specification at the site level, and then use the Policy Administrator at the site level to define the access policy based on the document type, the group or groups provided access and the access level.

You should create only those site-level policies that provide broad access for types of information defined at the site level for all organizations in the system.

For more information about access control, see the [Access Control](#) chapter.

## Creating and Managing Profiles

You can dynamically control which actions are visible to a user, group of users, or users in an organization by associating that information with a profile. Profiles are not an access control mechanism; they are a user interface control mechanism. A profile represents a typical category of user within a company and is based on the roles and privileges associated with that particular user category.

Profiles expose only the necessary functionality and information needed by a user. This creates a focused and simplified user interface, reducing confusion by eliminating areas of the user interface that could be distracting.

For more information about profiles, see the [Profile Management](#) section in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

## Configuring External Vaults or Replication Sites to Optimize Performance

You can configure external file vaults so that document and part content is stored on a file system rather than in the database. This type of configuration can provide significant upload performance improvements and is appropriate when the site is frequently used to exchange large files (such as CAD model files).

By default, external storage rules are based on individual domains in each context. You can launch the External Storage Administrator client from the **Utilities** page in the context of a product, a library, or the site. (The client is not available from the context of a project, program, or organization.) When launched from a product or library context, product and library managers can set only the vaulting rules. The additional functionality within the External Storage Administrator is available only to site administrators. The client allows you to create vaulting rules for the System and Default domains associated with the context in which the client is launched.

**Note:** The rules needed for setting up an external file vault for a product or library cannot be inherited from the site.

If the increased number of vaults and file vault rules becomes unmanageable, you can force vaulting to be accomplished through a single vault by setting the `wt.fv.forceContentToVault` property to true. For how to set external file vault rules or set up a single vault, see Administering External File Vaults in the *Windchill System Administrator's Guide*.

You can also configure replication sites so the document and part content files are replicated at a remote site where local users have only very low bandwidth connections to the Windchill server. You can launch the File Server Administrator client from the **Utilities** page in the context of a product, a library, or the site. For a project or program, you can launch the Replication Rules Administrator from the project or program **Utilities** page.

**Note:** The File Server Administrator is not available from the context of an organization.

As a site administrator, you would typically define only the replica site and the replication schedule. The product, library, project, or program managers would configure the replication rules for a particular context. The additional functionality within the File Server Administrator is available only to site administrators. For more information on content replication and how to set replication rules, see the *Windchill System Administrator's Guide*.

## Configuring and Managing CAD Publishing Utilities

You can configure CAD workers that publish viewables for CAD models that can be accessed by participants that do not have native CAD authoring tools. The site includes utilities to configure the CAD workers and monitor and manage the publishing schedules and queues.

For additional information, see the [Visualization Services](#) chapter.

## Creating, Updating, and Managing Reports

You can create and update reports against the objects and attributes in the system as a whole. For more information about reporting functionality, see the [Reporting](#) chapter.

## Importing and Exporting Information Among Systems

You can exchange information between a staging server and production server, between servers, or between a server and a file system using the Windchill import/export utilities. These utilities read and write system information in an XML format. For information on exporting and importing XML template files, see [Creating the Contexts from which Users Work](#).

The workflow and life cycle administration utilities accessible to the site and organization administrator integrate the import and export functions with administering workflows and life cycles. For life cycle import and export information, see [Importing and Exporting Life Cycle Templates](#) in the Administering Life Cycles chapter. For workflow import and export information, see [Importing and Exporting Workflow Templates](#) in the Workflow chapter.

## Managing Calendar Settings

The Calendar Management utility, available from the **Site** tab **Utilities** page, allows you to view and change calendar schedules, and delegate work to others.

As the site administrator you can set up the calendar environment for the system to include non-working days such as company holiday. For more information, see the online help available from the Calendar Management utility.

## Setting the Property File for Delegating Tasks

Prior to using the delegation feature, set the following property to false in the wt.properties file using the xconfmanager utility:

```
wt.calendar.calculateDefaults
```

Setting the property to false ensures that the system checks the site and user calendars for any defined non-working days, such as company holidays, and takes these days into account when calculating task due dates and plan activity dates.

For information on using the xconfmanager utility, see the [Administration Overview](#) chapter.

## Monitoring Enterprise Systems Transactions Log

If you configured Windchill to exchange information with an Enterprise Resource Planning (ERP) system, you can monitor the transactions with the ERP system through the Enterprise Systems Transactions Log.

For more information, see the *Windchill Enterprise Systems Integration Administrator's Guide*.

## Purge, Archive, and Restore Jobs

You can create and manage purge jobs with the Purge Manager, accessed from the **Site** tab **Utilities** page. You can create purge jobs to permanently remove data from the Windchill system, or (if the archive functionality is installed at your site) you can create an archive of the data to restore at a later time. For more information see the help available from the Purge Manager.

## Managing Searches

You determine the searchable object types for the site. For more information, see the Adding a New Type to the Search User Interface section in the *Windchill Customizer's Guide*.

You can also determine which object attributes are searched upon in a basic, keyword type search. Out-of-the-box, the default is to search name and number however, you can determine other fields to be searchable.

You can search for and delete searches created by other users. This can be useful if a user or other administrator has created saved searches, but has since left the company.

To delete saved searches created by another user, click **Administrative Delete** on the **Saved Searches** table to open the **Find Searches For Delete** window. Specify your search criteria and click **Search**. You can use wildcards to broaden your search. Select the searches to be deleted from the search results, and click **Delete**. For more information, see the help available from the **Saved Searches** table.

You can determine whether a search from the **Search** page includes only the latest version of an object, rather than all versions. To search for only the latest version of an object, set the Latest Version Search preference to Yes using the Preference Manager on the **Site** tab **Utilities** page:

For more information on searching and finding information in Windchill, see the user's guide for your installed Windchill solution.

## Viewing and Managing Access Control Rules for Objects

You can view and manage the access control rules for objects (such as folders, parts, and documents) at the site level. Navigate to the object, and select the **Manage Security** action. From the **Manage Security** window you can view and set the access control for the object.

For more information, see the [Access Control](#) chapter and the online help available from the **Manage Security** window.

## Managing Overall System Configuration

There are many system configuration settings you can view, set, and update remotely using the system configuration utilities. Use these utilities to view and set system properties, view and manage queues, and view server status and logs.

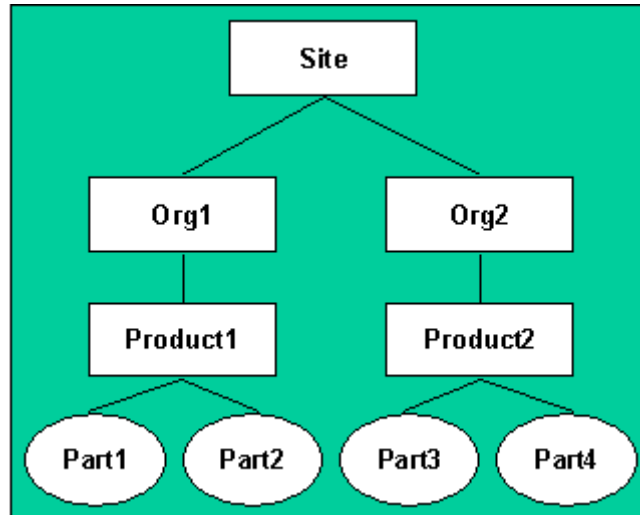
For more information, see the *Windchill System Administrator's Guide*.

## Business Object Uniqueness Considerations

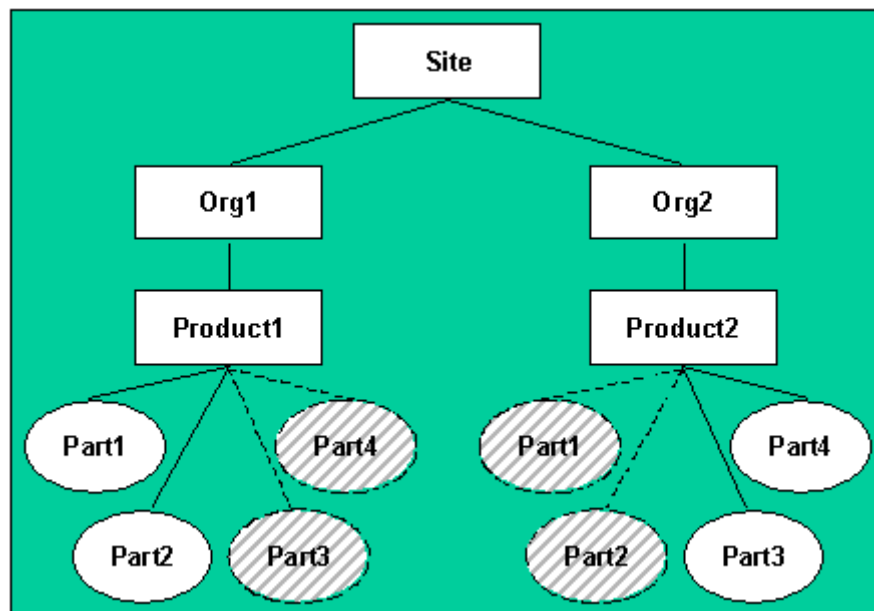
When your Windchill solution was installed and configured, choices were made regarding the uniqueness of business objects, such as parts and documents, when you have multiple organization contexts within your site: whether business objects must be unique across all organizations, or whether they can be unique within each organization.

**Note:** The business object uniqueness scope is set within the organization template. Once an organization context is created, the business object uniqueness scope cannot be changed for that organization.

In a default configuration with multiple organizations, business objects such as parts and documents must be unique across all organizations within the site, as shown in the diagram below. The colored area indicates the space within which each business object must be unique, also called the *business namespace*.

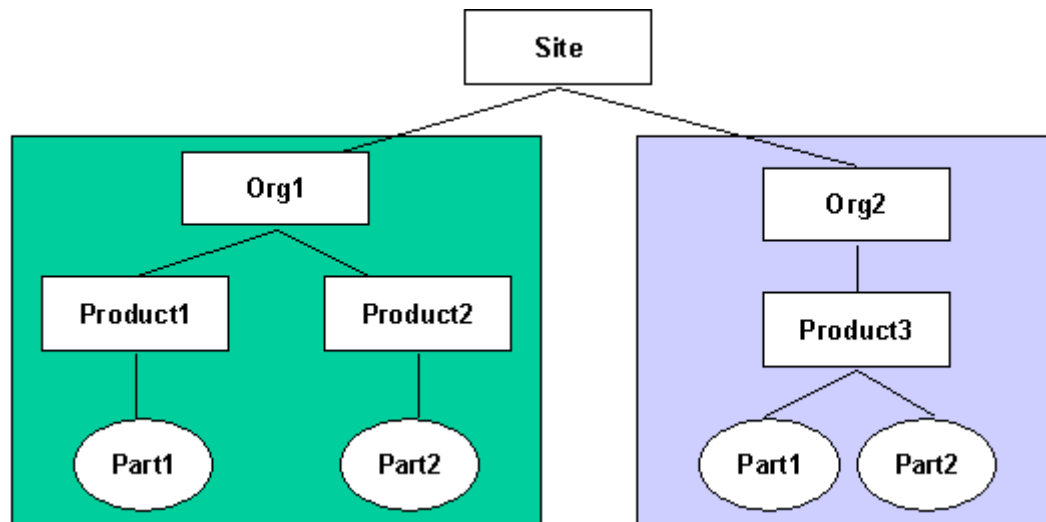


Since each business object must be unique across all organizations within the site, there cannot be a Part1 or Part2 created in Org2, because Part1 and Part2 already exist within Org1; similarly, since Part3 and Part4 already exist in Org2, a user could not create a Part3 or a Part4 in Org1, as shown in the diagram below.

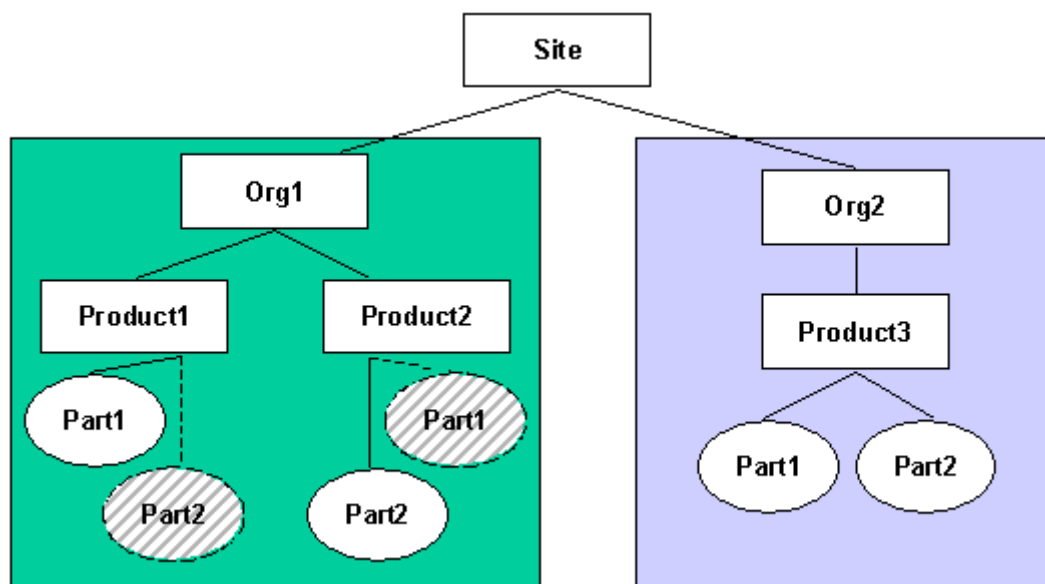


**Note:** CAD document filenames must be unique across all organizations within the site, regardless of the business object uniqueness configuration. For example, if CADDoc1 exists within Org1, there can not be a CADDoc1 in any other organization within your site.

If your system was configured to allow business objects uniqueness by organization, however, the organization becomes part of the uniqueness constraint for each business object. As a result, the uniqueness of a business object is constrained within each organization, rather than within the entire site. This allows for a Part1 to be created in Org2, regardless of whether a Part1 exists in Org1. In the following diagram, the colored areas again indicate the space within which business objects must be unique.



A part named Part1 can exist within both Org1 and Org2 because it is unique within each organization. Within Org1, however, a Part1 cannot be created in Product2 because there is already a Part1 in Product1, as shown in the diagram below.



Regardless of the configuration, users can search for all objects across all organizations in the site, limited only by their access controls and other site configurations. If your site is configured for business object uniqueness within each organization, however, business objects from one organization cannot be selected from the search results and used within a different organization.

**Note:** Regardless of whether business object uniqueness is configured for the site or organization level, the uniqueness of objects within projects and programs is not affected. Objects created within a project or program context are always unique within that project or program.

If you do not know whether your site was configured with business object uniqueness by organization, do one of the following:

- Check with the person who installed your Windchill system.
- See the *Windchill Installation and Configuration Guide - Advanced* for the business object uniqueness configuration details, and determine if they have been implemented.



## Out-of-the-Box Site Configuration

When your Windchill solution is installed, the following are defined for the site:

- Context structure; for more information, see [Installed Site Context Structure](#).
- Context participation; for more information, see [Installed Site Context Participation](#).
- Context policies; for more information, see [Installed Site Context Policies](#).
- Context data; for more information, see [Installed Site Context Data Types and Attributes](#).
- Context templates; for more information, see [Installed Site Templates](#).
- Context rules; for more information, see [Installed Site Object Initialization Rules](#).

## About the Site Tab

The **Site** tab provides access to the following pages:

Page	Description
Folders	Allows you to view information about and add information to folders. This is where the site administrator can store information needed for administering the site.
Administrators	Allows you to view, add, or remove users from the site administrators group.
Profiles	Allows you to view and manage profiles for your site. For more information, see <a href="#">Profile Management</a> .
Types	Allows you to create and update document types. This page also provides access to the Life Cycle Administrator, Attribute Manager, and Type Manager.
Templates	Allows you to view, create, or edit templates.
Reports	Allows you to view and edit reports at the site level.
Utilities	Allows access to utilities available to your site

## Best Practices

The following best practices are recommended by PTC.

### For Windchill Solutions with Products and Libraries

The following best practices are recommended for Windchill solutions which use products and libraries.

#### Setting Object Initialization Rules

The object initialization rules include the ability to set default values and display characteristics for the following attributes:

- Folder paths
- Default life cycle template
- Default team template
- Default numbering scheme
- Default version scheme

Additionally, rules can be used to set the following:

- Available life cycle templates
- Attribute values
- Display characteristics for other attributes

Although these rules can be set for individual product and library contexts, PTC recommends that you set them at either the organization or site level so that an entire organization or site is using the same set of rules. This promotes reuse and reduces unnecessary duplication at lower context levels.

For more information, see the [Object Initialization Rules](#) chapter.

#### Setting Up Enhanced Life Cycle Templates

The out-of-the-box Product Design template includes object initialization rules that set different default life cycle templates and versioning schemes for parts, documents, and CAD documents than what is set out-of-the-box at the site level. This template demonstrates the use of simplified life cycles and state-based versioning. Unless you have overriding business practices that require unique rules, PTC recommends that if you want to use these features, you modify the site object initialization rules and remove them from the Product Design template.

## Overriding and Reassigning Life Cycle and Team Templates

The life cycle and team templates for objects such as parts and documents can be manually selected during object creation. For existing objects, life cycles can be reassigned and team templates can be reset.

For more information, including the preferences used to enable this functionality, see [Overriding and Reassigning Life Cycle and Team Templates](#).



# 5

## Organizations

This chapter provides an overview for administering organizations and describes the typical duties that an organization administrator performs. It also provides additional information about some of the main administrative tasks for organizations.

Topic	Page
Overview .....	5-2
Typical Duties of Organization Administrators .....	5-3
Out-of-the-box Organization Templates .....	5-10
Creating an Organization Context .....	5-17
Owning Organization Principals .....	5-19
Using the Organization Utilities Page .....	5-22
Changing an Established Internet Domain .....	5-22
Using the OrganizationSync Utility for User Organization Changes .....	5-23
Best Practices .....	5-25

# Overview

Organization administrators are responsible for the configuration and management of an organization within the Windchill system. The organization may represent a business unit of the parent company hosting the Windchill system or it may represent a supplier or partner to the parent company. In an exchange environment, an organization represents a company paying for the ability to create projects.

There are two types of organizations, an organization principal (WTOrganization type) and an organization context.

- An organization principal (made up of an object of type WTOrganization and a directory service) represents a group of users. Each organization principal can be associated with and manage an organization context that allows creation of products, libraries, projects, and programs within that organization.

Not every organization principal should have a corresponding organization context. Only create an organization context if the organization principal has a need to manage its own products, libraries, projects and programs.

**Note:** Within the user interface, for example within the Policy Administrator, you may see the current organization principal represented by "This Org", rather than the organization principal's name.

For more information on organization principals, see [Principals \(Users, Groups, and Organizations\)](#).

- The organization context is associated with an organization principal, and provides the framework within other application contexts (products, libraries, projects, and programs) can be created, and organization-level administrative actions can be carried out, as discussed in this chapter.

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

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

All Windchill solutions, when configured, contain an initial organization context. This organization represents your enterprise and is associated with the organization principal created during installation.

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

**Note:** PTC recommends that if you are considering using multiple organization contexts in the future, you should create at least one additional organization context under which your products, libraries, projects, or programs are created, rather than creating them within the initial organization context. This allows you to add additional organizations (principals and contexts) without having to restructure your administrative data (such as soft types) so that members of one organization cannot see data from another organization.

Windchill solutions provide client user interfaces for doing most activities that are related to administering organizations. Organization administrators define the information that is common across all products, libraries, projects, and programs created within their organization.

This chapter contains information that an organization administrator needs to know, as well as information that a site administrator needs in order to make the organization functional.

## Typical Duties of Organization Administrators

Responsibilities of the organization administrator include the following:

- Managing organization members, groups, roles and shared teams
- Creating, updating, and managing organization folders and documents
- Managing organization-level types and attributes
- Managing organization templates
- Managing object initialization rules
- Auditing activities within the organization
- Creating and managing profiles
- Creating and managing access control policies
- Configuring numbering and versioning schemes
- Monitoring and managing viewable publishing
- Viewing reports

- Importing and exporting information
- Purging, archiving, and restoring jobs
- Managing organization preferences
- Undoing a user checkout
- Viewing and managing access control rules for objects

## Managing Organization Members, Groups, and Roles

From the **Organization** tab, you control the users who can administer the organization (known as organization administrators) using the **Administrators** page and those who can create and administer application contexts (products, libraries, projects, and programs) using the **Creators** page.

To enable a user to create products, projects, libraries, or programs, the user must be added to the Product, Project, Library, or Program Creators group for the organization. Only members of the organization can be added to the creators groups. Additionally, organization and site administrators who are members of the organization can create products, libraries, projects, or programs; otherwise, you must be a member of the project creators group to create projects, a member of the program creators group to create programs, a member of the product creators group to create products, or a member of the library creators group to create libraries within the organization.

In Windchill ProjectLink, all members of the organization, by default, are allowed to create projects and administer the projects they create. However, if the organization is set up so that members are not automatically added to the project creators group and empowered to create projects, you must manually add members to the project creators group. Organization members must be manually added to the program creators group in order for them to create a program. Projects are considered the least formal of the application contexts (product, library, project, and program), so it is generally appropriate to allow all users to create and administer a project.

User-defined groups created at the organization level can be used when members create product, library, project, program, and shared teams or when access policies are defined. For example, an organization may want to define user-defined groups for each of the functional teams with membership in the organization. For example, assume that the organization defines a sales and marketing group, an engineering group, a publications group, and a quality control group. These user-defined groups can then be added as members of a product or project team without adding each member individually. Furthermore, when user-defined groups are updated, the updates can be refreshed to update all the teams referencing the user-defined groups without the need for each product or project manager to update their team membership. For additional information about updating team membership, see [Context Teams](#) in the Teams chapter.



Shared teams minimize user administration by allowing a collection of roles and associated participants (a shared team) to be used in many application contexts. Any changes to a shared team are immediately available in the contexts where the shared team is used. By default, the site administrator can create shared teams in any organization context and organization administrators can create shared teams in their organization context. Additionally, on the **Creators** page, you control who can create shared teams by adding users to the shared team creators group. Adding a user to this group allows the user to create shared teams from the organization **Teams** page. For additional information about creating and managing shared teams, see [Shared Teams](#) in the Teams chapter.

An organization inherits the roles defined at the site (as defined in the system roles resource bundle). All the roles from the RoleRB.rbInfo file are displayed on the organization **Roles** page. Organization administrators can explicitly hide the roles that they do not want their organization to use.

For more information, see [Creating an Organization Context](#) later in this chapter.

## Creating, Editing, and Managing Organization Folders and Documents

Organization administrators can create folders, documents, and links in the organizations **Folders** page. The following are examples of the types of documents that administrators might create at the organization level:

- Organization configuration documentation.
- Organization environment change log that captures a record of changes to the organization.
- Organization administration rules and procedures.
- Internal training information for organization administrators.
- Key contact list for organization administrators.
- Documents describing organization-level types, document templates, life cycle templates, and workflow process templates.

## Managing Organization-level Types and Attributes

Organization administrators can create types; however, they cannot create attributes unless they are part of the Attribute Administrators group. You can use existing attributes created by the site administrator and link them to types for your organization.

When soft types and soft attributes are created within an organization context, they are kept separate from all other organizations and remain private. For more information about types and attributes, see the [Type and Attribute Manager](#) chapter.

## Managing Organization Templates

An organization context can define a number of document, life cycle, and workflow process templates that can be used by all the application contexts within the organization. To ensure consistency and maximize efficiency, an organization may want to define templates for specifications, presentations, reports, proposals, meeting minutes, and so forth. Life cycle and workflow process templates may be associated with each of the defined templates. Each organization inherits the templates defined in the site and can either use these site-defined templates or override them by defining organization-specific templates with the same name.

The organization can inherit the product, library, project, and program templates defined at the site level, or you can override these templates by defining templates for your organization. If Windchill ProjectLink is installed, you can create project and program templates. If Windchill PDMLink is installed, you can create product and library templates. For more information, see [Creating Context Templates](#).

You can restrict which templates are made available to the applications within your organization by choosing to hide certain templates at the organization level. By only hiding templates, you limit the templates displayed by default and still allow context managers to override which templates are hidden in their particular application context. If you want prevent a template from being used, you can disable it. For more information, see the help from the **Templates** table on the **Organization** tab.

## Managing Organization-level Object Initialization Rules

You can configure how attributes (such as number) are displayed and generated using the Object Initialization Rules administrator. This utility defines how the initial attribute values for the object are established, how the attributes are displayed, and can determine basic relationships, such as life cycle association, when an object is created.

The Object Initialization Rules Administrator utility is available from the **Utilities** page of the **Site** or **Organization** tab. The rules established by the parent site are inherited by each organization context by default, but they can be overridden at the organization level. The rules are defined in XML format; you can view and update them. For more information, see [Defining the Content of XML Documents used for Object Initialization Rules](#).

## Auditing Activities Within the Organization

The auditing framework allows the system to capture historical information of who did what that caused changes in the Windchill database. The security auditing feature also provides a mechanism to retain historical records of security related events, such as privilege changes, team changes, and denial of access. Security audit reporting is helpful in highly regulated industries where it is necessary to identify events that may have resulted in a potential security breach.

Out-of-the-box, only the **Organization Usage** events are enabled. Any other specific events that you want captured must be individually enabled by the site administrator. For more information, see the [Auditing](#) chapter.

## Creating and Managing Access Control Policies

You can define policies that control the level of access to information by organization members. For example, you may want to provide read access to all documents of type Engineering Specification to an engineering group in your organization. In this case, you need to first define an Engineering user-defined group and populate it with the appropriate members, then define a document type of Engineering Specification in your organization. Finally, you need to use the Policy Administrator to define the access policy based on the document type, the group or groups provided access, and the access level.

The Policy Administrator is used to create access control policies. You can create organization-level policies that apply to the entire organization from the **Utilities** page of the **Organization** tab. To establish site-level policies, create those policies that apply to all organizations in the system from the **Utilities** page of the **Site** tab.

For more information about access control, see the [Access Control](#) chapter.

## Viewing and Managing Access Control Rules for Objects

You can view and manage the access control rules for objects (such as folders, parts, and documents) at the organization level. Navigate to the object, and select the **Manage Security** action. From the **Manage Security** window you can view and set the access control for the object.

For more information, see the [Access Control](#) chapter and the online help available from the **Manage Security** window.

## Creating and Managing Profiles

You can dynamically control which actions are visible to user, or group of users, within your organization by associating that information with a profile. Profiles are not an access control mechanism; they are a user interface control mechanism. A profile represents a typical category of user within a company and is based on the roles and privileges associated with that particular user category.

Profiles expose only the necessary functionality and information needed by a user. They create a focused and simplified user interface, reducing confusion by eliminating areas of the user interface that could be distracting.

For more information about profiles, see the [Profile Management](#) section in the Principals (Users, Groups, and Organizations) chapter.

## Configuring Numbering and Versioning Schemes

You can configure the number and versioning schemes used to uniquely identify parts, documents, and other objects in the organization. The organization context inherits the site-level schemes, but can optionally define its own schemes. In general, when a company is hosting Windchill for its internal users, the numbering and versioning schemes should be defined at the site level; you should not override the site-level schemes by defining schemes at the organization level.

For more information about numbering, see [Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects](#).

For more information about versioning schemes, see [Administering the Versioning of Parts, Documents, and CAD Documents](#).

## Monitoring and Managing Viewable Publishing

You can monitor and manage the publishing of viewable files that are optionally generated when CAD models are checked into products, libraries, and projects. You can also configure and update the watermarks used by the visualization tool when viewing document and part content.

For more information, see the [Visualization Services](#) chapter.

## Viewing Reports

You can access several predefined reports for information contained within your organization. For more information on available reporting functionality, see the [Reporting](#) chapter.

## Importing and Exporting Information

You can exchange information with another server, or between servers, or between a server and a file system using the Windchill import/export utilities. These utilities read and write system information in the Windchill XML format. The workflow and life cycle administration utilities accessible to you integrate the import/export functions. See [Importing and Exporting Life Cycle Templates](#) in the Life Cycles chapter and [Importing and Exporting Workflow Templates](#) in the Workflow chapter.

## Purging, Archiving, and Restoring Jobs

You can create and manage purge jobs with the Purge Manager. If you have the applicable functionality installed, you can create purge jobs to permanently remove data from the Windchill system or create an archive of the data to restore at a later time. This would apply to viewing, restoring, and deleting the contents of projects and programs.

## Managing Preferences

Using the Preference Manager on the **Organization** tab **Utilities** page, you can manage preferences at the organization level. Preferences set at the organization level affect all application contexts and users within the organization. Some preference values can be overridden by application contexts within the organization, unless locked at the organization level.

For more information on the Preference Manager, see [Administering Preferences](#).

## Undoing a User Checkout

If a user in your organization has an object checked out, you can undo the checkout, reverting to the last modified version or iteration of the object. This is useful in situations where the user who had the object checked out has left the company, or if they are out of the office when the checked out object is needed.

If you already know the object or objects for which you want to undo the checkout, you can perform the undo checkout action on the object from the **Folders** page of the organization, from the **Folders** page of a product, library, project, or program within your organization, or from the information page for the checked out object. You can also find objects checked out by a particular user, by searching using the **Checked Out By** criteria on the **Advanced Search** page, then performing the undo checkout action for the object from the **Search Results** table.

An organization administrator can also delete an individual user's workspace. For more information, see the *Using Pro/ENGINEER Wildfire with Windchill* guide.

## Out-of-the-box Organization Templates

At installation, the following organization templates are loaded:

Template Name	Description
General (PDMLink)	A sample template that can be used to create an organization for Windchill PDMLink.
General (ProjectLink)	A sample template that can be used to create a general organization for Windchill ProjectLink.
Enterprise	A sample template that can be used to create an enterprise organization.
Supplier	A sample template that can be used to create a supplier organization.
Customer	A sample template that can be used to create a customer organization.

The organization templates can define the same basic information that is discussed in the [Context Administrative Items](#) section of the Administering Contexts chapter. The out-of-the-box organization templates define the following:

- Context structure
- Context participation
- Context access control policies
- Context data

The following sections describe the items that are defined in the templates.

### Context Structure

The organization templates define the following folder structure: Change Log, General, and Policies.

Some organization templates define user-defined groups that are automatically included in the organization. You can add users to the groups.

## Context Participation

The following groups are automatically created when an organization is created:

- Organization administrators
- Product creators (Windchill PDMLink)
- Library creators (Windchill PDMLink)
- Project creators (Windchill ProjectLink)
- Program creators (Windchill ProjectLink)

## Context Access Control Policies

Users who are members of the site Administrators group are granted Full Control to all object types at the site root domain. Users who are members of the organization Administrators group are granted Full Control to all object types at the root domains of the organization (Default, System, Private, and User with the same name as the organization).

During the creation of an organization context, additional domain-based access control rules are automatically created as follows:

- In the organization context System domain, the organization's All Participating Members group is granted read access to templates (such as document templates, life cycle templates, and workflow templates), objects, and initialization rules. For the complete list, see [System Domain Rules](#).
- In the organization context User domain, the organization's All Participating Members group is granted read access to the organization context and organization content.

Additional domain-based access control rules can be defined within an organization template. The following is a list of the access control rules defined in the out-of-the-box organization templates:

- In the General (PDMLink) organization template, organization members (all users) are granted read access to all Released objects in the organization's /Default/PDM domain. Only product and library contexts are affected by this rule.
- In the Enterprise organization template, project type groups are granted read access to projects contained in the project type domains. For example, a project of type Engineering has a corresponding group and organization domain. An access control rule is defined granting read access to the Engineering user-defined group in the Engineering domain. When a project of type Engineering is created, the project is put in the Engineering domain. A user who is added to the Engineering group is able to see all projects of type Engineering.

PTC recommends that you do not modify or delete the default set of access control rules automatically created during the creation of organization, product, library, project, or program contexts. It is acceptable to modify access control rules created from a template.

To adjust access control rules, use the Policy Administrator. To launch the Policy Administrator in the context of the organization, navigate to the **Utilities** page under the **Organization** tab, and click **Policy Administrator**. By launching the Policy Administrator from the **Organization** tab, the context is set to the organization context. In this context, only the domains and subdomains of the organization, plus any ancestor domains from the site are visible. Members of the organization's Administrators group can create and modify rules within the organization's domains. Below is a list of some of the automatically created organization domains with some basic rules:

- /Default – Rules created at this level are inherited by the default domains of all public products, libraries, projects, and programs contained within the organization. Typically, only business objects belong to this domain.
- /Default/PDM – Rules created at this level are inherited by the default domains of all public products and libraries contained within the organization. Typically, only business objects belong to this domain.
- /Default/Project – Rules created at this level are inherited by the default domains of all public projects contained within the organization. Typically, only business objects belong to this domain.
- /Private – Rules created at this level are inherited by the system domains of products, libraries, projects, and programs contained within the organization. The default domain of private products, libraries, projects, and programs also inherit these rules. PTC recommends that no additional access control rules be created within this domain.
- /System – Typically, only administrative objects (such as document templates, team templates, and life cycle templates) are in this domain.

To update the access control rules for an organization domain within the Policy Administrator, select a domain and click **Update**. From the **Administrator Domain** window, click the **Access Control** tab. This provides the list of existing access control rules for this domain. From this tab, you can modify or delete existing rules and create new rules. When creating or updating rules, the list of user-defined groups available from the **Groups** tab include groups defined at the site and organization levels. For more information, see [Searching for Principals in Administrative Clients](#) in the Principals (Users, Groups, and Organizations) chapter.



## Access Control Rules

Out-of-the-box access control rules are described in the following sections.

### Default Domain Rules

No permissions are created by default in this domain.

### System Domain Rules

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the organization.

Object Type	State	Permissions	Principal
WTObject	All	Full Control (All)	ORG ADMIN
DefaultWTContainerTemplate	All	Read	LIBRARY CREATOR
DefaultWTContainerTemplate	All	Read	PRODUCT CREATOR
DefaultWTContainerTemplate	All	Read	PROJECT CREATOR
DefaultWTContainerTemplate	All	Read	PROGRAM CREATOR
WTDocument	All	Read and Download	All Participating Members
WTPart	All	Read and Download	All Participating Members
FilteredDynamicEnumSet	All	Read	All Participating Members
Notebook Template	All	Read	All Participating Members
Notebook Template	All	Read	This Org
DefaultWTContainerTemplate	All	Read	All Participating Members
Cabinet	All	Read	All Participating Members
SubFolder	All	Read	All Participating Members
TeamTemplate	All	Read	All Participating Members
WfTemplateObject	All	Read	All Participating Members
LifeCycleTemplate	All	Read	All Participating Members
EPMDocument	All	Read and Download	All Participating Members

Object Type	State	Permissions	Principal
Project2	All	Create	PROJECT CREATOR
WTLibrary	All	Create	LIBRARY CREATOR
TypeBasedRule	All	Read	All Participating Members
PDMLinkProduct	All	Create	PRODUCT CREATOR
TableViewDescriptor	All	Read	All Participating Members
WTGroup	All	Read, Modify, Create, and Delete	SHARED TEAM CREATOR
ContainerTeam	All	Read	PROJECT CREATOR
ContainerTeam	All	Create	SHARED TEAM CREATOR
ContainerTeam	All	Read	PRODUCT CREATOR
ContainerTeam	All	Read	LIBRARY CREATOR
Report Template	All	Read	All Participating Members

**Note:** The rules for all object types are defined programmatically when an organization context is created; they are not defined through the template that is used.

#### Organization User Domain (same name as the organization) Rules

The following table lists the combination of object type, life cycle state, and granted permissions that the organization user domain out-of-the-box rules define for the organization.

Object Type	State	Permissions	Principal
WTObject	All	Full Control (All)	ORG ADMIN
WTObject	All	Full Control (All)	OWNER
OrgContainer	All	Read	All Participating Members
WTGroup	All	Read	Unrestricted Organizations
WTOrganization	All	Read	All Participating Members
WTOrganization	All	Read	ALL

Object Type	State	Permissions	Principal
WTUser	All	Read	Unrestricted Organizations
OrgContainer	All	Read	<i>organization principal</i>
WTGroup	All	Read	<i>organization principal</i>
WTUser	All	Read	<i>organization principal</i>

**Note:** The rules for all object types are defined programmatically when an organization context is created; they are not defined through the template that is used.

#### **/Default/PDM Domain Rules for General (PDMLink) Template**

The following table lists the combination of object type, life cycle state, and granted permissions that the out-of-the-box rules define for the General (PDMLink) organization template:

Object Type	State	Permissions	Principal
WTOBJECT	Released	Read	This Org

#### **Default/PDM Domain Rules**

The following table lists the combination of object type, life cycle state, and granted permissions that the Default/PDM domain out-of-the-box rules define for the organization:

Object Type	State	Permissions	Principal
WTLibrary	All	Create	LIBRARY CREATOR
WTOBJECT	Released	Read	<i>organization principal</i>
PDMLinkProduct	All	Create	PRODUCT CREATOR

**Note:** The rules are set programmatically when an organization context is created.

## Default/Project Domain Rules

The following table lists the combination of object type, life cycle state, and granted permissions that the Default/Project domain out-of-the-box rules define for the organization:

Object Type	State	Permissions	Principal
Project2	All	Create	PROJECT CREATOR

**Note:** The rules are set programmatically when an organization context is created.


## Context Data

For organization contexts created using the General or Enterprise organization templates, document types are inherited from the site.

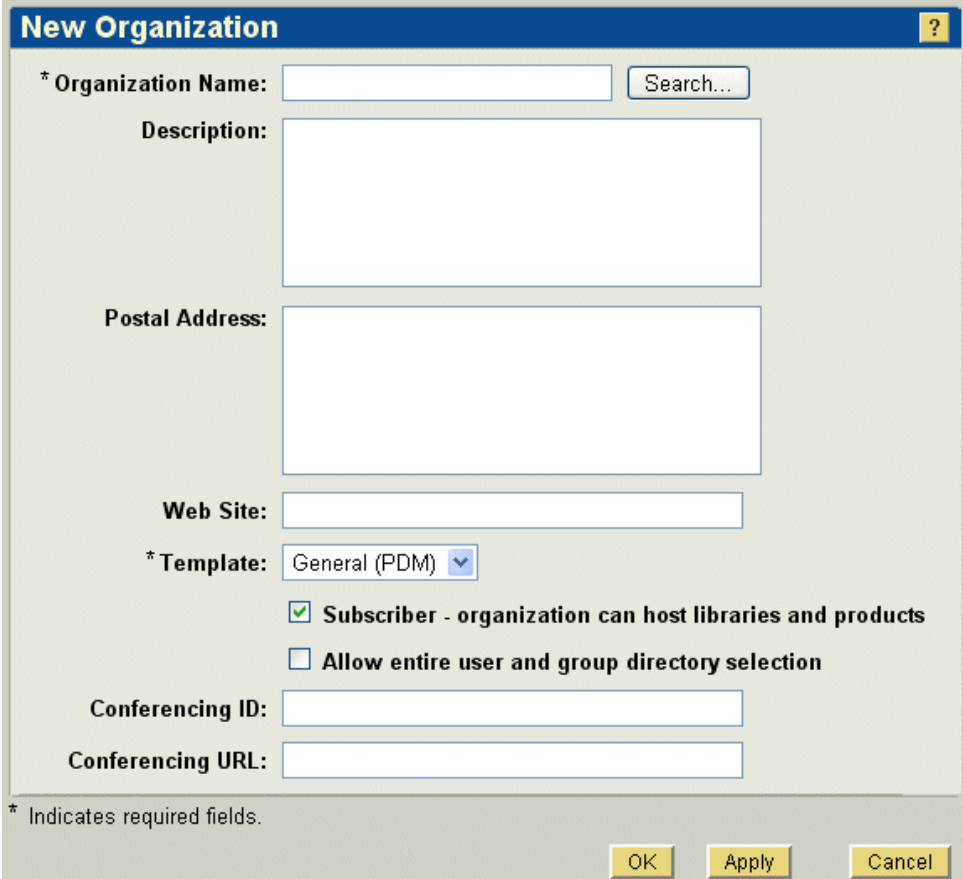
For organization contexts created using the Customer or Supplier organization templates, document types include those defined in the template as well as the ones inherited from the site. The Customer and Supplier organization templates define the following document types: Analysis, Contract, Design, Drawing, Issues, Memo, Proposal, Report, Requirements, Schedule, Specification, and Statement of Work.

For all organization contexts, document attributes, part attributes, and project types are inherited from the site.

## Creating an Organization Context

Only site administrators have the permissions to create organization contexts. Organization contexts are created from the **Organizations** page (visible only to site administrators) on the **Organization** tab. Click the new organization icon  at the top of the **Organizations** table.

The **New Organization** window that opens is similar to the following:



**New Organization** ?

\* **Organization Name:**

**Description:**

**Postal Address:**

**Web Site:**

\* **Template:**

☒ **Subscriber - organization can host libraries and products**

☐ **Allow entire user and group directory selection**

**Conferencing ID:**

**Conferencing URL:**

\* Indicates required fields.

To associate an existing organization principal to the organization context, click **Search**, next to the **Organization Name** field, and search for existing organization principals. You can also type in a new organization name (one that does not match an existing organization principal); in this case, a new organization principal is created along with the organization context.

**Note:** The internet domain defined on the organization principal is important when creating new soft types. The internet domain is used to distinguish which organization principal owns the type. During the installation process, the initial organization principal created included an internet domain. This organization principal is associated with the site context; any types defined at the site context are available to all organizations. In a multiple organization environment, PTC strongly recommends that the internet domains of new organizations be distinct from the internet domain for the site. Do not use underscores ( \_ ) in the internet domain.

By default, the **Subscriber** check box is selected. This means the organization can host products and libraries (for Windchill PDMLink, Arbortext Content Manager, and Pro/INTRALINK 9.0), and projects and programs (for Windchill ProjectLink). Members of a subscribing organization who belong to the appropriate application context creators group can create application contexts (products, libraries, projects, or programs) within the organization.

For Windchill ProjectLink, the **Automatically add new members to project creators group** check box shows on the window (not shown in figure above) and is selected by default.

By default, you are restricted to seeing only users who belong to the organization. Select the **Allow entire user and group directory selection** check box to provide the ability to search for all users. For user-defined groups, this setting allows you to see group entries contained in the Administrative LDAP and Enterprise LDAP directory hierarchy level of the installed directory service (which shows the groups defined in the current organization context) and any additional enterprise directories setup through a JNDI adapter (search scope set to ONELEVEL). If you have created groups in a different organization context, you cannot see those groups in this organization context. For information about the installed directory services, see the [Working with LDAP Directory Services](#) section in the Principals (Users, Groups, and Organizations) chapter.

For additional information on creating an organization context, see the help available from the **New Organization** window.

**Note:** The site administrator user who created the organization is owner of the organization and is the user identified in the **Owner** attribute on the organization information page. (The organization owner of the initial organization created during installation is the administrator user that was identified during installation, by default wadmin.) The organization owner can be changed to another user by editing the organization.

For a description of the contents of the templates, see [Out-of-the-box Organization Templates](#).

## Owning Organization Principals

Parts or documents created within an organization context are automatically owned by the organization principal associated with the organization context. However, the site may choose to allow users to specify a different organization principal, such as one associated with an external vendor or supplier, as the owner of an object. For example, if a part used in a part structure is acquired from a vendor, you can create an organization principal for that vendor, and select that vendor's organization principal as the owning organization for the part.

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

- Set up domains for use with owning organizations. For more information, see [Setting Up Domains for Use with Owning Organization Principals](#).
- Create or update organization principals to be used as owning organizations, sometimes referred to as internal organization. For more information, see [Creating and Updating Owning Organization Principals](#).
- Enable the appropriate preferences to display the owning organization on objects, using the Preference Manager:
  - **Expose Organization** -- Displays the organization for all objects except change objects.
  - **Expose Organization for Change Management Objects** -- Displays the organization for change objects. The Expose Organization preference must also be enabled.
- Configure the context where the parts or documents will reside so that the user enters the part or document number (rather than having the numbers auto-generated). How to turn off autonumbering is described in the Object Initialization Rules help that is accessible from the Object Initialization Rules Administrator. For information about the Object Initialization Rules Administrator, see [Object Initialization Rules](#).

## Setting Up Domains for Use with Owning Organization Principals

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

A simple approach to setting this up is the following:

1. Create a domain that will be used specifically for this purpose. For example create the /Vendors domain using the Policy Administrator from the site context.
2. Create the access control rule in the Vendors domain that grants READ permission on WTOrganization objects to ALL users.
3. When creating the organization principals that represent external vendors and suppliers using the Principal Administrator, select to the /Vendors domain for the **Windchill Domain** field.

For more information on domains, see [Managing User Access to Data](#).

## Creating and Updating Owning Organization Principals

Use the Principal Administrator to create a new organization principal or update an existing organization principal for use as an owning organization by including the following attributes on the principal:

- The **Organization ID Type** attribute indicates the type of organization identifier that is specified for the **Organization ID**. Select the type from the drop-down list. By default, the drop-down list contains **CAGE**, **DUNS**, and **ISO6523**.

**Note:** The organization identifier, such as a CAGE code or DUNS number, is an internationally coded number that is assigned when the organization registers with a specific site. For site registration information, see the [Registration Authority](#).

If the drop-down list does not contain the type required, you can specify a new list by adding the wt.org.organizationTypes property to the wt.properties file. The wt.org.organizationTypes property value is a comma-separated list of coding systems, where each coding system has two parts in the following format:

`<ICD_number>/<coding_system_name>`

where:

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

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



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

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

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

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

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

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

Use the xconfmanager utility to add the wt.org.organizationTypes property. For more information, see [About the xconfmanager Utility](#) in the Administration Overview chapter.

- The **Organization ID** attribute specifies the globally unique organization identifier under which the organization is registered.
- The **Windchill Domain** attribute identifies the administrative area where the organization object resides. The domain selected must have access control rules set for the WTOrganization object type so that the users who create part and documents have read access to the organization objects that you want to use as internal organizations. For information on how to set up a domain for this use, see [Setting Up Domains for Use with Owning Organization Principals](#).

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

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

where:

<ICD\_number> is the international code designator number assigned to the organization ID type. For example, the CAGE ICD number is 0141. For a list of ICD numbers, see the [Registration Authority](#).

<org\_ID> is the organization identification number assigned when the organization was registered.

## Using the Organization Utilities Page

The utilities listed on the **Utilities** page, which is accessible by clicking the **Utilities** link from the **Organization** tab, allow you to perform administrative actions at an organization level. Some of these utilities appear on other tabs. The difference is the context from which the utility is launched.

The utilities are grouped according to whether they are system administration utilities or business administration utilities. Many of links provided on the page give you access to the utilities that you need to use to perform the duties described in [Typical Duties of Organization Administrators](#).

To explore the use of each utility, click the corresponding link on the page and then click the help icon in the window that opens.

## Changing an Established Internet Domain

If the internet domain associated with an organization context is changed by modifying the organization principal using the Principal Administrator, the previous internet domain is no longer recognized as being associated with the organization context. Any types associated with the organization context's previous internet domain are no longer recognized as mapping to the organization. This can be corrected by adding the previous internet domain mapping to the internetDomain.properties file in the following format:

```
<internet_domain>=/wt.inf.container.OrgContainer=<org_name>
```

where *<internet\_domain>* is the previously defined internet domain for the organization principal, and *<org\_name>* is the name of the organization principal.

For example, the World Wide Tractors organization principal has an internet domain of worldwidetractors.com. A soft type of Engine created in that organization has the fully qualified name of com.worldwidetractors.Engine.

If the internet domain is later changed using the Principal Administrator to wwtractor.com, then the com.WorldWideTractors.Engine soft type is no longer associated with the organization's internet domain. By adding the following mapping to the internetDomain.properties file:

```
worldwidetractors.com=/wt.inf.container.OrgContainer=World Wide Tractors
```

any soft types associated with the old internet domain now map to the updated organization.

The internetDomain.properties file is also useful for historical tracking of previous internet domains associated with an organization principal.

**Note:** Updating the internetDomain.properties file constitutes a customization, and you must save a copy of the updated internetDomain.properties file using your site-defined practices for customization. For more information, see the *Windchill Customizer's Guide*.

## Using the OrganizationSync Utility for User Organization Changes

When a user is moved from one organization to another by using a third party tool that allows your site to modify user entries in your directory service, the domains of the user and of the user's personal cabinet are not automatically changed. As the site administrator, you can use the OrganizationSync command-line utility to update organization-related information for users who have changed organizations. After changing the user's organization through a third party tool, run the OrganizationSync utility to change the other organization-related information for the user. Using the utility ensures that properties of a user or set of users are in sync with their current organization.

**Note:** If you use the Principal Administrator (and not a third party tool) to update user information in your directory service, then the OrganizationSync utility is automatically run as a result of changing a user's organization from the **Update User** page.

The OrganizationSync utility does the following:

- Assigns a set of user-related business objects to the administrative domain of the user's current organization. If a user does not have an organization, these objects are assigned to the Unaffiliated domain. The objects reassigned include the WTUser object, the user's personal cabinet and content of that personal cabinet, and the user's notebook.
- Removes the user from the administrators and creators groups of their previous organization context, if necessary. The utility determines the previous context by extracting the context of the user's previous administrative domain.
- Updates the organization groups of any teams, in which the user was a member, to reflect the user's current organizational membership.

**Note:** It is safe to run the OrganizationSync utility on a user that was not actually updated.

The syntax of the OrganizationSync command is as follows:

```
java wt.inf.container.OrganizationSync [--users <user_list>] [--domains  
<domain_list>]{--domain-hierarchies <domain_list>} [--organizations  
<organization_list>}
```

The brackets ({} ) in the syntax indicate optional parameters and indicate parameters that you specify together. Parameter names are case-sensitive; enter the names using the case shown in the syntax and the following table.

Parameter Value	Description
<user_list>	A space-separated list of user names.
<domain_list>	A space-separated list of domain paths.
<organization_list>	A space-separated list of organization names.

When <domain\_list> is used with the --domains parameter, all users in the domain are synchronized. When <domain\_list> is used with the --domain-hierarchies parameter, all the users in each domain and in all descendent domains are synchronized.

**Note:** Always run the OrganizationSync utility from a windchill shell. For details on using the windchill shell, see [About the windchill shell](#).

Examples:

```
//Update the foo and bar users
java wt.inf.container.OrganizationSync --users foo bar

// Update all users in the site-level Unaffiliated domain
// and the "PTC Inc" domain in the "PTC Inc" org container
java wt.inf.container.OrganizationSync --domains
[/]/User/Unaffiliated "[/wt.inf.container.OrgContainer=PTC
Inc]/PTC Inc"

// Update all users in the site-level User domain, and in
// all of its child domains
java wt.inf.container.OrganizationSync --domain-hierarchies
[/]/User

// Update all members of the "Acme Inc." and
// "Widget Intl." organizations
java wt.inf.container.OrganizationSync --organizations "Acme
Inc." "Widget Intl."
```

In the above examples, the container path is enclosed in brackets to separate it from the domain path. [/] is the Site (root) container.

# Best Practices

The following best practices are recommended by PTC.

## For All Windchill Solutions

The following best practices are recommended for all Windchill solutions.

### E-mail Addresses

Ensure that users have an e-mail address; many features in Windchill require that users have an e-mail address. If users do not have the e-mail attribute set in their user directory service entry, they cannot participate in the features that require an e-mail address.

### Refreshing Teams

The recommended practice for defining groups is to define them at the organization level whenever possible and then use the user-defined groups in product, library, program, and project teams. The **Groups** page on the **Organization** tab can be used for this purpose.

Over the life of a product, library, project, or program, the members of user-defined groups referenced in teams may change. You need to put together a plan for how to update the teams on a regular basis. You can use the **Refresh Teams** button on the **Organization** tab to immediately update the teams affected by a group membership change. However, doing this uses a lot of system resources that can slow down system performance during peak hours. Instead, PTC recommends that you schedule daily and weekly team updates using tasks provided by PTC. The tasks can be run in off-peak hours by using a queue. Automating the update of teams is turned off by default. For details on turning on this feature, see the [Refreshing User-Defined Groups](#) section of the Teams chapter.

## For Windchill Solutions with Products and Libraries

The following best practices are recommended for Windchill solutions which use products and libraries.

### Setting Object Initialization Rules

The object initialization rules include the ability to set default values and display characteristics for the following attributes:

- Folder paths
- Default life cycle template
- Default team template

- Default numbering scheme
- Default version scheme

Additionally, rules can be used to set the following:

- Available life cycle templates
- Attribute values
- Display characteristics for other attributes

Although these rules can be set for individual product and library contexts, PTC recommends that you set them at either the organization or site level so that an entire organization or site is using the same set of rules. This promotes reuse and reduces unnecessary duplication at lower context levels.

For more information, see the [Object Initialization Rules](#) chapter.

## Setting Up Enhanced Life Cycle Templates

The out-of-the-box Product Design template includes object initialization rules that set different default life cycle templates and versioning schemes for parts, documents, and CAD documents than what is set out-of-the-box at the site level. This template demonstrates the use of simplified life cycles and state-based versioning. Unless you have overriding business practices that require unique rules, PTC recommends that to use these features, you modify the site object initialization rules and remove them from the Product Design template.

## For Windchill Solutions with Projects and Programs

The following best practices are recommended for Windchill solutions which use projects and programs.

### Allowing All Organization Members Read Access to Project or Program Content

To allow all organization members access to project or program content, even if they are not a member of the project or program, you must update the access control policies.

To update an existing organization, navigate to the Policy Administrator on the **Utilities** page of the organization. Select the /Default/Project domain, and create a new access control rule granting the This Org group Read access to the WObject type in all states. By creating this new access control rule, all organization members will have read access by default to all business objects within all project or programs that are created by selecting an **Access Group** option other than **Project Members Only**. (This access control rule will not affect access to objects within projects or programs created with the Members Only access group.)

For more information on the Policy Administrator, see [Using the Policy Administrator](#).

Alternatively, the site administrator can update the organization template so that all organizations subsequently created include this access control policy by default.

Add the following XML to the organization template:

```
<AccessControlRule>
  <domainName>/Default/Project</domainName>
  <externalTypeId>WCTYPE|wt.fc.WTObject</externalTypeId>
  <lifecycleState>ALL</lifecycleState>
  <WTPrincipalReference isInternal="true">
    <groupName>This Org</groupName>
    <groupType>This Org</groupType>
  </WTPrincipalReference>
  <grantPermissionSet>
    <AccessPermissionSet>
      <permissionField name="Read"/>
    </AccessPermissionSet>
  </grantPermissionSet>
</AccessControlRule>
```





# 6

## Projects and Programs

This chapter provides an overview for administering projects and programs and describes the typical duties that an administrator performs. It also provides additional information about some of the main administrative tasks for projects and programs.

<b>Topic</b>	<b>Page</b>
Overview .....	6-2
Typical Duties of Project and Program Managers .....	6-2
Out-of-the-box Project and Program Templates .....	6-10
Configuring Part and Document Relationships.....	6-13

## Overview

Project and program managers (also known as project and program administrators) are responsible for creating and managing projects and programs hosted by a parent organization. The capabilities of project and program managers are nearly identical.

Project and program managers control the project and program configuration, and control the membership in their project and program teams within the confines of a specific project or program application context. They control access to project and program information; they define the schedule, resources, and plan details, and monitor and manage the project and program activities.

Project application contexts are on-line work areas used to collect information that teams need to track, collaborate, and manage as well as plan work activities to meet a specific objective. Project contexts are defined by project creators who are authorized by the parent organization. Projects inherit templates, roles, user-defined groups, and policies from their parent organization context. The project manager can also define project-specific templates, roles, and policies.

Program application contexts provide a central area to link together information, people, and plans for a series of projects. Program contexts are defined by program creators who are authorized by the parent organization. Programs inherit templates, roles, user-defined groups, and policies from their parent organization context. The program manager can also define program-specific templates, roles, and policies.

For general information about context contents and how to create contexts, see [Contexts](#).

## Typical Duties of Project and Program Managers

Project and program managers are responsible for creating and managing the content of projects and programs. The capabilities of project and program managers are nearly identical. Your typical duties include the following:

- Creating and updating the project or program
- Overriding user profiles
- Managing team members and roles
- Creating, updating, and managing documents and folders
- Creating, updating, and managing activities, deliverables, resources, and action items
- Managing document templates
- Importing and exporting information

- Undoing a user checkout
- Viewing and managing access control rules for objects

The following sections describe some of the duties in more detail.

## Creating and Editing Projects and Programs

When you create a project or program, you can choose the project or program template on which the project or program is based, define the key project or program attributes, define the project or program properties and select the configuration options. You can also update the project or program attributes as work progresses. These attributes include the project or program description, scope, status description, risk value and description, project or program number, and so forth.

Only members of the project creators group and the organization administrator in an organization are allowed to create projects. By default, organizations allow all members to create projects, but the organization or site administrator may update an organization restricting project creation privileges to identified creators.

Only members of the program creators group and the organization administrator in an organization are allowed to create programs. By default, only the organization administrator can create programs. The organization administrator must manually add users to the program creators group to allow users to create programs.

For information about creating projects and programs, see the *Windchill ProjectLink User's Guide* or the help available from the **New Project** or **New Program** windows.

**Note:** If you use a shared team that is not extended locally when creating a program or project, and you are not a member of the shared team, you may not have access to the program or project that is created.

## Overriding User Profiles

Profiles define which parts of the user interface (for example, actions, tabs, and attributes) should be visible to a principal (such as a user, group of users, or the users of an organization).

For more information about creating and editing profiles, see the Profiles online help.

## Managing Team Members and Roles

Project and program managers can add members and team roles to a local team, as well as override profiles.

If the team includes a shared team, the shared team is managed from the organization context, and not the project or program context.

For more information about local and shared teams, see [Context Teams](#).

For more information about profiles, see the [Profile Management](#) section in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

## Establishing Roles

Each project team contains two fixed roles that cannot be removed: Project Manager and Guest. Each program team contains two fixed roles that cannot be removed: Program Manager and Guest. Additionally, you can select team roles from the list of roles inherited from the parent organization, or you can create new roles.

The Guest role is designed to include user-defined groups and users that are not active team members and need only read access to project or program information. It is possible to hide the Guest role for projects and programs. For more information, see [Hiding the Guest Role](#).

Guest members do not receive invitations; projects or programs in which a user is a guest do not show up on project or program lists for the user. You can allow guests to see project and programs on their context lists by adding the following property and value to the wt.properties file:

```
com.ptc.netmarkets.project.displayContainerListForGuest=true
```

Use the -s parameter on the xconfmanager command to add Windchill properties. See [About the xconfmanager Utility](#) for details. This property controls the guest visibility on all project and program lists.

The creator of the project or program is automatically established as both a project or program manager and as the project or program owner. The project or program manager can change the owner and can add members to the project or program managers role. All members of the project or program managers role have the same privileges.

You can invite groups to the project or program roles from the user-defined groups defined in the parent organization.

## Controlling the Visibility of Actions

As the project or program manager, you can control visibility to information and actions within the project or program based on the roles in the project or program local team.

For additional information, see [Restricting the Visibility of Actions in the User Interface through Teams](#).

## Overriding Profiles

As the project or program manager, you can override profile settings that are established at the site and organization contexts. For more information about profiles, see the [Profile Management](#) section in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

The site or organization administrator determines whether the various user interface elements and actions are configurable from an application context. You can override an action or user interface element in a context instance using Configure Actions for Roles which is accessed from the Team page. For additional information, see [Restricting the Visibility of Actions in the User Interface through Teams](#).

## Creating, Editing, and Managing Documents and Folders

You can create documents for the project or program, and you can define a folder structure. By default, any member of the project or program can define folders and subfolders in the folder structure.

You have full control over all documents created in the project or program; no member can prevent you from reading, editing, or deleting any object. You can also modify the access rules on any folder or any document.

You can delete discussion topics and postings. Non-project or program managers can only post to discussion topics; they cannot delete postings or topics, even if they created them.

## Creating, Editing, and Managing Activities, Deliverables, Resources, and Action Items

You alone can create activities, milestones, deliverables, and resources in the project or program. The owner of an activity, milestone, or deliverable can edit and complete the item, but only project or program managers can create plan items.

## Limiting Edit Privileges for All Action Items

You can limit the edit privileges on all action items, preventing the action item assignee from updating the following fields:

- Name
- Description
- Assignee
- Due Date
- Priority

To limit the update privileges for all action items, a site or organization administrator can set the **Restrict Action Item Update by Assignee** preference to Yes using the Preference Manager from the **Utilities** page on the either **Site** or **Organization** tab. Assignees will not be allowed to update the fields listed above for any action items.

## Managing Templates

You control the document templates available to a project or program. You can:

- Add to the list of document templates inherited from the parent organization or the site, by creating a new document template within your project or program.
- Override the inherited template by creating a project- or program-specific template of the same name.
- Disable a document template that was created within the context of the project or program.
- Choose to hide or show inherited document templates for use in your project or program.

Document templates that are enabled and shown are available to users when creating a new document from a template on the project or program **Folders** page; document templates that are hidden or disabled are not available. For more information, see the help available from the **Templates** table.

## Importing and Exporting Information

You can export information from one project and import it into another project, or export information from one program and import it into another program on the same system or a different Windchill system. Generally, you can only import into a system at the same or higher release version as the system from which the data was exported. You can export project or program information as a template. A site or organization administrator can then create a template from that information that can be reused by the organization.

You can also save an existing project or program as a new project or program as a means of quickly starting a similar project or program.

You can also import a Microsoft Project plan into the system and export project and program information from Windchill into the Microsoft Project format.

**Note:** By default, milestones that have resources assigned in Microsoft Project are converted to activities in Windchill ProjectLink. To change this behavior so that milestones that have resources assigned in Microsoft Project are converted to milestones in Windchill ProjectLink and the resources are dropped, set the following property to TRUE in the wt.properties file:

```
wt.projmgmt.msproject.dropMilestoneResource
```

## Undoing a User Checkout

If a user in your project or program has an object checked out, you can undo the checkout, reverting to the last modified iteration of the object. This is useful in situations where the user who had the object checked out has left the company, or if they are out of the office when the checked out object is needed.

If you already know the object or objects for which you want to undo the checkout, you can perform the undo checkout action on an object from either the **Folders** page of your project or program, or from the information page for the checked out object. You can also find objects checked out by a particular user, by searching using the **Checked Out By** criteria on the **Advanced Search** page, then performing the undo checkout action for the object from the **Search Results** table.

## Viewing and Managing Access Policies

You can view and manage the access control rules for objects (such as folders, and documents) in your project or program. Navigate to the object, and select the **Manage Security** action. From the **Manage Security** window you can view and set the access control to the object for roles within your project or program team.

For more information, see [Creating and Managing Access Control Rules](#) in the Access Control chapter, and the online help available from the **Manage Security** window.

You can define policies that control the level of access to information in a program or project. When defining a policy, the object types and user-defined groups defined in the parent organization can be used as well as the system groups representing the team roles in a program or project. For example, you could create a policy that provides Write access to all documents of type “Agenda” to the project team role/group called “Scheduler”.

For general information about domains and policies, see [Administering Domains and Policies](#).

For additional information about creating or updating access control rule policies, see the [Access Control](#) chapter.

## Managing Utilities

A number of administrative utilities are available through the **Utilities** page of the **Project** and **Program** tabs.

**Note:** By default, only site and organization administrators can access the utilities on the **Project** tab and must act on behalf of the project manager. Access is restricted by default because these utilities are complex and require a level of training that is not expected for typical project managers. The site or organization administrator can set the **Project Utilities Page** preference to grant all project managers access to the project utilities functions.

The following utilities are available through the **Utilities** page:

- File Server Manager
- Replication Rules Administrator
- Numbering Schemes Administration
- Object Initialization Rules Administrator
- Policy Administrator
- Preference Manager
- Publish and Thumbnail Control
- Publish Monitor
- Publish Scheduler Administrator
- Report Manager
- Visualization Configuration Administrator

### File Server Manager

You can download light-weight file server installers and manage file servers for a project or program.

### Replication Rules Administrator

You can establish project and program content replication rules. Replication is useful when sharing large files with remote sites where the remote users have low bandwidth connections to the Windchill server. Replication sites are configured so that document and part content files are replicated at a remote site and made available locally to the remote users through a higher bandwidth connection. Replication can decrease the time required to upload and download files at the remote site. Site administrators must define the replica site and the replication schedule; project and program managers configure the replication rules for a particular replication site established for the system.



Define replication only for those projects or programs in which remote sites are participating. Replication imposes a substantial performance burden on the Windchill server.

Replication rules can be configured on a project-by-project or program-by-program basis. See the *Windchill System Administrator's Guide* for information about replication.

## Numbering Schemes Administration

You can update the numbering scheme applied to establish uniqueness for documents in a project or program, and parts in a project. For more information see [Changing Numbering Schemes](#) in the Object Initialization Rules chapter.

## Object Initialization Rules Administrator

You can configure how attributes (such as number) are displayed and generated using the Object Initialization Rules administrator from a project or program. The Object Initialization Rules Administrator how the initial attribute values for the object are established, how the attributes are displayed, and can determine basic relationships, such as life cycle association, when an object is created.

The rules established by the parent organization are inherited by each project or program by default, but they can be overridden by a project or program. The rules are defined in an XML format; you can view and edit them if a site or organization administrator has given you access to them.

## Policy Administrator

From the Policy Administrator you can manage policies, such as notifications and access control, for your individual project or program.

## Preference Manager

From the Preference Manager, you can view and set the preferences available for your individual project or program. For more information about preferences and the Preference Manager, see [Administering Preferences](#).

## Publish Monitor

You can access the Publish Monitor utility to monitor the publishing of viewable files generated when CAD models are checked into the project.

## Publish Scheduler Administrator

You can manage the scheduling of publishing CAD files that are checked into the project.

## Report Manager

You can create, update, and delete reports from the Report Manager utility using the Query Builder tool. A report is the output of a predefined query, or report template, and is used for generating statistics and matrix against business information. For more information on reports, see the [Reporting](#) chapter.

## Visualization Configuration Administrator

If given access to this utility, you can manage the visualization configurations to set up watermark files that users see when they view content from the project or program. You can also set publish rules for the project or program.

## Out-of-the-box Project and Program Templates

When Windchill ProjectLink is installed, the following templates are loaded:

Template Name	Description
General	Sample templates that can be used to create a general project or program.
Design	A sample template that can be used to create a design project.
Manufacturing	A sample template that can be used to create a manufacturing project.
Custom	A sample template that can be used to create a custom project.

The project and program templates can define the same basic information that is discussed in the [Context Administrative Items](#) section of the Contexts chapter. However, the out-of-the-box templates define only the following:

- Context structure
- Context environment
- Context participation

The following tables describe the items that are defined in the templates.

<b>Container Structure</b>				
	<b>General (Project or Program)</b>	<b>Design</b>	<b>Manufacturing</b>	<b>Custom</b>
Folder structure	General	Analysis Contracts General Designs Plans Prototypes Specifications Standards	Analysis Contracts Designs General Plans Specifications	Changes Contracts Designs General Installation Plans Prototypes Specifications

<b>Container Environment</b>				
	<b>General (Project or Program)</b>	<b>Design</b>	<b>Manufacturing</b>	<b>Custom</b>
Discussion topics	General	Analysis Design General Manufacturing Specifications	Design General Manufacturing Quality Specifications	Changes Contract Design General Manufacturing Quality Specifications
Notebook template	Documents General Links Parts	Documents General Links Parts	Documents General Links Parts	Documents General Links Parts

Container Participation				
	General (Project or Program)	Design	Manufacturing	Custom
Project or program team roles	Guest Members Project or Program Manager Collaboration Manager	Consultant Designer Guest Manufacturer Members Project Manager Collaboration Manager	Engineer Guest Manufacturer Members Production Planner Project Manager Purchasing Agent Quality Engineer Supplier Collaboration Manager	Customer Engineer Guest Members Project Manager Quality Engineer Supplier Collaboration Manager

Template Plan Objects				
	General (Project or Program)	Design	Manufacturing	Custom
Milestones	Project or Program Start Project or Program Finish	Project Start Contract Signed Specifications Approved Design Approved Prototype Approved Design Released Project Finish	Project Start Contract Signed Specifications Approved Design Approved Testing Complete Prototype Approved Production Part Approved Production Initiated Project Finish	Project Start Contract Signed Specifications Approved Design Approved Testing Complete Prototype Approved Production Part Approved Production Initiated Project Finish
Deliverables	Project or Program Start Project or Program Finish	Project Start Contract Signed Specifications Approved Design Approved Prototype Approved Design Released Project Finish	Project Start Contract Signed Specifications Approved Design Approved Testing Complete Prototype Approved Production Part Approved Production Initiated Project Finished	Project Start Contract Signed Specifications Approved Design Approved Testing Complete Prototype Approved Production Part Approved Production Initiated Project Finish

# Configuring Part and Document Relationships

The relationships between a part and a document, two parts, or two documents vary depending upon how the two objects are linked. This section briefly describes the different types of links that form these relationships and then describes the configuration options that are available.

For detailed information about understanding relationships between parts and documents, see the *Windchill ProjectLink User's Guide*.

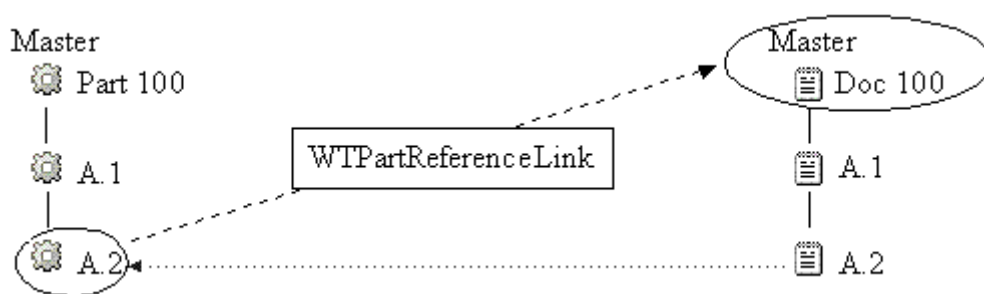
## Part to Document Relationships

**Note:** Part to document relationships apply to projects only; parts are not used within programs.

The association between a part and a document is created from the part information page, document information page, Product Structure Explorer, or Product Structure Browser. Associations can be viewed from the part information page under the **Related Objects > Documents** menu, on the Described By Documents table or the References Documents table, depending on the relationship type. These same relationships can be viewed from the document information page on the **Referenced by Parts** and **Referenced by Part Instances** tables.

There are two types of links that can be established:

- A WTPartReferenceLink -- Using this type of link always links the part to the master document. Regardless of the version of the document the user selects, the part will always link to the master document when this link type is used. This is true even when the system has been configured to allow the user to select the link type and the document type.



Reference links always display a document version of the master based on life cycle state rules. For example, when a user clicks the reference link for Part 100 version A.1, Windchill searches for the latest released version of Doc 100 to display. If no version of the document has been released, it displays the latest working version (in this case, Doc 100 version A.2).

How to configure the life cycle state rules is described later in this section.

- A WTPartDescribeLink -- Using this type of link links the part to the latest iteration of a document and vice versa.



For example, when the user clicks a describe link for Part 100 version A.2, Windchill searches for the latest iteration of Doc 100 to display (in this case, Doc 100 version A.2).

You can configure the link types for part to document relationships in the following ways:

- From **Related Objects > Documents** on the part information page, the user creates the link using the actions from the **Described By** or **References** tables.
- From **Related Objects > Parts** on the document information page, the user creates the link using the actions from the **Describes** or **Referenced By** tables.
- From **Structure** on the part information page, the user creates the link using the **Add References Document** or **Add Described By Document** actions from the action list on the **Product Structure** table.

To allow the user to select the link type regardless of the document soft type, set the Part to Document Association Logic preference to Yes from the Preference Manager on the **Site** or **Organization** tab **Utilities** page. For details on using the Preference Manager, see [Administering Preferences](#).

Additional configuration options are described in the following sections.

For additional information on part to document relationships, see the *Windchill ProjectLink User's Guide*.

## Revised or Saved Part to Related Document

When a user revises a part using the **Revise** action or saves the part using the **Save As** action, the new version of the part carries forward the link to the document by default. While the **Revise** action always carries the link forward, you can choose to prevent the link from being carried forward during a **Save As** action by removing the Relationship copy rules related to this operation from wt.properties.

For example, assume the following properties are set in wt.properties:

```
wt.enterprise.copyRuleDelimiter=,  
  
wt.enterprise.copyRulesN=wt.part.WTPart,Relationship,  
wt.part.WTPartReferenceLink-references
```

The first property sets the delimiter for the copy rules to the comma (,).

The wt.enterprise.copyRulesN property is the Relationship copy rule for wt.part.WTPart. This rule copies the references forward when the type of link is WTPartReferenceLink.

If you remove the wt.enterprise.copyRulesN property, then reference links are not carried forward.

**Note:** There can be no gaps in the sequence of copy rules. If you remove a copy rule, you must renumber those rules that follow. For example if there are six copy rules and you remove copyRules5, then you must renumber copyRules6 so that it is copyRules5.

Use the xconfmanager utility when modifying the wt.properties file. For information on using this utility, see [About the xconfmanager Utility](#). For additional information about the properties used for copy rules, see the description of wt.enterprise.copyRules in the properties.html file.



**Caution:** PTC recommends that you do not change the value of the wt.enterprise.copyServiceRules property. The property is used by internal services.

## Document Version Used with Reference Link

As described earlier, reference links (WTPartReferenceLink) link to a document master, but display a document version of the master based on the life cycle state rules for the document.

The default behavior is that Windchill searches for the latest released version of the document to display. If no version of the document has been released, it displays the latest working version of the document.

To change the default behavior, change the value set for the **List of comma separated live cycle states preference** in the Preference Manager.

**Note:** The states must be valid life cycle states. The states are defined as key value pairs in StateRb.rbinfo and can be viewed in the life cycle template associated with the object. States are always specified using uppercase characters.

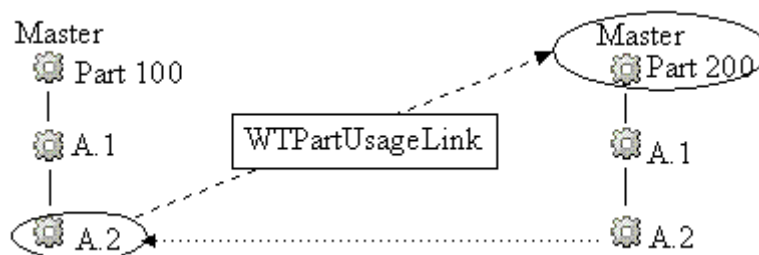
For example, to change the search to include the Released, Approved, and then Completed states of a document, set the preference value to:

RELEASED, APPROVED, COMPLETED

After this preference is set, Windchill searches for the latest Released version first. If none is found, it searches for the latest Approved version. If none is found, it searches for the latest Completed version. So, if a part is linked to Reference Document 4 which has three versions (A, B, and C), where A = Released, B = Approved, and C = In Work, based on the state settings in wt.properties, Windchill displays the latest iteration of version A which was released and ignores the rest.

## Part to Part Relationships (Projects Only)

A user associates one part with another by using the Product Structure Browser or the Product Structure Explorer. When a user makes the association, a WTPartUsageLink is created, forming a "uses part" relationship between a part and a part master.



The only configuration option for part to part relationships is described in the next section.

For additional information on part to part relationships, see the *Windchill ProjectLink User's Guide*.



## Revised or Saved Parent Part to Child Part

When a user revises a parent part using the **Revise** action or saves the part using the **Save As** action, the new version of the part carries forward the usage link.

To prevent a link from being copied forward on a **Revise** or **Save As** action, you must remove copy rules from wt.properties. For example, assume the following properties are set in wt.properties:

```
wt.enterprise.copyRuleDelimiter=,  
  
wt.enterprise.copyRulesN=wt.part.WTPart,Relationship,  
wt.part.WTPartUsageLink-uses
```

The first property sets the delimiter for the copy rules to the comma (,).

The wt.enterprise.copyRulesN property is the Relationship copy rule for wt.part.WTPart. This rule copies the references forward when the type of link is WTPartUsageLink.

If you remove the wt.enterprise.copyRulesN property, then the usage links are not carried forward.

**Note:** There can be no gaps in the sequence of copy rules. If you remove a copy rule, you must renumber those rules that follow. For example if there are six copy rules and you remove copyRules4, then you must renumber copyRules5 and copyRules6 so that copyRules5 becomes copyRules4 and copyRules6 becomes copyRules5.

Use the xconfmanager utility when modifying the wt.properties file. For information on using this utility, see [About the xconfmanager Utility](#). For additional information about the properties used for copy rules, see the description of wt.enterprise.copyRules in the properties.html file.



**Caution:** PTC recommends that you do not change the value of the wt.enterprise.copyServiceRules property. The property is used by internal services.

## Document to Document Relationships

There are several different types of document to document relationships in Windchill ProjectLink:

- A document can reference another document under **Related Objects > Documents** on the document information page. This relationship creates a link of type WTDocumentDependencyLink.
- A document can serve as the parent to a second child document under **Structure > Document Structure** on the document information page. This relationship creates a link of type WTDocumentUsageLink.

There are no configuration options available for document to document relationships.

For additional information on document to document relationships, see the *Windchill ProjectLink User's Guide*.

# 7

## Products and Libraries

This chapter provides an overview for administering product and libraries and describes the typical duties that an administrator does. It also provides additional information about some of the main administrative tasks for products and libraries.

Topic	Page
Overview .....	7-2
Typical Duties of Product and Library Administrators.....	7-2
Out-of-the-box Product and Library Context Templates .....	7-9
Creating a Product .....	7-42
Creating a Library .....	7-43
Administering Teams .....	7-45
Using the Product and Library Utilities Page.....	7-46
Configuring Products or Libraries for Dynamic Documents .....	7-46
Configuring Part and Document Relationships.....	7-48
Best Practices For Object Initialization Rules.....	7-53

## Overview

Product and library administrators (also known as product and library managers) are responsible for managing product and library contexts. The capabilities of product and library administrators are nearly identical.

Product and library administrators control the product and library configuration, and the membership in their product and library teams within the confines of a specific product or library application container. They control access to product and library information. They define the specific life cycles, templates and processes, and they monitor and manage the product and library activities.

Product application contexts are used to define new product models or instances and collect all the information associated with the product. Product contexts are defined by product creators who are authorized by the parent organization under which a product is created. Products inherit templates, roles, user-defined groups, and policies from their parent organization context. In addition, the administrator can define product-specific templates, roles, and policies.

Library application contexts are used to manage standard parts and documents that are used across products and projects in an organization. Library contexts are defined and managed by authorized library creators in the parent organization under which a library is created. Libraries inherit templates, roles, user-defined groups, and policies from their parent organization context. In addition, the administrator can define library-specific templates, user-defined groups, roles, and policies.

If Arbortext products are installed in your environment, product and library contexts can be used to manage dynamic documents and the published documents created from dynamic documents.

For general information about context contents and how to create contexts, see [Contexts](#).

## Typical Duties of Product and Library Administrators

Product and library administrators are responsible for managing the content of products and libraries. The capabilities of product and library administrators are nearly identical.

The typical duties of product and library administrators include the following:

- Managing team members and roles
- Managing folders
- Managing templates
- Managing object initialization rules
- Managing access policies
- Configuring numbering and versioning schemes

- Managing the life cycle of parts, documents, CAD documents, and dynamic documents
- Managing viewable publishing
- Managing preferences
- Undoing a user checkout
- Viewing and managing access control rules for objects
- Importing and exporting information
- Configuring external vaults or replication sites

The following sections describe some of the duties in more detail.

## Managing Team Members and Roles

Product and library managers can add members and team roles to a local team.

If the team includes a shared team, the shared team is managed from the organization context, and not the product or library context.

For more information about local and shared teams, see [Context Teams](#).

The team membership in a product or library can change or the current members can take on different responsibilities. To accommodate these types of changes, product and library managers have the ability to reassign tasks from one member to another.

## Establishing Roles

A product or library inherits the roles defined by its parent organization and the site. Additionally, roles can be defined in the context template used to create the product or library. You can then use these roles in the product or library, or you can add product or library-specific roles to product and library teams.

Each local team contains two fixed roles that cannot be removed: Product or Library Manager and Guest. The Guest role is designed to include user-defined groups and users that are not active team members and need only read access to product or library information.

It is possible to hide the Guest role for product and libraries. For more information, see [Hiding the Guest Role](#).

The products and libraries in which a user is a guest can be seen by selecting the **Guest Member** view from the **Full Product List** and **Full Library List** that is available from the **Product** and **Library** tabs.

You define the team members and roles for the product and library local teams that you manage. Each product team has a product manager group and each library team has a library manager group. Any individual that is a member of one of these managers groups has the rights to administer the product or library after it has been created. The creator of a library or product is automatically defined as a member of the product or library managers group, and is identified as the product or library owner, by default.

## Controlling the Visibility of Actions

As the product or library manager, you can control visibility to information and actions within the product or library based on the roles in the product or library local team.

For additional information, see [Restricting the Visibility of Actions in the User Interface through Teams](#).

## Overriding Profiles

As the product or library manager, you can override profile settings that are established at the site and organization contexts. For more information about profiles, see the [Profile Management](#) section in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

The site or organization administrator determines whether the various user interface elements and actions are configurable from an application context. You can override an action or user interface element in a context instance using Configure Actions for Roles which is accessed from the Team page. For additional information, see [Restricting the Visibility of Actions in the User Interface through Teams](#).

## Additional Product and Library Team Information

For additional team information, see [Administering Teams](#) later in this chapter.

## Managing Folders

You can define folders and links within products and libraries.

By default, only product or library managers can define folders and subfolders in a product or library. This is typically a good policy because it prevents members from adopting a multitude of folder organization models, thereby creating folder chaos.

By default, a folder inherits the domain of the parent folder. You can change this behavior as described in the [Assigning Domains to Folders in Solutions with Products and Libraries](#) section of the [Contexts](#) chapter.

## Managing Templates

You can define the document, CAD document, life cycle, team, and workflow templates that you want used in the context of a product or library. Each product or library inherits the templates defined by its parent organization context and the site. Additionally, you can create new product- or library-specific templates. If the name you specify is the name of an inherited template, then the new template overrides the inherited template that has the same name. You can restrict which templates are made available within your product or library by choosing to hide certain templates. For additional information about templates, see [Templates](#).

## Managing Object Initialization Rules

The use of object initialization rules provides a way to specify the following:

- Default values for the attributes of a specific object type. The default values are then used when a Windchill solution creates objects of that type and the user interface does not set a value. For example, you can set the default values for numbering and versioning.
- Display characteristics that constrain attribute values. If the user interface checks for the constraints, they are then used when the attribute value is displayed in the user interface. For example, the constraints for numbering, and versioning are honored in the user interface.

By default, the object initialization rules established by the site are inherited by each organization context and then inherited by the product or library. However, they can be overridden by an organization or overridden in a product or library. The rules are defined in an XML format and can be viewed and edited by a product or library manager. For PTC recommendations in this area, see [Best Practices For Object Initialization Rules](#), later in this chapter.

For additional information about object initialization rules, see the [Object Initialization Rules](#) chapter.

## Viewing and Managing Access Policies

You can view and manage the access control rules for objects (such as folders, parts, and documents) in your product or library. Navigate to the object, and select the **Manage Security** action. From the **Manage Security** window you can view and set the access control to the object for roles within your product or library team.

For more information, see [Creating and Managing Access Control Rules](#) in the Access Control chapter, and the online help available from the **Manage Security** window.

You can define policies that control the level of access to information in a product or library. When defining a policy, the object types and user-defined groups defined in the parent organization can be used as well as the system groups representing the team roles in a product or library. For example, you could create a policy that provides read access to all documents of type “Quality Assessment” to the product team role/group called “Testers”. You can also choose to extend read access to this document type to an organizational user-defined group with the name “Quality Assurance” (if such a policy is not already granted at the parent organization level).

For general information about domains and policies, see [Administering Domains and Policies](#).

For additional information about creating or updating access control rule policies, see the [Access Control](#) chapter.

## Configuring Numbering and Versioning Schemes

You can configure the number and versioning scheme used to uniquely identify parts, documents and other objects in the product or library. The numbering and versioning schemes defined at the site and organization level are inherited by products and libraries by default, but each product or library can optionally define its own schemes. Use the Object Initialization Rules Administrator to configure the number and versioning schemes.

In general, when a company is hosting Windchill for its internal use, the numbering and versioning schemes should all be defined in the Site context and should not be overridden by an organization or in a product or library. The Product Design template demonstrates how to set up versioning (and other required functionality) for use in a basic life cycle management environment, where multiple life cycle templates are in use and state-based versioning is set up. If you choose to use this type of environment, consider defining the versioning (and other initialization rules) in the Site context rather than in individual product or library contexts so that all of your products and libraries use the same set of rules.

For additional information, see [Administering the Numbering of Parts, Documents, CAD Documents, and Change Objects](#) and [Administering the Versioning of Parts, Documents, and CAD Documents](#).



## Managing the Life Cycle of Parts, Documents, CAD Documents, and Dynamic Documents

Within your Windchill solution, you have the choice of using a variety of out-of-the-box life cycle templates to help manage the life cycle of parts, documents, CAD documents, and dynamic documents. You can also create your own life cycle template or modify one of the existing templates. The options available to you range from the ability to fully automate the process of promoting a business object through maturity levels with the use of workflow processes to providing manual transition process where users in specified roles can promote objects to new states explicitly through the promote action. Default life cycle templates are established for object types through the use of object initialization rules and a default set of Site rules is loaded during the installation process.

For additional information, see the [Object Initialization Rules](#) and the [Life Cycles](#) chapters.

## Managing Viewable Publishing

You can monitor and manage the publishing of viewable files that are optionally generated when CAD models are checked into products and libraries. You can also configure and update the watermarks used by the visualization tool when viewing document and part content from the product or library.

For additional information, see the [Visualization Services](#) chapter.

## Managing Preferences

Using the Preference Manager on the **Product** or **Library** tab **Utilities** page, you can manage preferences at the product or library level. Preferences set at the product or library level affect the product or library for which they are set. Different preferences can be set for different products or libraries.

For example, a number of preferences are available which allow you to determine the types of content allowed for documents. These preferences are found in the **Attachments** category of the Preference Manager.

For more information on the Preference Manager, see [Administering Preferences](#).

## Undoing a User Checkout

If a user in your product or library has an object checked out, you can undo the checkout, reverting to the last modified version or iteration of the object. This is useful in situations where the user who had the object checked out has left the company, or if they are out of the office when the checked out object is needed.

If you already know the object or objects for which you want to undo the checkout, you can perform the undo checkout action on an object from either the **Folders** page of your product or library, or from the information page for the checked out object. You can also find objects checked out by a particular user, by searching using the **Checked Out By** criteria on the **Advanced Search** page, then performing the undo checkout action for the object from the **Search Results** table.

## Importing and Exporting Information

You can import information into a product or library, and export information to a local file system. The import/export facilities support information exchange with another Windchill server or non-Windchill system. These utilities read and write system information in an XML format. For general import and export information, see the Windchill Import and Export chapter in the *Windchill System Administrator's Guide*.

The workflow and life cycle administration utilities accessible to the site and organization administrator integrate the import and export functions with administering workflows and life cycles.

For life cycle import and export information, see [Importing and Exporting Life Cycle Templates](#) in the Administering Life Cycles chapter.

For workflow import and export information, see [Importing and Exporting Workflow Templates](#) in the Administering Workflows chapter.

## Configuring External Vaults or Replication Sites to Optimize Performance

You can configure the vaulting and replication rules for the external file and replica sites established by the site administrator.

When external vaults are configured, document and part content is stored on a file system rather than in the database. This configuration can provide significant upload performance improvements and is appropriate when the site is frequently used to exchange large files (such as CAD model files).

By default, external storage rules are based on individual domains in each context. You can launch the **External Storage Administrator** from the **Utilities** page in the context of a product or library. The utility allows you to create vaulting rules for the domains - /System and /Default - associated with the context in which the client is launched.

**Note:** The rules needed for setting up an external file vault for a product or library are not inherited from the site.

If the increased number of vaults and file vault rules becomes unmanageable, the site administrator can force vaulting to be accomplished through a single vault by setting the `wt.fv.forceContentToVault` property to true. For how to set external file vault rules or set up a single vault, see the *Windchill System Administrator's Guide*.

Site administrators typically define the replica site and the replication schedule, and the product or library managers configure the replication rules for a particular product or library. Product and library managers can access the **File Server Administrator** utility, but can only work on setting the replication and file vaulting rules. For how to set replication rules, see the *Windchill System Administrator's Guide*.

## Tracking Changes

You can enable functionality allowing creations of new change object revisions, and display this information on the change object's revision history. Change objects include: problem reports, change requests, change notices, and variances.

To enable this functionality, set the value of the **Change Modification Tracking** preference.

For more information, see the *Windchill PDMLink User's Guide*.

## Out-of-the-box Product and Library Context Templates

When Windchill PDMLink is installed, the following product and library templates are loaded:

- **Product Design** -- This template provides a demonstration of how to setup a product container for use with the One Phase Development and Two Phase Development life cycle templates and state-based versioning. As described in detail later in this section, the default set of roles defined are Members, Promotion Reviewers, Promotion Approvers, Collaboration Manager, Variance Approvers, Marketing, Procurement Engineer, Designer, Quality Engineer, Manufacturing Engineer, Design Team Leader, Change Review Board, Change Admin I, Change Admin II, and Change Admin III. Also, the Guests and Product Manager roles are created automatically. Some basic information about these roles is as follows:
  - The Members role is used as a basic role to grant container team membership. A set of access rules are defined for confirmed members of the container team.
  - The Designer and Design Team Leader roles are unique to this template and identify the people who are responsible for creating and maintaining the business objects used to design a product.
  - Additionally, there are marketing and engineering roles that are unique to this template that can be used to delineate those functions in the design process.

- Access policies defined for Change Objects have been setup to work in conjunction with the closed-loop change process.
- Members of the Guest role are granted Read access to all objects.
- General Product and General Library -- These templates provide examples of how to setup basic access control for a general product or library container. As described in detail later in this section, the default set of roles defined are Members, Promotion Reviewers, Collaboration Manager, Variance Approvers, Promotion Approvers, Change Review Board, Change Admin I, Change Admin II, and Change Admin III. Also, the Guests and Product/Library Manager roles are created automatically. Some basic information about these roles is as follows:
  - The Members role is used as a basic role to grant container team membership. A set of access rules are defined for confirmed members of the container team.
  - Access policies defined for Change Objects have been setup to work in conjunction with the closed-loop change process.
  - Members of the Guest role are granted Read access to all objects.
- Part Library -- This template provides an example of how you can define a parts library. The default set of roles defined are Members, Change Admin I, Change Admin II, and Change Admin III. Members of the Guest role are granted Read access to all objects.
- Document Library -- This template provides an example of how you can define a document library. The default set of roles defined are Members, Change Admin I, Change Admin II, and Change Admin III. Members of the Guest role are granted Read access to all objects.

The product and library templates can define the same basic information that is discussed in the [Context Administrative Items](#) section of the Administering Containers chapter. If you want to create a new product or library template, see [Creating Context Templates](#).

When users are working primarily with dynamic documents, you can start with the context template that is closest to your business model and modify the product or library that you create to meet your needs. For some general guidelines on administrative activities that you should consider doing, see [Configuring Products or Libraries for Dynamic Documents](#).

## Product Design Template

The Product Design out-of-the-box template defines the following container administrative items:

- Promotion Requests subfolder in the Default domain
- Container participation
- Container access control policies
- Object initialization rules for the following objects:
  - wt.maturity.PromotionNotice
  - wt.part.WTPart
  - wt.epm.EPMDocument
  - wt.doc.WTDocument

Through the object initialization rules, the following life cycle templates are set as defaults in the Design Product template:

- Two Phase Development life cycle template - default template for parts and functional specifications. It has the following states:



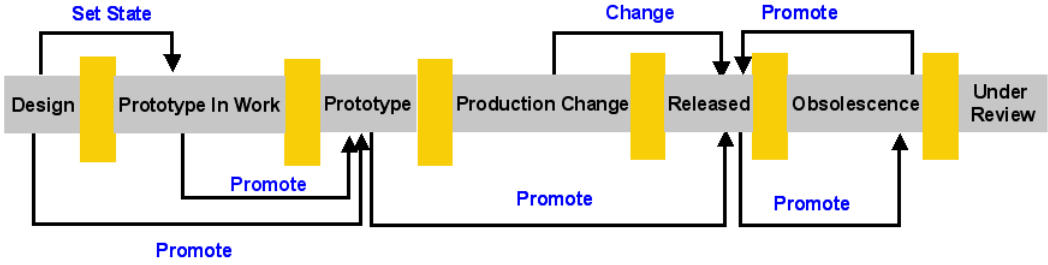
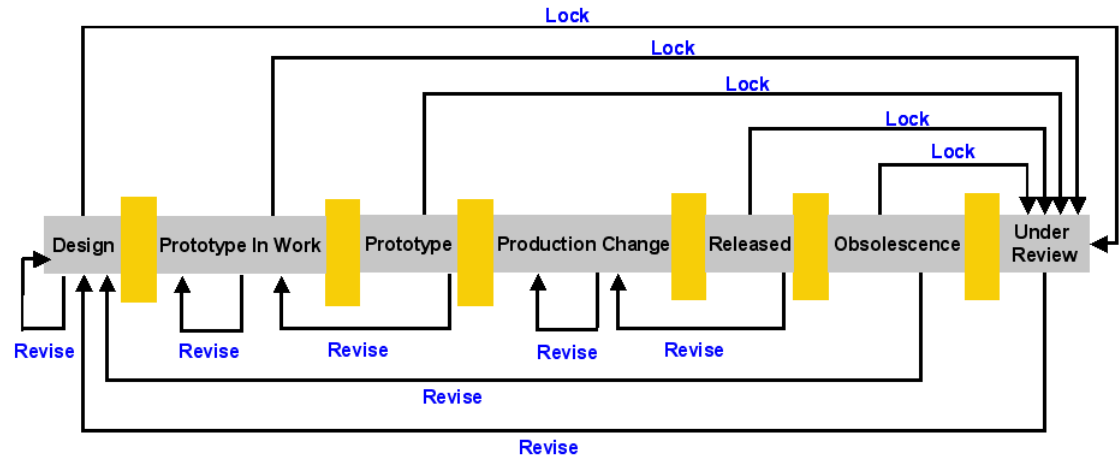
- One Phase Development life cycle template - default template for documents and CAD documents. It has the following states:



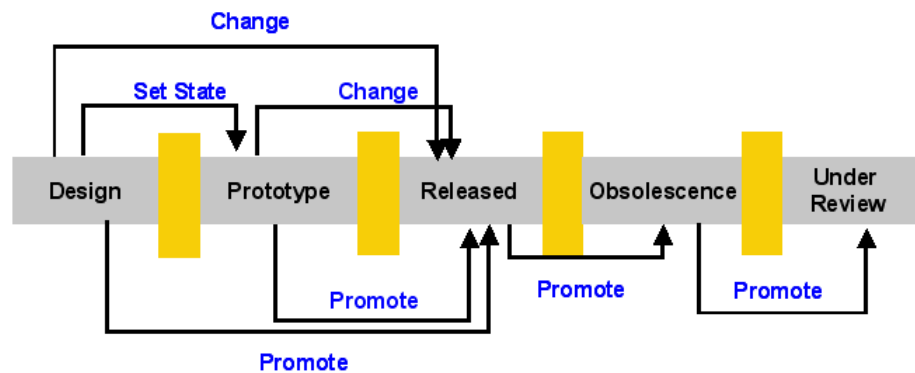
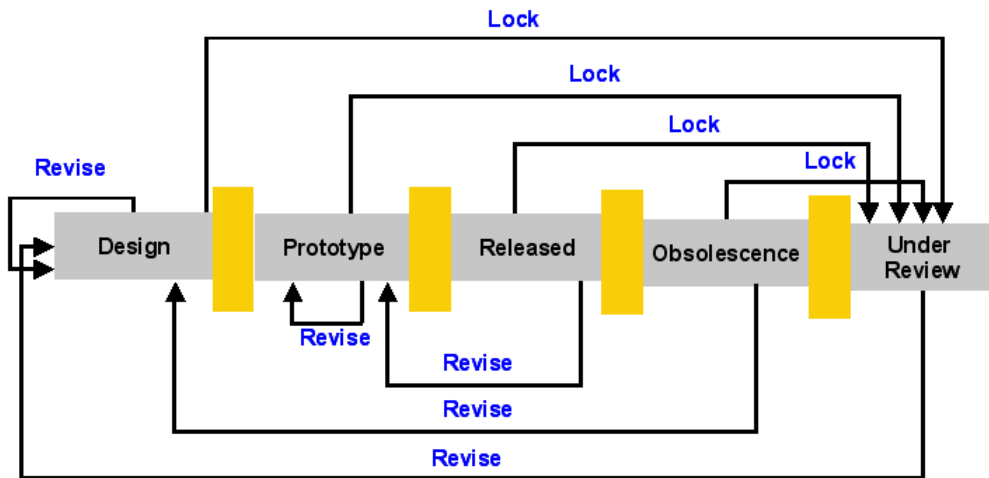
**Note:** Although you can use multiple life cycle templates within a product container as is done with the demonstration Product Design template, PTC recommends that you become familiar with the functionality available through these life cycle templates and select one of the out-of-the-box templates (or create your own template) to use for all life-cycle managed objects in a product container.

The Two Phase Development and One Phase Development life cycle templates have the following transitions defined: Lock, Revise, Promote, Change, and Set State. Each template establishes different transition rules out of the box.

The following diagrams show the out-of-the-box transition rules for the Two Phase Development life cycle:



The following diagrams show the out-of-the-box transition rules for the One Phase Development life cycle:



For additional information about modifying or creating life cycle templates, see the [Life Cycles](#) chapter. The out-of-the-box workflows that are used with the default life cycle templates are:

- Promotion Request Approval Process
- Promotion Request Review Process

For additional information about these workflows, see [Promotion Process](#) in the Life Cycles chapter.

The following sections describe the items that are defined in the Product Design template.

## Out-of-the-box Subfolder for wt.maturity.PromotionNotice Objects

In the Default cabinet and Default domain, the Product Design template creates the Promotion Requests subfolder. This folder is then defined as the default folder for the wt.maturity.PromotionNotice object type. Creating and using this folder is not required, but provides a useful folder for promotion requests.

## Out-of-the-box Context Participation

The Product Design template defines roles that are automatically included in product teams. These roles are the roles that automatically appear on the **Members** table. Additionally, roles available through the organization context can be added manually to the team.

The following roles are automatically included in the team:

- Product Manager
- Guest, which is the role to use for users who you do not want as a member of the team, but you do want them to have limited access to the product. Users who are guests will not see the product on their **Product List**. They would need to search for the product to locate it. Out of the box, guests have only read access to product data.

You can allow guests to see products and libraries on their product and library lists by adding the following property and value to the wt.properties file:

```
wt.inf.team.displayContainerListForGuest=true
```

**Note:** Use the -s parameter on the xconfmanager command to add Windchill properties. See [About the xconfmanager Utility](#) for details. This property controls the guest visibility on project and program lists as well as product and library lists.

The out-of-the-box Product Design template adds the following roles to the team:

- Members, which is a role that can be used for team members who do not belong in other roles on the team.
- Change Admin I, which is the change administrator I change management role.
- Change Admin II, which is the change administrator II change management role.
- Change Admin III, which is the change administrator III change management role.
- Promotion Reviewers, which are the team members for reviewing promotion requests.
- Promotion Approvers, which are the team members for approving promotion requests.



- Change Review Board, which are the team members of the Change Review Board.
- Design Team Leader, which is the Design Team Leader role. This role is unique to this template.
- Designer, which is the role used to create, edit and manage business objects within the system. This role is unique to this template.
- Manufacturing Engineer, which is the Workflow Manufacturing Engineer role. This role is unique to this template.
- Marketing, which is the Marketing role. This role is unique to this template.
- Procurement Engineer, which is the Procurement Engineer role. This role is unique to this template.
- Quality Engineer, which is the Workflow Quality Engineer role. This role is unique to this template.
- Collaboration Manager
- Variance Approvers

## Out-of-the-box Context Access Control Policies

When you create a product context using the out-of-the-box Product Design template, domain-based access control rules are defined in the System and Default domains of a product context. The rules that are created use the system groups that correspond to the out-of-the-box roles established for the team.

The following section lists the roles and their corresponding groups. Additional sections provide the out-of-the-box rules generated that reference the groups listed in the next section. All rules are generated as a result of administrative policy items that are in the out-of-the-box template.



**Caution:** PTC recommends that you use caution when modifying the access control rules that are set when contexts are created. Modifying rules can cause problems with access to data and administrative functionality.

**Note:** There are no out-of-the-box rules set for the Members or Change Admin III roles.

## Team Roles and Groups

The following table lists the out-of-the-box Product Design template team roles and system groups, and identifies which group that is associated with each role:

Team Role	Corresponding System Group
Change Admin I	CHANGE ADMINISTRATOR I
Change Admin II	CHANGE ADMINISTRATOR II
Change Admin III	CHANGE ADMINISTRATOR III
Change Review Board	CHANGE REQUEST REVIEW BOARD
Promotion Approvers	PROMOTION APPROVERS
Promotion Reviewers	PROMOTION REVIEWERS
Design Team Leader	DESIGN TEAM LEADER
Designer	DESIGNER
Manufacturing Engineer	MANUFACTURING ENGINEER
Marketing	MARKETING
Procurement Engineer	PROCUREMENT ENGINEER
Quality Engineer	QUALITY ENGINEER
Guests	GUEST
Members	MEMBERS
Product Manager	PRODUCT MANAGER
Collaboration Manager	COLLABORATION MANAGER
Variance Approvers	VARIANCE APPROVERS

**Note:** As shown in the Generation Method column, the rules for Cabinet and SubFolder object types are defined programmatically when a product context is created; they are not defined through the template that is used. All other rules are defined through the template.

## Rules for the GUEST Group

Out-of-the-box Product Design access control rules for the GUEST group (Guest role) are described in the following sections.

### Default Domain Rules for the GUEST Group

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for guests of a product:

Object Type	State	Permissions
Cabinet	All	Read
SubFolder	All	Read
WTOBJECT	All	Read

### System Domain Rules for the GUEST Group

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the guests of a product or library:

Object Type	State	Permissions
WTOBJECT	All	Read

## Rules in Default Domain for the MARKETING Group

Out-of-the-box Product Design access control rules that are defined for the MARKETING group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the MARKETING group:

Object Type	State	Permissions
EPMDocument	Prototype In Work	Read and Download
EPMDocument	Production Change	Read and Download
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context

Object Type	State	Permissions
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise
WTPart	Prototype In Work	Read and Download
WTPart	Prototype	Read and Download
WTPart	Production Change	Read and Download
WTPart	Released	Read and Download
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the MARKETING group.

#### Rules in Default Domain for the PROCUREMENT ENGINEER Group

Out-of-the-box Product Design access control rules that are defined for the PROCUREMENT ENGINEER group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the PROCUREMENT ENGINEER group:

Object Type	State	Permissions
EPMDocument	Prototype In Work	Read and Download
EPMDocument	Production Change	Read and Download
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context

Object Type	State	Permissions
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise
WTPart	Prototype In Work	Read and Download
WTPart	Prototype	Read and Download
WTPart	Production Change	Read and Download
WTPart	Released	Read and Download
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the PROCUREMENT ENGINEER group.

#### Rules in Default Domain for the QUALITY ENGINEER Group

Out-of-the-box Product Design access control rules that are defined for the QUALITY ENGINEER group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the QUALITY ENGINEER group:

Object Type	State	Permissions
EPMDocument	Prototype In Work	Read and Download
EPMDocument	Production Change	Read and Download
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context

Object Type	State	Permissions
WTDocument	Released	Revise
WTPart	Prototype In Work	Read and Download
WTPart	Prototype	Read and Download
WTPart	Production Change	Read and Download
WTPart	Released	Read and Download
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the QUALITY ENGINEER group.

#### Rules in Default Domain for the DESIGNER Group

Out-of-the-box Product Design access control rules that are defined for the DESIGNER group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the DESIGNER group:

Object Type	State	Permissions
EPMDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
EPMDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Released	Read, Revise, and Download
EPMDocument	Obsolescence	Read and Download
EPMDocument	Under Review	Read and Download

Object Type	State	Permissions
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context, and Set State
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise
WTPart	Design	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
WTPart	Prototype In Work	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Prototype	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Production Change	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Released	Read, Revise, New View Version, and Download
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the DESIGNER group.

## Rules in Default Domain for the MANUFACTURING ENGINEER Group

Out-of-the-box Product Design access control rules that are defined for the MANUFACTURING ENGINEER group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the MANUFACTURING ENGINEER group:

Object Type	State	Permissions
EPMDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
EPMDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Released	Read, Revise, and Download
EPMDocument	Obsolescence	Read and Download
EPMDocument	Under Review	Read and Download
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise



Object Type	State	Permissions
WTPart	Design	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
WTPart	Prototype In Work	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Prototype	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Production Change	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Released	Read, Revise, New View Version, and Download
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the MANUFACTURING ENGINEER group.

#### Rules in Default Domain for the DESIGN TEAM LEADER Group

Out-of-the-box Product Design access control rules that are defined for the DESIGN TEAM LEADER group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the DESIGN TEAM LEADER group:

Object Type	State	Permissions
EPMDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
EPMDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context

<b>Object Type</b>	<b>State</b>	<b>Permissions</b>
EPMDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Released	Read, Revise, and Download
EPMDocument	Obsolescence	Read and Download
EPMDocument	Under Review	Read and Download
WTDocument	Design	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
WTDocument	Prototype In Work	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Prototype	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Production Change	Read, Modify, Create, Delete, Revise, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise
WTPart	Design	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, Change Context, and Set State
WTPart	Prototype In Work	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Prototype	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, and Change Context
WTPart	Production Change	Read, Modify, Create, Delete, Revise, New View Version, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Released	Read, Revise, New View Version, and Download

Object Type	State	Permissions
WTPart	Obsolescence	Read and Download
WTPart	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the DESIGN TEAM LEADER group.

#### Rules in Default Domain for PROMOTION REVIEWERS Group

Out-of-the-box Product Design access control rule for the PROMOTION REVIEWERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the PROMOTION REVIEWERS group.

#### Rules in Default Domain for CHANGE REQUEST REVIEW BOARD Group

Out-of-the-box Product Design access control rule for the CHANGE REQUEST REVIEW BOARD group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeRequest2	Under Review	Read and Download
WTChangeOrder2	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the CHANGE REQUEST REVIEW BOARD group.

### Rules in Default Domain for PROMOTION APPROVERS Group

Out-of-the-box Product Design access control rule for the PROMOTION APPROVERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the PROMOTION APPROVERS group.

### Rules for PRODUCT MANAGER Group

Out-of-the-box Product Design access control rules for the PRODUCT MANAGER group (Product Manager role) are described in the following sections.

#### Default Domain Rule for PRODUCT MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the Product Manager:

Object Type	State	Permissions
WTOBJECT	All	Full control (All)

#### System Domain Rule for PRODUCT MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the Product Manager:

Object Type	State	Permissions
WTOBJECT	All	Full control (All)

### Rules in Default Domain for CHANGE ADMINISTRATOR I Group

Out-of-the-box Product Design access control rules for the CHANGE ADMINISTRATOR I group (Change Admin I role) are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeActivity2	All	Full Control (All)
WTChangeIssue	All	Full Control (All)

Object Type	State	Permissions
WTChangeOrder2	All	Full Control (All)
WTChangeProposal	All	Full Control (All)
WTChangeRequest2	All	Full Control (All)
WTVariance	All	Full Control (All)

**Note:** No rules are defined in the System domain for the CHANGE ADMINISTRATOR I group.

#### Rules in Default Domain for CHANGE ADMINISTRATOR II Group

Out-of-the-box Product Design access control rules for the CHANGE ADMINISTRATOR II group (Change Admin II role) are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeActivity2	All	Full Control (All)
WTChangeOrder2	All	Full Control (All)
WTChangeProposal	All	Read

**Note:** No rules are defined in the System domain for the CHANGE ADMINISTRATOR II group.

#### Rules in Default Domain for TEAMMEMBERS Group

Out-of-the-box Product Design access control rule for the TEAMMEMBERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTPart	All	Read and Download
WTPart	Prototype	Read and Download
WTPart	Released	Read and Download
WTPart	Obsolescence	Read and Download
WTPart	UnderReview	Read and Download

<b>Object Type</b>	<b>State</b>	<b>Permissions</b>
WTPartSubstituteLink	All	Read, Modify, Create, Delete, Change Domain, and Create by Move
WTPartAlternateLink	All	Read, Modify, Create, Delete, Change Domain, and Create by Move
WTDocument	All	Read and Download
WTDocument	Prototype	Read and Download
WTDocument	Released	Read and Download
WTDocument	Obsolescence	Read and Download
WTDocument	UnderReview	Read and Download
EPMDocument	All	Read and Download
EPMDocument	Prototype	Read and Download
EPMDocument	Released	Read and Download
EPMDocument	Obsolescence	Read and Download
EPMDocument	UnderReview	Read and Download
ManagedBaseline	All	Read
Managed Baseline	InWork	Read, Modify, Create, Delete, Change Domain, Create by Move, Change Context, and Modify Identity
StructuredAnnotationSet	All	Read
StructuredAnnotationSet	InWork	Read, Modify, Create, Delete, Change Domain, Create by Move, and Change Context
WTProductConfiguration	All	Read
WTProductConfiguration	InWork	Read, Modify, Create, Delete, Change Domain, Create by Move, and Change Context
WTProductInstance2	All	Read
WTProductInstance2	InWork	Read, Modify, Create, Delete, Change Domain, Create by Move, and Change Context

<b>Object Type</b>	<b>State</b>	<b>Permissions</b>
WTProductInstance2	Released	Read and Revise
WTChangeIssue	All	Read and Download
WTChangeIssue	Open	Read, Modify, Create, Download, Modify Content, and Create by Move
WTChangeRequest2	All	Read and Download
WTChangeRequest2	Open	Read, Modify, Create, Download, Modify Content, and Create by Move
WTChangeProposal	Open	Read, Modify, Create, Download, Modify Content, and Create by Move
WTChangeOrder2	All	Read and Download
WTChangeOrder2	Open	Read, Modify, Download, and Modify Content
WTChangeOrder2	Implementation	Read, Modify, Download, and Modify Content
WTChangeActivity2	All	Read and Download
WTChangeActivity2	Open	Read, Download, and Modify
WTChangeActivity2	Implementation	Read, Download, and Modify
PromotionNotice	All	Read and Download
PromotionNotice	Open	Read, Modify, Create, Delete, Download, Modify Content, Change Domain, Create by Move, and Change Context
WTVariance	All	Read and Download
WTVariance	Open	Read, Modify, Create, Download, Modify Content, Create by Move, and Modify Identity
WTObject	All	Read
WfExecutionObject	All	Read, Modify, Create, Download, and Modify Content
Cabinet	All	Read and Modify
SubFolder	All	Read and Modify

### Rules in Default Domain for COLLABORATION MANAGER Group

Out-of-the-box Product Design access control rule for the COLLABORATION MANAGER group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTDocument	All	Change Permissions
WTPart	All	Change Permissions
EPMDocument	All	Change Permissions
WTChangeIssue	All	Change Permissions
WTChangeProposal	All	Change Permissions
WTChangeRequest2	All	Change Permissions
WTChangeOrder2	All	Change Permissions
WTChangeActivity2	All	Change Permissions
WTVariance	All	Change Permissions
VariantSpec	All	Change Permissions
SubFolder	All	Change Permissions

**Note:** No rules are defined in the System domain for the COLLABORATION MANAGER group.

### Rules in Default Domain for VARIANCE APPROVERS Group

Out-of-the-box Product Design access control rule for the VARIANCE APPROVERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTVariance	UnderReview	Read and Download

**Note:** No rules are defined in the System domain for the VARIANCE APPROVERS group.



## Rules for SHARED TEAM MANAGER Group

Out-of-the-box Product Design access control rules for the SHARED TEAM MANAGER group (Shared Team Manager role) are described in the following sections.

### Default Domain Rule for SHARED TEAM MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the Shared Team Manager:

Object Type	State	Permissions
WTOBJECT	All	Full control (All)

### System Domain Rule for SHARED TEAM MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the Shared Team Manager:

Object Type	State	Permissions
WTOBJECT	All	Full control (All)

## Rules in Default Domain for OWNER

OWNER is a pseudo-user that is used by Windchill to represent the object owner. Out-of-the-box Product Design access control rule for the OWNER group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	Open	Read, Delete, Download, Change Domain, and Change Context
PromotionNotice	UnderReview	Read, Delete, Download, Change Domain, and Change Context
PromotionNotice	Rejected	Read, Delete, Download, Change Domain, and Change Context

## Out-of-the-box Object Initialization Rules

The Product Design template sets object initialization rules for the following object types:

- wt.maturity.PromotionNotice
- wt.part.WTPart
- wt.epm.EPMDocument
- wt.doc.WTDocument

The following table lists the object types, attributes, and default values set by the object initialization rules that are in the template:

Object Type	Attribute	Default Value
PromotionNotice	folder.id	/Default/Promotion Requests
WTPart	folder.id	/Default
WTPart	lifeCycle.id	Two Phase Development
WTPart	teamTemaplate.id	Default
WTPart	MBA versionInfo	wt.series.HarvardSeries.StateBased
EPMDocument	folder.id	/Default
EPMDocument	lifeCycle.id	One Phase Development
EPMDocument	teamTemaplate.id	Default
EPMDocument	MBA versionInfo	wt.series.HarvardSeries.StateBased
WTDocument	folder.id	/Default
WTDocument	lifeCycle.id	One Phase Development
WTDocument	teamTemaplate.id	Default
WTDocument	MBA versionInfo	wt.series.HarvardSeries.StateBased

**Note:** Setting object initialization rules in a product template means that the rules are different than those set for the organization or site. This is the case with this template and the out-of-the-box object initialization rules that are set in the site container. PTC recommends that you only use this template as a demonstration template and consider setting those rules that you want to use for parts, documents, and CAD documents in either the organization or site context.

For additional information about object initialization rules, see [Object Initialization Rules](#).

For additional information about state-based versioning, see [Changing Versioning Schemes](#).

For the details on the XML tag used to set object initialization rules in a template, see [Setting File-based or State-based Versioning for Objects](#).

## General Product and General Library Templates

The General Product and General Library out-of-the-box templates define only the following context administrative items:

- Context participation
- Context access control policies

The following sections describe the items that are defined in these templates.

### Out-of-the-box Container Participation

The General Product and General Library templates define roles that are automatically included in product and library teams. These roles are the roles that automatically appear on the **Members** table. Additionally, roles available through the organization context can be added to the team.

The following roles are automatically included in the team:

- Product Manager (for products) and Library Manager (for libraries)
- Guest, which is the role to use for users who you do not want as a member of the team, but you do want them to have limited access to the product or library. Users who are guests will not see the product or library on their Product List or Library List. They would need to search for the product or library to locate it. Out of the box, guests have only read access to product or library data.

The out-of-the-box General Product and General Library templates add the following roles to the team:

- Members, which is a role that can be used for team members who do not belong in other roles on the team.
- Change Admin I, which is the change administrator I change management role
- Change Admin II, which is the change administrator II change management role
- Change Admin III, which is the change administrator III change management role
- Promotion Reviewers, which are the team members for reviewing Promotion Requests

- Promotion Approvers, which are the team members for approving Promotion Requests
- Change Review Board, which are the team members of the Change Review Board
- Collaboration Member
- Variance Approvers

## Out-of-the-box Context Access Control Policies

When you create a product or library context using the out-of-the-box General Product and General Library templates, domain-based access control rules are defined in the System and Default domains of a product or library context. The rules that are created reference the groups that correspond to the out-of-the-box roles established for the team.

The following section lists the roles and their corresponding groups. Additional sections provide the out-of-the-box rules that reference the groups listed in the next section. All rules are generated as a result of administrative policy items that are in the out-of-the-box templates.



**Caution:** PTC recommends that you use caution when modifying the access control rules that are set when contexts are created. Modifying rules can cause problems with access to data and administrative functionality.

**Note:** There are no out-of-the-box rules set for the Members or Change Admin III roles.

## Team Roles and Groups

The following table lists the out-of-the-box General Product and General Library team roles and groups, and identifies which group that is associated with each role:

Team Role	Corresponding Group
Change Admin I	CHANGE ADMINISTRATOR I
Change Admin II	CHANGE ADMINISTRATOR II
Change Admin III	CHANGE ADMINISTRATOR III
Change Review Board	CHANGE REQUEST REVIEW BOARD
Promotion Approvers	PROMOTION APPROVERS
Promotion Reviewers	PROMOTION REVIEWERS
Guests	GUEST

<b>Team Role</b>	<b>Corresponding Group</b>
Members	MEMBERS
Product Manager (products only)	PRODUCT MANAGER
Library Manager (libraries only)	LIBRARY MANAGER
Collaboration Manager	COLLABORATION MANAGER
Variance Approvers	VARIANCE APPROVERS

### **Rules for the GUEST Group**

Out-of-the-box General Product and General Library access control rules for the GUEST group (Guest role) are described in the following sections.

#### **Default Domain Rules for the GUEST Group**

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for guests of a product or library:

<b>Object Type</b>	<b>State</b>	<b>Permissions</b>
Cabinet	All	Read
SubFolder	All	Read
WTOBJECT	All	Read

#### **System Domain Rules for the GUEST Group**

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the guests of a product or library:

<b>Object Type</b>	<b>State</b>	<b>Permissions</b>
WTOBJECT	All	Read

### Rules in Default Domain for CHANGE REQUEST REVIEW BOARD Group

Out-of-the-box General Product and General Library access control rules for the CHANGE REQUEST REVIEW BOARD group are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeRequest2	Under Review	Read and Download
WTChangeOrder2	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the CHANGE REQUEST REVIEW BOARD group.

### Rules in Default Domain for PROMOTION APPROVERS Group

Out-of-the-box General Product and General Library access control rule for the PROMOTION APPROVERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the PROMOTION APPROVERS group.

### Rules in Default Domain for PROMOTION REVIEWERS Group

Out-of-the-box General Product and General Library access control rule for the PROMOTION REVIEWERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	Under Review	Read and Download

**Note:** No rules are defined in the System domain for the PROMOTION REVIEWERS group.

### Rules for PRODUCT MANAGER and LIBRARY MANAGER Groups

Out-of-the-box General Product and General Library access control rules for the PRODUCT MANAGER group (Product Manager role) and LIBRARY MANAGER group (Library Manager role) are described in the following sections.

#### Default Domain Rule for PRODUCT MANAGER and LIBRARY MANAGER Groups

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the Product Manager or Library Manager:

Object Type	State	Permissions
WTObject	All	Full control (All)

#### System Domain Rule for PRODUCT MANAGER and LIBRARY MANAGER Groups

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the Product Manager or Library Manager:

Object Type	State	Permissions
WTObject	All	Full control (All)

#### Rules in Default Domain for CHANGE ADMINISTRATOR I Group

Out-of-the-box General Product and General Library access control rules for the CHANGE ADMINISTRATOR I group (Change Admin I role) are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeActivity2	All	Full Control (All)
WTChangeIssue	All	Full Control (All)
WTChangeOrder2	All	Full Control (All)
WTChangeProposal	All	Full Control (All)
WTChangeRequest2	All	Full Control (All)
WTVariance	All	Full Control (All)

**Note:** No rules are defined in the System domain for the CHANGE ADMINISTRATOR I group.

### Rules in Default Domain for CHANGE ADMINISTRATOR II Group

Out-of-the-box General Product and General Library access control rules for the CHANGE ADMINISTRATOR II group (Change Admin II role) are described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTChangeActivity2	All	Full Control (All)
WTChangeOrder2	All	Full Control (All)
WTChangeProposal	All	Read

**Note:** No rules are defined in the System domain for the CHANGE ADMINISTRATOR II group.

### Rules in Default Domain for TEAMMEMBERS Group

Out-of-the-box General Product and General Library access control rule for the TEAMMEMBERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTDocument	All	Read and Download
WTDocument	InWork	Modify, Create, Delete, Modify Content, Change Domain, Create by Move, and Change Context
WTDocument	Released	Revise
WTPart	All	Read and Download
WTPart	In Work	Modify, Create, Delete, Modify Content, Change Domain, Create by Move, and Change Context
WTPart	Released	Revise and New View Version
WTPartSubstituteLink	All	Read, Modify, Create, and Delete
WTPartAlternateLink	All	Read, Modify, Create, and Delete
EPMDocument	All	Read and Download
EPMDocument	InWork	Modify, Create, Delete, Modify Content, Change Domain, Create by Move, and Change Context
EPMDocument	Released	Revise



Object Type	State	Permissions
ManagedBaseline	All	Read
ManagedBaseline	InWork	Modify, Create, Delete, Change Domain, Create by Move, and Change Context
StructuredAnnotationSet	All	Read
StructuredAnnotationSet	InWork	Modify, Create, and Delete
WTProductConfiguration	All	Read
WTProductConfiguration	InWork	Modify, Create, and Delete
WTProductInstance2	All	Read
WTProductInstance2	InWork	Modify, Create, and Delete
WTProductInstance2	Released	Revise
WTChangeIssue	All	Read and Download
WTChangeIssue	Open	Modify and Create
WTChangeRequest2	All	Read and Download
WTChangeRequest2	Open	Modify and Create
WTChangeProposal	Open	Modify and Create
WTChangeOrder2	All	Read and Download
WTChangeOrder2	Open	Modify
WTChangeOrder2	Implementation	Modify
WTChangeActivity2	All	Read and Download
WTChangeActivity2	Open	Modify
WTChangeActivity2	Implementation	Modify
PromotionNotice	All	Read and Download
PromotionNotice	Open	Modify and Create
WTVariance	All	Read and Download
WTVariance	Open	Modify and Create
WTObject	All	Read

Object Type	State	Permissions
WfExecutionObject	All	Read, Modify, Create, Download, and Modify Content
Cabinet	All	Read and Modify
SubFolder	All	Read and Modify

**Note:** No rules are defined in the System domain for the TEAMMEMBERS group.

#### Rules in Default Domain for COLLABORATION MANAGER Group

Out-of-the-box Product Design access control rule for the COLLABORATION MANAGER group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTDocument	All	Change Permissions
WTPart	All	Change Permissions
EPMDocument	All	Change Permissions
WTChangeIssue	All	Change Permissions
WTChangeProposal	All	Change Permissions
WTChangeRequest2	All	Change Permissions
WTChangeOrder2	All	Change Permissions
WTChangeActivity2	All	Change Permissions
WTVariance	All	Change Permissions
VariantSpec	All	Change Permissions
SubFolder	All	Change Permissions

**Note:** No rules are defined in the System domain for the COLLABORATION MANAGER group.

### Rules in Default Domain for VARIANCE APPROVERS Group

Out-of-the-box Product Design access control rule for the VARIANCE APPROVERS group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
WTVariance	UnderReview	Read and Download

### Rules for SHARED TEAM MANAGER Group

Out-of-the-box Product Design access control rules for the SHARED TEAM MANAGER group (Shared Team Manager role) are described in the following sections.

#### Default Domain Rule for SHARED TEAM MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the Default domain out-of-the-box rules define for the Shared Team Manager:

Object Type	State	Permissions
WTObject	All	Full control (All)

#### System Domain Rule for SHARED TEAM MANAGER Group

The following table lists the combination of object type, life cycle state, and granted permissions that the System domain out-of-the-box rules define for the Shared Team Manager:

Object Type	State	Permissions
WTObject	All	Full control (All)

### Rules in Default Domain for OWNER

OWNER is a pseudo-user that is used by Windchill to represent the object owner. Out-of-the-box Product Design access control rule for the OWNER group is described in the following table. The table lists the combination of object type, life cycle state, and granted permissions that are defined in the Default domain out-of-the-box rules:

Object Type	State	Permissions
PromotionNotice	UnderReview	Read, Download, and Delete
PromotionNotice	Rejected	Read, Download, and Delete

## Updating Access Control Rules



**Caution:** PTC recommends that you use caution when modifying the access control rules that are set when contexts are created. Modifying rules can cause problems with access to data and administrative functionality.

To adjust the access control rules that are defined, you can use the Policy Administrator. Navigate to the product or library and click the **Policy Administrator** link from the **Utilities** page. Select the /Default domain and click **Update**. From the **Administrative Domain** window, click the **Access Control** tab to view the access control rules for the domain. From this tab, you can create new rules and update or delete existing rules. When creating or updating rules, the **Groups** tab shows the groups available from in the current context. These groups include the groups that map to the roles being used in the product or library team.

For additional information about creating or updating access control rules, see the [Access Control](#) chapter.


## Creating a Product

A product context provides the space in which a team of people can create and manage all of the information that is relevant to the design, manufacture, and support of a customer product. This information includes the following:

- A defined data storage area for the business objects associated with a customer product.
- A set of rules that control the access to the product and optionally set the numbering scheme, versioning scheme, life cycles, and workflows that are used with the objects that are associated with a product.
- The team of users who have access to the product.

Product contexts are created within an organization by members of the product creators group that is defined in the organization or by the organization administrator. The organization administrator can add users who are members of the organization to the product creators group by accessing the **Creators** link from the **Organization** tab.

The following procedure summarizes the steps needed to create a product (for detailed instructions, see the help available from the **New Product** window):

1. Navigate to the **Product** tab and click the new product icon . (The icon appears only if you can create new products.)

The **New Product** window opens.

2. On the **Define Object** step, select the type of product you are creating, and if you are using a shared team as the context team for the product, enter the name of the shared team.

3. On the **Set Attributes** step, select the context template and fill in the other fields in the **New Product** window.

**Note:** The slash (/) character and other special characters are not allowed in a product name. See the online help available from the **New Product** window for the complete list of restricted special characters.

For a description of the contents of the templates, see [Out-of-the-box Product and Library Context Templates](#).

4. Determine whether you want the product to be accessible through the normal domain-based access control rules or accessible only to the product team members.

When a shared team is used, the domain structure automatically changes to use the shared team domain that was created when the shared team was created.

When there is only a local team, this selection is made through the **Private Access** check box. If the check box is selected, then the product domain structure inherits from the Private organization domain rather than the normal PDM domain. This means that access is restricted to use only the access control policies that are defined within the product context being created; access control policies are not inherited from the parent PDM context.

For additional information on the domain structure, see [Managing Access to Data through Access Control Rules](#).

5. Click **Finish** to create the product and close the window.

**Note:** If you use a shared team that is not extended locally when creating a product and you are not a member of the shared team, you may not have access to the product that is created.


## Creating a Library

A library context provides the space in which you can store and provide access to business information. For example, all documents owned by a department can be stored in a department library.

Libraries can also hold objects that are not related to a single product. For example, parts that are related to more than one product could be stored in a common parts library (such as a Commodity Parts or Engineered Parts library), from which you allow multiple product teams access to those parts.

Libraries are created under an organization by members of the library creators group or by the organization administrator. The organization administrator can add users who are members of the organization to the library creators group by accessing the **Creators** link that is on the **Organization** tab.

The following procedure summarizes the steps needed to create a library (for detailed instructions, see the help available from the **New Library** window):

1. Navigate to the **Library** tab and click the new library icon .

**Note:** The icon appears only if you are a member of the library creators group for your organization.

2. On the **Define Object** step, select the type of library you are creating, and if you are using a shared team as the context team for the product, enter the name of the shared team.
3. On the **Set Attributes** step, select the context template and fill in the other fields in the **New Library** window.

**Note:** The slash (/) character and other special characters are not allowed in a product name. See the online help available from the **New Product** window for the complete list of restricted special characters.

For a description of the contents of the templates, see [Out-of-the-box Product and Library Context Templates](#).

4. Determine whether you want the library to be accessible through the normal domain-based access control rules or accessible only to the library team members.

When a shared team is used, the domain structure automatically changes to use the shared team domain that was created when the shared team was created.

When there is only a local team, this selection is made through the **Private Access** check box. If the check box is selected, then the library domain structure inherits from the Private organization domain rather than the normal PDM domain. This means that access is restricted to use only the access control policies that are defined within the library context being created; access control policies are not inherited from the parent PDM context.

For additional information on the domain structure, see [Managing Access to Data through Access Control Rules](#).

5. Click **Finish** to create the library and close the window.

**Note:** If you use a shared team that is not extended locally when creating a library and you are not a member of the shared team, you may not have access to the library that is created.

# Administering Teams

One of the main activities that a product or library manager has is to administer the local team associated with the product or library.

**Note:** If the context team created in a product or library includes a shared team or is exclusively made up of a shared team, the shared team is managed from the organization context and is not managed by the product or library manager.

When a product or library that uses a local team is first created, a base set of roles for the local team is established from the context template that is used in the creation. For information about these roles and the access control rules that are set for the roles that are in the out-of-the-box templates, see [Out-of-the-box Context Access Control Policies](#).

To administer a product or library local team, perform the following activities:

- Establish the roles that you want used in the team. You can add, remove, or create new roles.

The base set of roles that are established through the out-of-the-box templates is the minimal set of roles that you should have for a team. (See [Out-of-the-box Container Participation](#).) Before removing any of these roles, consider the consequences of the removal. For example in a product team, you should not remove the Product Manager role because this role defines who can administer the product.

The roles you can add are roles inherited from your organization. For additional information about these roles, see [Installed Site Context Participation](#).

You can also create roles specifically for your product or library local team. In addition to creating a new role through the **Team** page, you must define the access control rules for the role by using the Policy Administrator. As part of this activity, create the rules against the system group that is created for the team role. This group has the same name as the role that you create.

**Note:** Any new roles that you create are not available for life cycle, workflow, or team templates.

- Add users to the team by adding the users in the specific roles.

You can add users by selecting individual users or by selecting the user-defined groups or organizations that are available. For example, if the organization administrator has created groups for the organization context, you can select one or more of these groups to be a member of a role.

**Note:** When the organization context is created, the site administrator has the option of restricting user access. By default, users are restricted so that they see only users that belong to the organization. Selecting the **Allow entire user and group directory selection** check box when the organization container is created provides the ability to search for all users. If this check box was not selected when the organization was created, the site or organization administrator can update the organization context to select it.

- Change task assignments for members of a team.

As the membership of a team changes or members change responsibilities, you can manage the ownership of the tasks assigned to team members.

From the **Product** or **Library** tab, click the **Team** link to access the **Members** table. From this table, you can administer a product or library local team. For additional information about teams, see the [Teams](#) chapter. Be sure to read the [Team Best Practices](#) section in the chapter.

Click the **Assignments** link to access a table of assignments with the following views: **All**, **Open**, **Overdue**, **Assigned to me**, **Completed**, and **Uncompleted**. From this table you can manage the tasks assigned to your team.

## Using the Product and Library Utilities Page

The utilities listed on the **Utilities** page that is accessible from the **Product** and **Library** tabs allow you to perform administrative actions at a product and library level. The same set of utilities appears on both the **Product** and **Library** tabs. The difference is the context from which each utility is launched:

- Clicking a link from a product **Utilities** page launches the utility within the context of the current product.
- Clicking a link from a library **Utilities** page launches the utility within the context of the current library.

The utilities are grouped according to whether they are system administration utilities or business administration utilities. Many of the links provided on the page give you access to the utilities that you need to use to perform the duties described in [Typical Duties of Product and Library Administrators](#).

To explore the use of each utility, click the corresponding link on the page and then click the help icon in the window that opens.

## Configuring Products or Libraries for Dynamic Documents

Products and libraries that will be used primarily for managing dynamic documents are generally no different from products and libraries used for managing parts. When setting up a product or library for this purpose, review all of the activities described in the [Typical Duties of Product and Library Administrators](#) section, keeping in mind that the main business object type being managed is the Dynamic Document object type.



Since the out-of-the-box context templates do not have anything specifically set up for this object type, you may want to look closely at the following administration activities as they relate to managing this type of object:

- Create folders for use in organizing the dynamic documents that are checked into your Windchill solution.
- Create additional soft types and attributes for dynamic documents. The soft types and soft attributes correspond with Arbortext document types and attributes that are set up for use with the Arbortext Editor. See [Type and Attribute Manager](#).
- Set up specific teams, life cycle templates, and workflow templates for use with dynamic documents. For details on these topics, see the associated chapters: [Teams](#), [Life Cycles](#), and [Workflow](#).
- Review and set object initialization rules for Dynamic Document soft types. For information on setting object initialization rules, see [Object Initialization Rules](#).

**Note:** Autonumbering must be turned on for the Dynamic Document soft type and its subtypes. By default, autonumbering is turned on for the wt.epm.EPMDocument type and all of its subtypes through an object initialization rule in the site context. Since dynamic documents are subtypes of the wt.epm.EPMDocument type, the rule automatically applies dynamic documents unless additional rules have been set up that override the site container rule. For information about numbering objects, see [Dynamic Document Numbering](#).

- Set access control policies on the Dynamic Document soft types. For information on setting access control policies, see [Administering Domains and Policies](#).

Additionally, there are activities unique to using the Arbortext Editor and the Arbortext Publishing Engine with product and libraries that need to be performed. These include the following:

- Establish the configuration settings used by the Windchill connection to Arbortext Editor. For details, see the Windchill connection online help accessible from the Arbortext Editor.
- Manage the bursting rules used when a dynamic document is stored in Windchill. For information on bursting rules, see the Arbortext Editor online help.
- Manage additional soft attributes for WTDocument object type that are used by the publishing agent when publishing dynamic documents. For information on specifying publishing rules, including specifying rules for the resulting documents, see the *Windchill System Administrator's Guide*.

For general information about using product and libraries with Arbortext products to manage dynamic documents, see the *Getting Started Authoring Arbortext Documents* guide.

## Configuring Part and Document Relationships

The relationships between a part and a document, two parts, or two documents vary depending upon how the two objects are linked. This section briefly describes the different types of links that form these relationships and then describes the configuration options that are available.

For detailed information about understanding relationships between parts and documents, see the user's guide for your Windchill solution.

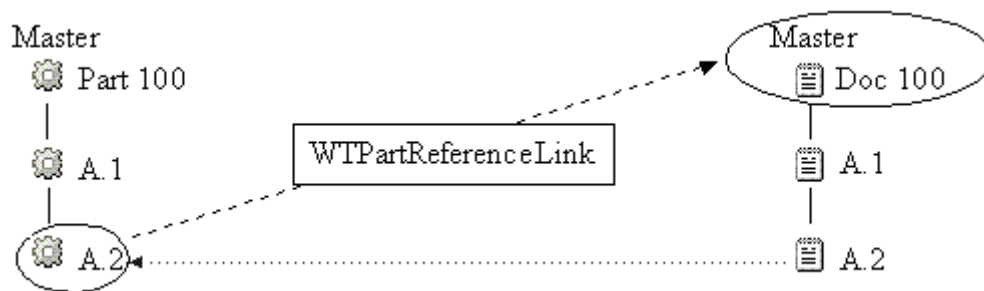
### Part to Document Relationships

**Note:** Part to document relationships do not apply to Arbortext Content Manager or ProINTRALINK 9.0.

The association between a part and a document is created from the part information page, document information page, Product Structure Explorer, or Product Structure Browser. Associations can be viewed from the part information page under the **Related Objects > Documents** menu, on the Described By Documents table or the References Documents table, depending on the relationship type. These same relationships can be viewed from the document information page on the **Describes Parts** and **Referenced by Parts** tables.

There are two types of links that can be established:

- A WTPartReferenceLink -- Using this type of link always links the part to the master document. Regardless of the version of the document the user selects, the part will always link to the master document when this link type is used. This is true even when the system has been configured to allow the user to select the link type and the document type.



Reference links always display a document version of the master based on life cycle state rules. For example, when a user clicks the reference link for Part 100 version A.1, Windchill searches for the latest released version of Doc 100 to display. If no version of the document has been released, it displays the latest working version (in this case, Doc 100 version A.2).

How to configure the life cycle state rules is described later in this section.

- A WTPartDescribeLink -- Using this type of link links the part to the latest iteration of a document and vice versa.



For example, when the user clicks a describe link for Part 100 version A.2, Windchill searches for the latest iteration of Doc 100 to display (in this case, Doc 100 version A.2).

You can configure the link types for part to document relationships in the following ways:

- From **Related Objects > Documents** on the part information page, the user creates the link using the actions from the **Described By** or **References** tables.
- From **Related Objects > Parts** on the document information page, the user creates the link using the actions from the **Describes** or **Referenced By** tables.
- From **Structure** on the part information page, the user creates the link using the **Add References Document** or **Add Described By Document** actions from the action list from the **Product Structure** table.

To allow the user to select the link type regardless of the document soft type, set the Part to Document Association Logic preference to Yes from the Preference Manager on the **Site** or **Organization** tab **Utilities** page. For details on using the Preference Manager, see [Administering Preferences](#).

Additional configuration options are described in the following sections.

For additional information on part to document relationships, see the *Windchill PDMLink User's Guide*.

## Revised or Saved Part to Related Document

When a user revises a part using the **Revise** action or saves the part using the **Save As** action, the new version of the part carries forward the link to the document by default. While the **Revise** action always carries the link forward, you can choose to prevent the link from being carried forward during a **Save As** action by removing the Relationship copy rules related to this operation from wt.properties.

For example, assume the following properties are set in wt.properties:

```
wt.enterprise.copyRuleDelimiter=,  
  
wt.enterprise.copyRulesN=wt.part.WTPart,Relationship,  
wt.part.WTPartReferenceLink-references
```

The first property sets the delimiter for the copy rules to the comma (,).

The wt.enterprise.copyRulesN property is the Relationship copy rule for wt.part.WTPart. This rule copies the references forward when the type of link is WTPartReferenceLink.

If you remove the wt.enterprise.copyRulesN property, then reference links are not carried forward.

**Note:** There can be no gaps in the sequence of copy rules. If you remove a copy rule, you must renumber those rules that follow. For example if there are six copy rules and you remove copyRules5, then you must renumber copyRules6 so that it is copyRules5.

Use the xconfmanager utility when modifying the wt.properties file. For information on using this utility, see [About the xconfmanager Utility](#). For additional information about the properties used for copy rules, see the description of wt.enterprise.copyRules in the properties.html file.



**Caution:** PTC recommends that you do not change the value of the wt.enterprise.copyServiceRules property. The property is used by internal services.

## Document Version Used with Reference Link

As described earlier, reference links (WTPartReferenceLink) link to a document master, but display a document version of the master based on the life cycle state rules for the document.

The default behavior is that Windchill searches for the latest released version of the document to display. If no version of the document has been released, it displays the latest working version of the document.

To change the default behavior, change the value set for the **List of comma separated live cycle states preference** in the Preference Manager.

**Note:** The states must be valid life cycle states. The states are defined as key value pairs in StateRb.rinfo and can be viewed in the life cycle template associated with the object. States are always specified using uppercase characters.

For example, to change the search to include the Released, Approved, and then Completed states of a document, set the preference value to:

RELEASED, APPROVED, COMPLETED

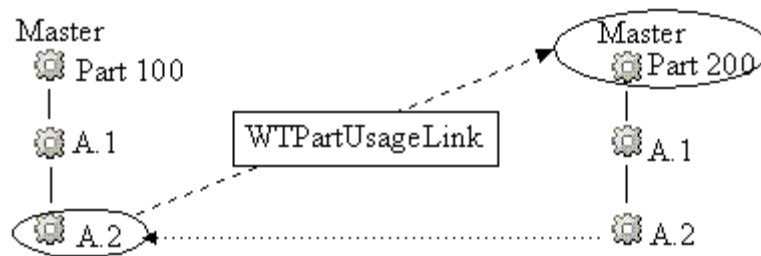
After this preference is set, Windchill searches for the latest Released version first. If none is found, it searches for the latest Approved version. If none is found, it searches for the latest Completed version. So, if a part is linked to Reference Document 4 which has three versions (A, B, and C), where A = Released, B = Approved, and C = In Work, based on the state settings in wt.properties, Windchill displays the latest iteration of version A which was released and ignores the rest.

## Part to Part Relationships

**Note:** Part to document relationships are available only in Windchill PDMLink, and do not apply to Arbortext Content Manager or ProINTRALINK 9.0.

A user associates one part with another by using the Product Structure Browser or the Product Structure Explorer. When a user makes the association, a WTPartUsageLink is created, forming a "uses part" relationship between a part and a part master.

A user associates one part with another by using the Product Structure Browser or the Product Structure Explorer. When a user makes the association, a WTPartUsageLink is created, forming a "uses part" relationship between a part and a part master.



The only configuration option for part to part relationships is described in the next section.

For additional information on part to part relationships, see the *Windchill PDMLink User's Guide*.

## Revised or Saved Parent Part to Child Part

When a user revises a parent part using the **Revise** action or saves the part using the **Save As** action, the new version of the part carries forward the usage link.

To prevent a link from being copied forward on a **Revise** or **Save As** action, you must remove copy rules from wt.properties. For example, assume the following properties are set in wt.properties:

```
wt.enterprise.copyRuleDelimiter=,  
  
wt.enterprise.copyRulesN=wt.part.WTPart,Relationship,  
wt.part.WTPartUsageLink-uses
```

The first property sets the delimiter for the copy rules to the comma (,).

The wt.enterprise.copyRulesN property is the Relationship copy rule for wt.part.WTPart. This rule copies the references forward when the type of link is WTPartUsageLink.

If you remove the wt.enterprise.copyRulesN property, then the usage links are not carried forward.

**Note:** There can be no gaps in the sequence of copy rules. If you remove a copy rule, you must renumber those rules that follow. For example, if there are six copy rules and you remove copyRules4, then you must renumber copyRules5 and copyRules6 so that copyRules5 becomes copyRules4 and copyRules6 becomes copyRules5.

Use the xconfmanager utility when modifying the wt.properties file. For information on using this utility, see [About the xconfmanager Utility](#). For additional information about the properties used for copy rules, see the description of wt.enterprise.copyRules in the properties.html file.



**Caution:** PTC recommends that you do not change the value of the wt.enterprise.copyServiceRules property. The property is used by internal services.

## Document to Document Relationships

There are several different types of document to document relationships in your Windchill solution:

- A document can reference another document under **Related Objects > Documents** on the document information page. This relationship creates a link of type WTDocumentDependencyLink.
- A document can serve as the parent to a second child document under **Structure > Document Structure** on the document information page. This relationship creates a link of type WTDocumentUsageLink.

There are no configuration options available for document to document relationships.

For additional information on document to document relationships, see the user's guide for your Windchill solution.

## Best Practices For Object Initialization Rules

Although you can set object initialization rules for individual product and library contexts, consider the consequences of doing this.

For most sites, the recommended approach is to set the rules at either the organization or site level so that an entire organization or site is using the same set of rules.

If you choose to set different rules in a product or library context, be aware that the ability to move objects between contexts may be restricted. Specifically, if you create a life cycle template in a specific product or library and the template is not available in other products or libraries, then you cannot move any object that uses the life cycle template. The same is true if the team template is unique to a product or library; objects using the team template cannot be moved. Similarly, the versioning scheme used for an object must match the scheme set in the destination product or library or the object cannot be moved.

**Note:** For demonstration purposes, the Product Design template (available by default when you create a product) includes object initialization rules that sets different default life cycle templates and versioning schemes for parts, documents, and CAD documents than is set out-of-the-box at the site level. This template demonstrates the use of simplified life cycles and state-based versioning. Unless you have overriding business practices that require unique rules, PTC recommends that if you want to use these features, you modify the organization or site object initialization rules and remove them from the Product Design template.





# 8

## Principals (Users, Groups, and Organizations)

This chapter describes the LDAP nodes, principals (user, group, and organization objects) that are used in your Windchill solution and details about managing the principals. It also provides information about profile management for dynamic control of visibility to actions.

Topic	Page
Overview .....	8-2
Working with LDAP Directory Services .....	8-5
Using the Principal Administrator.....	8-9
Understanding Principals (Users, Groups, and Organizations).....	8-24
Best Practices for Assigning Domains to Principals .....	8-25
Receiving Administrative Notifications .....	8-26
Managing the Principal Cache .....	8-27
.....	8-28
Profile Management .....	8-29

## Overview

Windchill uses the term *principal* to mean a user, group, or an organization; it includes any combination of users, groups, or organizations.

As the Windchill system administrator for any Windchill solution, you can create and update Windchill user, group, and organization objects through the Principal Administrator. As an Organization Administrator, you can update the Windchill user, group, and organization objects that are in the organization context that you are administering.

**Note:** When a Windchill solution is installed, the system administrative user (Administrator), the system administrative group (Administrators), and the initial organization object are always created. By default, the user Administrator (for example, wadmin) belongs to the Administrators group. This user does not have an organization affiliation (as defined by the LDAP organization attribute, which is "o" by default). Members of the Administrators group are granted all the permissions required to accomplish the administration tasks described in this guide.

**Note:** If your site does not use the organization attribute in the directory service entry, users can be assigned to an organization by specifying the usersOrganizationName property in a JNDI adapter. For more information on using this property, see adding enterprise directories in the *Windchill Installation and Configuration Guide - Advanced*.

Windchill uses both the Windchill database and a directory service when creating principals. For each principal, there is an entry in a directory service and a Windchill object stored in the database:

- The directory service entry contains attributes specific to the type of principal. For example, user entries have attributes for the user's full name, e-mail address, and organization.

The Aphelion Directory Service is set up when your Windchill solution is installed. Other directory services can be established by setting up JNDI adapter entries through the Info\*Engine Property Administrator and adding the adapter entries to the wt.federation.org.directoryServices property value. For additional information, see the *Windchill Installation and Configuration Guide - Advanced*.

- The Windchill object contains information that is relevant to Windchill (such as the associated domain) and the *Unique Federation Identifier (UFID)* associated with principal.

The UFID contains the distinguished name of the principal and identifies the directory service where the principal entry resides.

The following sections provide additional details about Windchill principals.

## Windchill Users

A Windchill user object identifies a user and is used when establishing group membership and policy rules for that user. It is stored in the Windchill database and holds user information for those users who have access to Windchill. This information includes the user name, the UFID associated with the user, the Windchill domain of the user, and administrative flags that are set if the object needs to be repaired or is disabled.

A Windchill user object is automatically created and persisted in the Windchill database the first time the user is selected from a search or the first time the user logs on to Windchill. In both of these cases, the corresponding directory service entry for the user already exists and is then referenced in the object that is created. As an administrator, you can also create, update, and delete users through the Principal Administrator.

Windchill does not rely on the user object to authenticate users. Rather, the Web server authenticates users and passes the authenticated user name to Windchill. The user's Web server ID is then mapped directly to the user object that has a matching user name.

Windchill users are usually affiliated with an organization that is set through the directory service organization attribute (by default, "o"). If the organization attribute is not set, then the user is an unaffiliated user and cannot create products, libraries, projects, or programs. However, that user can be invited to a team by e-mail and can do the same things within the product, library, project, or program as any other member.

**Note:** If your site does not use the organization attribute in the directory service entry, users can be assigned to an organization specifying the `usersOrganizationName` property in a JNDI adapter. For more information on using this property, see adding directory services in the *Windchill Installation and Configuration Guide - Advanced*.

## Windchill Groups

Windchill has two types of groups:

- User-defined groups are those groups created and managed by the users of a Windchill solution. These groups can be created through the Principal Administrator (or from the Organization Groups page), or can be created through a third party LDAP tool and have a corresponding UFID that is maintained in an LDAP directory service.
- System groups are groups created and managed internally by the system and do not have a corresponding UFID. Windchill uses system groups for managing context team membership and other system activities.

Organizing users into user-defined groups provides you with a more efficient way to apply policies for access control, indexing, and event notification, to populate participants in team and life cycle roles, and to populate recipients of workflow tasks. Each user-defined group object identifies selected users, organizations, and possibly other groups, under one name. You can create user-defined groups so that you can efficiently apply administrative tasks to groups of users, rather than to each user individually.

User-defined groups are associated with the context in which they are created. From the Principal Administrator, you only have access to the user-defined groups created through the Principal Administrator in the current context or in ancestor contexts. Some Windchill solutions also create and manage system groups that are used to manage team role membership. These groups are not accessible from the Principal Administrator.

A Windchill user-defined group object holds the group name, the UFID associated with the group, the Windchill domain of the group, and administrative flags that are set if the object needs to be repaired or is disabled. The UFID contains the distinguished name of the user-defined group and identifies the directory service where user-defined group entry resides.

## Windchill Organizations

Categorizing users by organization provides an additional way in which you can identify a set of principals by name. Windchill manages Windchill objects within each organization context and users affiliated with an organization can be given access to objects created from within an organization context.

An organization can be a company, company division, university, or some other list of people. Organization membership is defined by user directory entries that include the organization name defined for the organization object. For example, if the organization name is DIV1, then all users in the configured directory services who have their organization attribute (o, by default) set to DIV1 are members of the DIV1 organization.

**Note:** If your site does not use the organization attribute in the directory service entries for users, users can be assigned to an organization using the `usersOrganizationName` property. For more information on using this property, see adding directory services in the *Windchill Installation and Configuration Guide - Advanced*.

Organizations principals are associated with the context in which they are created. From the Principal Administrator, you only have access to the organization principals created in the current context or in ancestor contexts. Organization principals are used to define organization identifiers (such as CAGE codes or DUNS numbers), context ownership, and suppliers for supplier management activities.

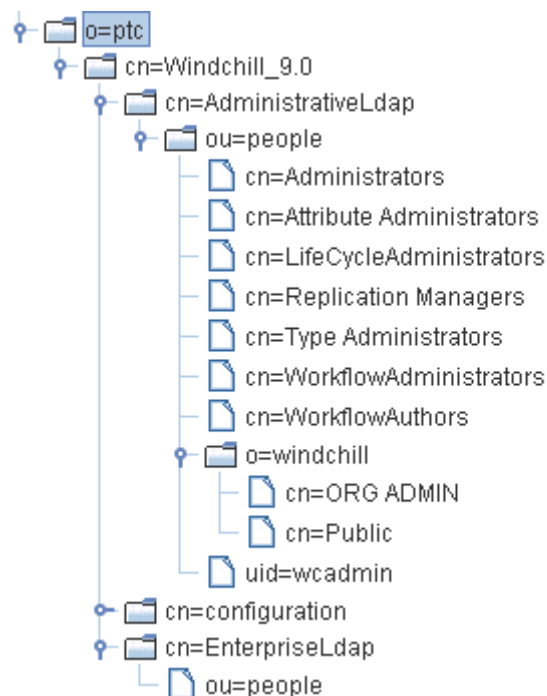
Each Windchill organization principal holds the organization name, the UFID associated with the organization, the Windchill domain of the organization, and administrative flags that are set if the principal needs to be repaired or is disabled. The UFID also contains the distinguished name of the organization and identifies the directory service where the organization entry resides.

## Working with LDAP Directory Services

When your Windchill solution was installed, two LDAP directory services were configured:

- Administrative LDAP -- This LDAP is intended to be used for your administrative users and any user-defined groups that you want to have administrative access. The display name for this service uses the reverse of the site domain and ends in .Ldap.
- Enterprise LDAP -- This LDAP is intended to contain your end users and any user-defined groups that you do not want to have administrative access. The display name for this service uses the reverse of the site domain and ends in .EnterpriseLdap.

For example, if your site has a domain of pct.ptcnet.com, and you accepted the default LDAP directory service options during installation, your directory service structure is similar to the following.

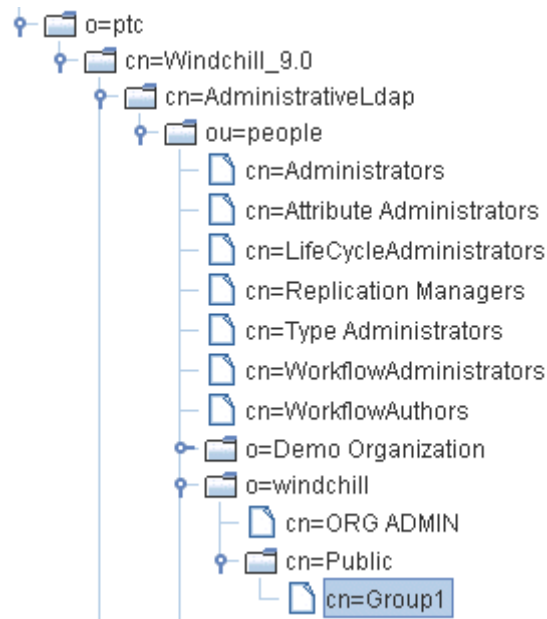


The Administrative LDAP includes the system-defined administrative groups.

When you create a new group using the Principal Administrator, you can select which LDAP it is created under by selecting the desired directory service as shown below:

<b>Directory Service:</b>	com.ptc.ptcnet.EnterpriseLdap
<b>Domain of Group:</b>	com.ptc.ptcnet.EnterpriseLdap
	com.ptc.ptcnet.Ldap

For example, if you use the Principal Administrator to create a new group from the site context, and you choose the Administrative LDAP as the directory service, the group is added as shown below:



If you use the Principal Administrator to create a new group from the site context and select the Enterprise LDAP, the group is added as shown below:

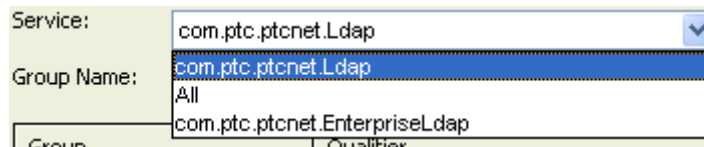


Any users created using the Principal Administrator are added under the Enterprise LDAP, as shown below:



For more information on the Principal Administrator, see [Using the Principal Administrator](#).

When creating an access control rule for a group using the Policy Administrator, select the directory in which the group resides to find the group, as shown below:



For more information on the Policy Administrator, see [Using the Policy Administrator](#).

By default, the Windchill search for groups uses a one level search scope in the selected directory service. This means that for Windchill to find the groups, you must store them at the top level established in the service. The Principal Administrator automatically does this.

If you add groups through a third party tool and the following things are true:

- You do not store the groups in the default search base associated with the service.
- You do not want to create a new directory service to map to this additional search base.

Then, you must add the search base to the following preference so the search base is included when Windchill searches for groups:

#### **Additional search bases which need to be queried**

Change the preference value using the site-level Preference Manager.

For more information on the Preference Manager, see [Administering Preferences](#).

For more information on adding directory services, see the *Windchill Installation and Configuration Guide - Advanced*.

**Note:** If your site requires changes in your established directory service structure, you should contact PTC Technical Support for assistance.

## Searching for Principals in Administrative Clients

The following administrative clients use a common interface when searching for principals (users, groups, and organizations):

- Policy Administrator -- When working with policy rules, you select principals against which the rules are applied.
- Life Cycle Administrator -- When defining an advanced life cycle template, you can select principals as participants for any of the roles defined for that life cycle template.
- Team Administrator -- When defining team in a team template, you can select principals as participants for roles.
- Workflow Administrator -- When defining activities, you can select principals to complete each assigned activity.

The method used to display users, groups, or organizations within an administrative client participants window is determined by the setting of the `wt.org.client.applet.ActiveSearch` property. By default, the property is set to false and the interface initially displays no users, groups, or organizations. When this is the case, the user enters values to restrict the search and clicks **Find** to display a subset of users, groups, or organizations. By setting the following property value in the `wt` properties file:

```
wt.org.client.applet.ActiveSearch=true
```

You can change the behavior of the interface so that when the participants window appears, it is automatically populated with all groups or organizations that are available.

**Note:** If you have a large number of groups or organizations, then automatically populating the participants window with them may take a long period of time. PTC recommends that you use the default setting for the above property unless you have a small number of groups and organizations.

In these administrative clients, the ability to locate users, groups, and organizations is determined by the administrator's access permissions and by the search scope used within each of the services searched. The administrator who is performing the search must have Read permission for principals to appear in the search result.

The service's default search scope is used to locate users and organizations (that is, the search scope configured in the service's JNDI adapter).



In the administrative clients, two types of groups can be displayed: system and user-defined groups:



- For the system groups, the groups established in the current context and in ancestor contexts are displayed.
- For user-defined groups, the search scope of the selected directory service determines which groups are displayed. See [Working with LDAP Directory Services](#).





## Using the Principal Administrator

You can access the **Principal Administrator** utility from the **Utilities** pages that are under the **Site** and **Organization** tabs:

- The Principal Administrator link from the **Site** tab provides you, the site administrator with access to users and to the user-defined groups and organizations created in the site context.
- The link from the **Organization** tab provides access to only those principals that belong to the organization context from which you launched the Principal Administrator and those that belong to the site context (which is its ancestor context). However, if the organization was set up so that it allows entire user and user-defined group directory selection, then you can see all users and groups (except for the system groups maintained by your solution). However, access control rules may be set to prohibit you from seeing some users and user-defined groups.


The Principal Administrator allows administrators to manage Windchill principal objects using the links on the **Principal Administrator** main page. The following table describes the links:

Link	Description
Home	Re-displays the designated home page of the Windchill solution from which you opened the Principal Administrator.
Users	Displays the <b>Users</b> table from which you can manage users. Click the Add users to this table icon  to add existing users to the table. Click the New User icon  to create a new user.  For additional information, see <a href="#">Managing Users</a> .

Link	Description
Groups	<p>Displays the <b>Groups</b> table from which you can manage user-defined groups that are created in either the current context or ancestor contexts. The Add Groups to this table icon  allows you to search for and populate the Groups table with a subset of the groups that exist in the system. Click the New Group icon  to create a new group.</p> <p>For additional information, see <a href="#">Managing User-defined Groups</a>.</p>
Organizations	<p>Displays the <b>Organizations</b> table from which you can manage organizations. The Add Organizations to this table icon  allows you to search for and populate the Organizations table with a subset of the organizations that exist in the system. Click the New Organization icon  to create a new organization object.</p> <p>For additional information, see <a href="#">Managing Organizations</a>.</p>
Maintenance	<p>Displays the <b>Disconnected Principals</b> table. This table contains the principals for which there is no match between the distinguished name of the principal and the distinguished name in the UFID that is stored in the database. From this table, you can search for additional principals that have nonexistent distinguished names, update disconnected principals, delete disconnected principals, or purge all principals from the cache.</p> <p>For additional information, see <a href="#">Maintaining the Connections between Principal Objects and their Directory Service Entries</a>.</p>

## Best Practices for Windchill PDMLink and Windchill ProjectLink

The Principal Administrator creates only organization objects (also referred to as the organization principal) and not organization contexts.

**Note:** Creating an organization principal in the Principal Administrator does not create an organization context. Both the organization principal and its context can be created from the **Organization** tab by using the New Organization icon . For more information about organization contexts, see the [Getting Started](#) chapter.

Ensure that users have an e-mail address, as many features in Windchill require users to have an e-mail address. If users do not have the e-mail attribute set in their user directory service entry, they cannot participate in the features that require an e-mail address.

### Searching for Users and Groups

By default, users are restricted to seeing only users and groups that belong to the organization in which a user is a member. When creating or editing an organization context from the **Organization** tab, you can select the **Allow entire user and group directory selection** check box to provide the ability to search for all users and groups. For user-defined groups, this setting allows you to see the group entries contained in the administrative LDAP directory hierarchy level of the Aphelion directory and each enterprise directory setup through a JNDI adapter (search scope set to ONELEVEL). Unless you need to restrict access to users and groups, PTC recommends that you select this check box.


For information about the out-of-the-box directory services that are set up, see [Working with LDAP Directory Services](#).


## Best Practices When Maintaining a Directory Service Outside of Windchill

Because your Windchill solution may be connected to a corporate LDAP rather than a Windchill-owned LDAP, you may not be creating users through the Principal Administrator; instead, users are automatically created in the Windchill database when the users become active in the solution. As users are removed or changed in the user directory service through an external tool, you will need to manage the Windchill user objects by doing the following:

- Deleting Windchill user objects that no longer have valid user directory service entries (see [Maintaining the Connections between Principal Objects and their Directory Service Entries](#)).
- Cleaning up after deleted users (see [Deleting Users](#)).
- Managing the principal cache so that changes in a user directory service are available in Windchill (see [Managing the Principal Cache](#)).

## Managing Users

Clicking the **Users** link on the **Principal Administrator** main page displays the **Users** table from which you can manage users. Clicking the Add users to this table icon  allows you to locate existing users and add them to the **Users** table.

Clicking the New User icon  allows you to create a new user. There are some restrictions for the user name you enter for a new user:

- User names cannot include the slash (/) character. This is because the slash is used as the folder separator in Windchill.
- Because of known problems with the Apache and Sun ONE Web servers, user logon names cannot contain extended ASCII characters nor multi-byte characters.

**Note:** Although the Principal Administrator does not require that users have an e-mail address, many features in Windchill require that users have an e-mail address.

After a user is added to the table, you can manage the user. In previous releases, users were identified by a user ID. The user ID is now known as the user name.

Managing users includes performing the following activities:

- Creating users, either from scratch or by starting with a similar user  
To create a similar user, access the information page of a current user and select the **Similar User** link.
- Searching for users
- Editing and deleting users  
When deleting users, you can delete them from just the Windchill database or delete them from both the database and the user directory service.
- Associating users with profiles
- Editing the domain and personal cabinet of a user
- Viewing information about users
- Defining electronic signatures for users.

For additional information on using electronic signatures, see the [Electronic Signatures](#) section in the Workflow chapter.

- Managing personal cabinet names

From Windchill PDMLink and Windchill ProjectLink, you can administer personal cabinets from the **Personal Cabinets Administration** link on the Site **Utilities** page.

- Purging users from the principal cache
- Refreshing the teams' membership using the **Recompute Team Membership** option.

For specific instructions on how to perform these activities, click **Help** from within the Principal Administrator.

## Changing User Passwords

When editing a user from the Principal Administrator, you can change the user password along with changing other user information if you have the rights needed to make changes in the directory service.

When you change the password for a user, the old password may still be active for a period of time. This is the case if the Web server you are using caches user information, as is true with the Apache Web server. If a user password is changed in LDAP, there may be a short period of time where the new password will not work, or where the old password and the new password will both work due to the cache. This is only temporary and will resolve itself when the Web server cache expires and the Web server refreshes the LDAP information. Generally, the time to live for cache entries is short. For example, the default for Apache is 10 minutes.

Additionally, you can configure your Windchill solution so that users can change their own passwords. For the details on how to configure password management, see the *Windchill System Administrator's Guide*.

## Naming a User's Personal Cabinet

Since a Windchill user name does not need to be unique and all personal cabinet names must be unique, Windchill uses the `wt.folder.personalCabinetNamingAttribute` property in the `wt.properties` file to determine what the initial personal cabinet name should be for a given user. The `wt.folder.personalCabinetNamingAttribute` property contains the following default ordered list of attributes:

`name` -- The cabinet name used is the user's name.

`eMail` --The cabinet name used is the user's e-mail address.

`fullName` --The cabinet name used is the user's full name.

For the cabinet name, Windchill uses the value of the first attribute in the list that produces a unique name. In most cases, the name of the personal cabinet is the user's name. If there is already a personal cabinet with that name, the user's e-mail address is used for the personal cabinet name. If the e-mail address is already being used as a cabinet name, then the full name is used. If the full name is already being used, the object identifier for the user (OID) is used as the cabinet name. The OID is a unique string that identifies each object in Windchill. If the OID is already in use, Windchill appends an underscore and an integer, starting with 1, to the object identifier (<oid>\_1, <oid>\_2, etc.) until a unique cabinet name is discovered.

You can change the attributes used in creating the personal cabinet name or the order of these attributes by modifying the list of attributes in the wt.folder.personalCabinetNamingAttribute property using the xconfmanager utility. Valid values are those attributes used in user directory service entries. For example, to use the full name before using an e-mail address, you could specify the following xconfmanager command from a windchill shell:

```
xconfmanager -s wt.folder.personalCabinetNamingAttribute=name,fullName,eMail,oid  
-t <Windchill>/codebase/wt.properties -p
```

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

To use a user's telephone number instead of the e-mail address, you could specify the following property and value pair:

```
wt.folder.personalCabinetNamingAttribute=name,fullName,  
telephoneNumber,oid
```

If the attribute list for wt.folder.personalCabinetNamingAttribute has been modified and no personal cabinet name is discovered using the modified list, then Windchill derives the cabinet name from the user's name with the OID appended (as discussed earlier in this section).

For additional information about using the xconfmanager utility, see [About the xconfmanager Utility](#).

## Associating Users with Profiles

Profiles define which parts of the user interface (for example, actions, tabs, and attributes) should be visible to a principal (such as a user, group of users, or the users of an organization). By associating principals with profiles, the user interface reflects the settings in the profile for that principal.


When creating a new user, editing an existing user, or creating a similar user you can associate that user with a profile. On the **Profiles** tab, specify the profile to which the user should be added as a member. You will need to create a profile on the **Profiles** sub-navigation link, prior to associating a user to a profile.

**Note:** Users can be added to a profile on the **New User** window, the **Edit User** window, and the **Similar User** window in all Windchill solutions except Pro/INTRALINK.

For more information about profiles, see the [Profile Management](#) section.

## Editing the Domain of a User

A domain is an administrative area that defines a set of administrative policies, such as access control, indexing, and notification. Objects associated with a domain are subject to its policies.



Using the Edit User Domain icon  on the **Users** table, you can change the domain of the user and the domain of the user's personal cabinet. For more information about naming the personal cabinet for a user, see [Naming a User's Personal Cabinet](#).

## Deleting Users



**Caution:** Do not delete a user unless you understand how it affects the system, as described in this section.

These two actions result in deleting a user:

- Delete Only from Windchill 
- Delete from Windchill and Directory Service 

The first action has the effect of deleting the user from the Windchill database. The second action deletes the user from both the Windchill database and the user directory service. To use the second action, you must have the required permissions to be able to delete users from the directory service as well as the database.

**Note:** You cannot delete the Administrator or the Administrator group. You also cannot delete yourself.

The results of deleting a user from the Windchill database are:

- The user is removed from all groups.
- All access control rules that specifically identify the user are removed.
- The user is removed from all notification lists within notification policy rules and, if deleting the user from the list results in an empty list, then the rule is also deleted.

The following rules govern tasks associated with a workflow process when a user is deleted from the Windchill database:

- If a user is deleted after a workflow process has been initiated, but prior to assignment of a task, the user is removed from the list of participants and no task is assigned. The value of the wt.property called ignoreUnresolvedRoles decides if the task should be completed or assigned to the responsible role.
- If the user is deleted after a workflow process has been initiated, and a task has been assigned, that task must be manually reassigned. The task will be reassigned to a context creator. (For more information, see the [Workflow](#) chapter.)
- A deleted user continues to appear in iteration history, object properties pages, and so on, but the name is not displayed as an e-mail link.
- When a user is deleted, the user is automatically removed from the list of participants in any workflow process template. The user is also removed from any role mappings created as part of a life cycle or team definition. The change does not result in an iteration to the workflow or life cycle templates.
- If a user is identified as a participant in a workflow template definition and that user is deleted from the system after the workflow has been initiated, any task that would have been assigned to the user is assigned to the Responsible Role which is generally the user who initiated the process.
- If both the template creator and a user identified in a workflow process template are deleted after the workflow process is initiated, the workflow process stops until the tasks assigned to the deleted user are manually reassigned.
- A user can be created with an identical user name, but if the original user's personal cabinet was not deleted, the new personal cabinet will have a different name. For more information, see [Naming a User's Personal Cabinet](#).
- If a deleted user is specified as the user of a collection defined in the index properties, a stack trace prints in the Method Server log when an attempt is made to index an object.

The results of deleting a user from both the Windchill database and the directory service include all results described earlier for deleting a user from the Windchill database and additionally include the following:

- A user is not authenticated when attempting to log into Windchill.
- The user's name is not included in search results.

If a user is not removed from the user directory service, a new user object is created in Windchill database when the user tries to log on or when the user is selected from a search. This new user object is not the same object that was deleted, and all of the results of the earlier deletion are still true. For example, the user is no longer a member of the groups to which the user had been a member.



After deleting a user from the Windchill database, you must perform the following clean-up steps:

- Reassign any items in the user's list of assignments.
- Unlock any objects the user has checked out of the Windchill database.
- Remove the user's personal cabinet and any folders or objects within it. From Windchill PDMLink, use the **Personal Cabinets Administration** link from the **Site Utilities** page.

## Changing the Organization to which a User Belongs

By using the Principal Administrator (if you have write permission to the directory service) or by using a third party tool that allows your site to modify user entries in your directory service, you can change the organization attribute for users. Changing the organization attribute by using the Principal Administrator automatically changes the other organization-related information for users. However, changing this attribute by using a third party tool does not change the other organization-related information for users.

If you change the organization affiliation of a user, the domain and personal cabinet of the user are automatically updated to reflect the change (by default). When the site preference named **Synchronize Domains for User Organization Changes** is set to true and the organization of a user changes, the user domain and the domain of the user's personal cabinet are reassigned to the new organization root domain. Additionally, the organization groups that are associated with the context teams for which the user is a member, are updated. The user is removed from the group for the old organization and added to the group for the new organization.


If this preference is set to false, the site administrator will have to manually execute the OrgSync command line utility to perform the update after using a third party tool to change a user's organization. For details on using this utility, [Using the OrganizationSync Utility for User Organization Changes](#).

You can turn off this automatic update through the **Synchronize Domains for User Organization Changes** preference that is under the **Display** category. For more information about preferences, see [Administering Preferences](#).

## Refreshing Team Membership for Users and User-defined Groups



Users and user-defined groups can be members of teams. Team membership can be refreshed when creating or editing users and groups, or at a later time by using a schedule. See the [Refreshing User-Defined Groups](#) section to schedule the team refresh at a later point in time.

Refresh the team membership for users from the **Groups** tab by selecting the **Recompute Team Membership** check box. This option is available when creating a new user, editing an existing user or user-defined group, and when creating a similar user or user-defined group. If the group membership is modified from within Windchill, the team's memberships are automatically updated. The Recompute Team Membership action should only need to be invoked if the membership of the group changes in LDAP.

Manually refresh the team membership for groups by accessing the **Members** table from each application context in which the group is participating; that is, from the Project context, Program context, Product context, or the Library context and click the Synchronize Teams with Groups icon .

**Note:** Depending on the number of teams the user or group is part of and the size of the affected teams, manually refreshing the team membership can add considerable processing time during peak usage periods. See the [Refreshing User-Defined Groups](#) section in the Teams chapter.

## Managing User-defined Groups

Clicking the **Groups** link on the **Principal Administrator** main page displays the **Groups** table from which you can manage user-defined groups. Clicking the Add Groups to this table icon  allows you to locate existing user-defined groups and add them to the **Groups** table. Clicking the New Group icon  allows you to create a new user-defined group.

Managing user-defined groups from the **Principal Administrator** includes performing the following activities:

- Creating user-defined groups, either from scratch or by starting with a similar group

To create a similar group, access the information page of a current group and select the **Similar Group** link.

- Searching for user-defined groups
- Editing and deleting user-defined groups

When deleting user-defined groups, you can delete them from just the Windchill database or delete them from both the database and the directory service.

- Viewing information about user-defined groups

- Purging user-defined groups from the principal cache
- Refreshing the teams' membership using the **Recompute Team Membership** option

If the group membership is modified from within Windchill, the team's memberships are automatically updated. The Recompute Team Membership action should only need to be invoked if the membership of the group changes in LDAP.

From within an organization, you can also use the **Groups** link to add or update user-defined groups that are defined in the organization context.

**Note:** The groups that can be managed from the Principal Administrator do not include system groups. These system groups (also referred to as internal groups) are automatically created and used by Windchill. For example, the system groups created for the context team roles can only be managed from the **Teams** link; they are not visible through the Principal Administrator.

For specific instructions on how to perform these activities, access the help from within the Principal Administrator.

The following sections provide additional information about working with groups and deleting groups.

## Working with User-defined Groups that are Maintained in a Directory Service

Any groups created in a node of a directory service that can be searched by your Windchill solution are automatically added to the Windchill database when the node is searched. The name of each Windchill user-defined group object is taken from the **cn** attribute of the LDAP group entry distinguished name (unless the mapping done in the JNDI adapter specifies a different attribute). This allows you to create groups using the editing tool available through a directory service rather than using the interfaces available in Windchill.

Adding and removing members from a user-defined group can also be accomplished using the editing tool available through a directory service. After updating group membership in the directory service, you can refresh the Windchill group membership as described in [Team Best Practices](#).

When access to Windchill has been limited, you can rename a group using the editing tool available through a directory service while maintaining the association of the directory service group with an existing Windchill group; however, the name of the group in the Windchill database cannot be changed to match the name stored in the directory service unless the user has write permission to both the database and the directory service. PTC recommends that you do not rename groups outside of the Principal Administrator.

Use the following steps to re-establish the connection between an existing Windchill group to an LDAP entry for a user-defined group that has been renamed in your directory service:

1. Limit access to Windchill before renaming the user-defined group.



**Caution:** If you do not limit access and someone searches and finds the renamed group or updates the user information for any member of the group, then a new Windchill group object is automatically created for the group. After a new Windchill group is associated with the LDAP entry, you cannot reconnect the renamed group to another Windchill group.

2. Rename the user-defined group using the editing tool available through a directory service.
3. From the Principal Administrator, click **Maintenance** to display the **Disconnected Principals** table.
4. In the row corresponding to the user-defined group that no longer has a related directory service entry, select the **Update** action.



The **Edit Principal** window opens from which you can re-establish the connection to an existing distinguished name for the user-defined group. The name of the user-defined group is updated to match the name in directory service entry.

## Deleting User-defined Groups



**Caution:** Do not delete a user-defined group unless you understand how it affects the system, as described in this section.

There two actions result in deleting a user-defined group:

- Delete Only from Windchill 
- Delete from Windchill and Directory Service 

The first action has the effect of deleting the user-defined group from the Windchill database. The second action deletes the group from both the Windchill database and the directory service. To use the second action, you must have the required permissions to be able to delete user-defined groups from the directory service as well as the database.

The results of deleting a user-defined group from the Windchill database are as follows:

- Users who were members of the user-defined group no longer belong to the group.
- All access control rules that specifically refer to this user-defined group are removed. If any users had access permissions derived solely from membership in the deleted group, it may be necessary to create new rules to restore the lost permissions.
- The user-defined group is removed from all notification lists within notification policy rules; if deleting the group from the list results in an empty list, then the rule is also deleted.

The following rules govern tasks associated with a workflow process when a user-defined group is deleted from the Windchill database:

- If a user-defined group is deleted after a workflow process has been initiated, but prior to assignment of a task, the group is removed from the list of participants.

If removing the group leaves no participants for a role, then the role resolution is determined by the settings in the `wt.properties` file:

- If the `wt.workflow.engine.ignoreUnresolvedRole` property is set to true and if the `ignoreUnresolvedRole` event configuration is set for this activity; then there will be no task created and the `WfAssignment` object completes so the workflow does not hang.
- If the `wt.workflow.engine.ignoreUnresolvedRole` property is set to false, one task is created that goes to the Responsible Role defined in the activity template. The default for the Responsible Role is the process creator. When the workflow process is started through a life cycle, the process creator is the creator of the business object.

For more information, see the [Workflow](#) chapter.

- If the user-defined group is deleted after a workflow process has been initiated and tasks are assigned, deleting the group has no effect on the process because the group itself is no longer being referenced. Tasks are assigned to the individual users that were in the group.
- When a user-defined group is deleted, it is automatically removed from the list of participants in any workflow process template. The group is also removed from any role mappings created as part of a life cycle or team definition.
- If a user-defined group is identified as a participant in a workflow template definition, and that group is deleted from the system after the workflow has been initiated, any task that would have been assigned to the group is reassigned to the responsible role. Generally, the responsible role is the user who initiated the workflow process.


The results of deleting a user-defined group from both the Windchill database and the directory service include all results described earlier for deleting a group from the Windchill database and additionally include the following:


- The user-defined group is not included in search results.


If a user-defined group is not removed from the directory service, a new group object is created in Windchill database when the group is selected from a search. This new group object is not the same object that was deleted and all of the results of the earlier deletion are still true. For example, the users who had been members of the group are no longer members.

## Managing Organizations

Clicking the **Organizations** link on the **Principal Administrator** main page to display the **Organizations** table from which you can manage organization objects, that are also known as organization principals. Clicking the Add

Organizations to this table icon  allows you to search for existing organization principals and add them to the **Organizations** table. Clicking the New

Organization icon  allows you to create a new organization principal.

**Note:** Creating an organization principal in the Principal Administrator does not create an organization context, it only creates the principal. Both the organization principal and its context can be created from the **Organization** tab by using the New Organization icon . When creating an organization context, you can associate the context with either an existing organization principal (that is not already associated with a context) or you can create a new organization principal.

Organization principals created using the Principal Administrator are considered restricted organizations. This means that no access control rules are automatically added to allow users in one organization to see users and groups from other organizations. When you create an organization context, you can select a check box that allows users in the organization to see all users and groups. This adds the organization to the Unrestricted Organizations group, which has the access control rules set to allow users to see other users and groups.

Organization principals can be used to identify an organization as the owner of specific parts and documents. By default, Windchill solutions are not set up to allow organization ownership selection. The organization under which the parts and documents are created automatically own them. There are multiple steps involved in enabling organization ownership selection, one of which is creating the organization principals. For details on how to enable organization ownership selection, see the [Owning Organization Principals](#) section in the Organizations chapter.

Managing organizations includes performing the following activities:

- Creating organization principals, either from scratch or by starting with a similar organization

To create a similar organization, access the information page of a current organization and select the **Similar Organization** link.

- Searching for organization principals
- Editing and deleting organization principals

When deleting organization principals, you can delete them from just the Windchill database or delete them from both the database and the directory service.

- Viewing information about organization principals
- Purging organization principals from the principal cache

For specific instructions on how to perform these activities, access the help from within the Principal Administrator.

**Note:** When specifying the internet domain name of an organization, the name you enter can contain only alphanumeric characters and the hyphen (-) character. Do not enter any other types of characters in the name.

The following section provides additional information about deleting organization principals.



## Deleting Organizations



**Caution:** Do not delete an organization principal unless you understand how it affects the system, as described in this section.

**Note:** You cannot delete an organization principal that is associated with an organization context.

There are two actions that result in deleting an organization principal. They are:

- Delete Only from Windchill 
- Delete from Windchill and Directory Service 

The first action has the effect of deleting the organization principal from the Windchill database. The second action deletes the organization principal from both the Windchill database and the directory service. To use the second action, you must have the required permissions to be able to delete organization principals from the directory service as well as the database.

The results of deleting an organization principal from the Windchill database are as follows:

- The organization is removed from all notification lists.
- All access control rules that identify this organization principal are removed from the access control policy for a domain. If any users had access permissions derived solely from membership in the deleted organization, it may be necessary to create new rules to restore the lost permissions.

When an organization principal is deleted from both the Windchill database and the directory service, it results in the above points in addition to the organization no longer being included in search results.

If an organization LDAP entry is not removed from the directory service, a new organization principal is created in Windchill database when the organization is selected from a search. This new organization object is not the same object that was deleted and all of the results of the earlier deletion still hold.

## Understanding Principals (Users, Groups, and Organizations)

Principals are identified across Windchill systems through the use of the principal Unique Federation Identifiers (UFIDs). The syntax for principal UFIDs is as follows:

*<distinguished\_name>:<guid>@<domain>*

where:

*<distinguished\_name>* is the distinguished name of the principal (which is made up of directory attributes that include the user name and directory location).

*<guid>* is the globally unique identifier of the repository in which the principal was first created or discovered.

*<domain>* is the Internet-style domain name of the repository in which the principal currently resides.

Together, *<guid>@<domain>* identifies the directory service in which the principal resides.



## Best Practices for Assigning Domains to Principals

When principal objects are created, they are associated with default domains as follows:

- Users who are affiliated with an organization (the organization attribute on their directory service entries is set) are associated with the domain that was created for the organization when the organization object was created. This domain usually has the same name as the organization (or a shortened version of the organization name) and the domain is a child of the site context User domain. If the organization is associated with an organization context, then the domain is in the organization context; otherwise, the domain is in the site context.
- Users who are not affiliated with an organization (the organization attribute on their directory service entries is not set) are associated with the site context Unaffiliated domain. This domain is a child of the User domain. One exception to this rule is the Administrator user, which is associated with site context System domain.
- User personal cabinets are associated with the same domain as the user. The one exception to this rule is that the personal cabinet for the Administrator user is associated with the User domain in the site context.
- User-defined groups created in the site context are associated with the site context Unaffiliated domain. This domain is a child of the User domain.
- User-defined groups created in an organization context are associated with the domain that was created for the organization when the organization object was created. This domain usually has the same name as the organization (or a shortened version of the organization name) and the domain is a child of the site context User domain. The domain is in the organization context.
- Organizations are associated with the domain that was created for the organization when the organization object was created. This domain usually has the same name as the organization (or a shortened version of the organization name) and the domain is a child of the site context User domain. If the organization is associated with an organization context, then the domain is in the organization context; otherwise, the domain is in the site context.

**Note:** The default domain associations described in the previous list only apply if you have not set up JNDI adapters that configure principal domain assignments as described in previous releases. The domain assignments in a JNDI adapter take precedence over the system defaults.

User objects and personal cabinets are created automatically when users are selected in a search or when users log in. These user objects and personal cabinets are always associated with the default domains described earlier. When you create a user through the Principal Administrator, you can select the domains. But in most cases, you should use the defaults (which are used when you do not select a domain).

When creating user-defined group objects through the Principal Administrator, using the default domain is usually a good choice. However, you may want to choose a different domain if you want policy rules from a domain other than the default to apply to the group object.

When creating organization objects through the Principal Administrator, using the default domain is usually a good choice. However, you may want to choose a different domain if you want policy rules from a domain other than the default to apply to the organization object. This may be the case if you want to allow users to select the owning organization when they create parts and documents. For the details on how to turn on the organization ownership feature, see the [Owning Organization Principals](#) section in the Organizations chapter.

## Receiving Administrative Notifications

When a user is deleted using the Principal Administrator, the administrator receives a notification by e-mail (or possibly through Windchill workflow). The administrator is notified that the principal has been disabled and that any additional manual actions should be taken, such as removing a user's personal cabinet.

Similarly, Windchill will notify the administrator that a repair is needed when it detects that a user, user-defined group, or organization needs to be repaired because the object in the Windchill database no longer references an existing directory service entry, possibly because the entry has been removed from or relocated in the directory service, the administrator is notified that repair is needed.

These notifications are initiated by calling the Info\*Engine tasks named `NotifyPrincipalDisabled.xml` and `NotifyPrincipalRepair.xml` in the `<Windchill>/tasks/wt/federation` directory (where `<Windchill>` is the Windchill installation directory). You can customize these tasks to tailor the way in which notification is done.

Also, you can change the e-mail address used for the notifications by setting the `wt.org.principalAdministratorEmail` property in the `wt.properties` file. The default e-mail address used is the e-mail address of the Administrator user (if one is set) or the `postmaster@<server_hostname>` e-mail address, where `<server_hostname>` is the value of the `java.rmi.server.hostname` property in the `wt.properties` file.

## Managing the Principal Cache

To improve the access time required for users, user-defined groups, and organizations, Windchill maintains an in-memory principal cache of user, user-defined group, and organization information.

**Note:** If user, user-defined group, or organization attributes are changed using an administration tool other than the Principal Administrator (for example, using a directory administration tool), then the cache containing those attributes must be purged in order for Windchill to display the changed attributes.

You can manage the principal cache in two ways:

- By setting the maximum time that the information about a principal can remain available in the cache. Cache entries are automatically purged when the system accesses stale entries.
- By selecting actions that purge information from the cache from within the Principal Administrator.

### Automatically Purging Entries from the Principal Cache

You can add the following properties to the `wt.properties` file to automatically purge principal cache entries:

- The `wt.principal.cache.timeToLive` property defines the amount of time that any given principal cache entry is available from the cache. Specify the property value in seconds. If the property value is not set or is set to zero or less than zero, then cache entries are not automatically removed from the cache. Out of the box, this property is not set.
- The `wt.principal.cache.timeToLiveRandomizer` property adds a random amount of time to the time stamp of each cache entry so that a large number of entries do not expire at the same time. Specify the property value in seconds. If the property is not set, the value defaults to 600 seconds (10 minutes). If the property is set to zero or set to less than zero, then no random value is added to the time stamp of each cache entry. Out of the box, this property is not set.

When a valid value is specified, the random amount added to the time stamp varies between 1 second and the property value. For example, if the property value is 600 seconds (10 minutes), a value between 1 second and 600 seconds is added to the time stamp of a cache entry when the entry is added to the cache.

Use the xconfmanager utility to add the properties. For example to add the wt.principal.cache.timeToLive property, specify the following xconfmanager command from a windchill shell:

```
xconfmanager -s wt.principal.cache.timeToLive=600  
-t <Windchill>/codebase/wt.properties -p
```

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

For information about using the xconfmanager utility, see [About the xconfmanager Utility](#).

## Manually Purging Entries from the Principal Cache

From within the Principal Administrator, you can purge either the entire cache or individual entries from the cache. For specific instructions on how to perform these activities, access the help from within the Principal Administrator.

## Maintaining the Connections between Principal Objects and their Directory Service Entries

Sometimes the definitions of principals in the directory service are changed using a directory administration tool other than the Principal Administrator. For example, another tool is sometimes used to move users, groups, or organizations from one directory location to another. When this happens, the distinguished names for the principals that are moved change. During normal operation, Windchill keeps track of the objects it encounters that do not have valid distinguished names.

Clicking the **Maintenance** link on the **Principal Administrator** main page displays the **Disconnected Principals** table, which contains principals for which the distinguished name currently associated with the principal is not valid.

From the **Disconnected Principals** table, you can do the following activities:

- Search for additional principals that have nonexistent distinguished names
- Update disconnected principals
- Delete disconnected principals
- Purge all principals from the cache

For specific instructions on how to perform these activities, access the help from within the Principal Administrator.

# Profile Management

Profiles allow the site and organization administrator to dynamically control which actions are visible to a user, group of users, or users in an organization by associating that information with a profile. A profile represents a typical category of user within a company and is based on the roles and privileges associated with that particular user category.

By defining a profile that exposes only the necessary functionality and information needed by a user, it creates a focused and simplified user interface, reducing confusion by eliminating areas of the user interface that could be distracting. This capability allows customers to ensure that a supplier, customer, or group of users is presented with a streamlined and focused set of information and actions.

Each user or group can be associated with one or more profiles. The least restrictive settings for visibility are assumed. For example, if a user is associated with two profiles where one profile hides an action, but the other profile allows visibility of that action, then the user will see the action in the user interface. If one profile allows read only visibility to an attribute, but another profile gives full rights to that same attribute, then the user will have full rights to the attribute.

Changes that are made to profiles take effect after the user subsequently logs into the system.

## Creating Profiles

Profiles are created from the **Profiles** sub-navigation link on either the **Site** or the **Organization** tabs.

Users, groups, and organizations can be associated with a profile when creating or editing a profile from the **Profile** sub-navigation link. When creating a user from the Principal Administrator utility, the user can be associated with any existing profile.

- The **Profiles** link from the **Site** tab provides you, the site administrator with access to all profiles created in the site context.
- The link from the **Organization** tab provides access to only those profiles that are available through the organization context that is active when you access the Profiles table as well as any profiles created in the site context (the ancestor context to the organization context).

For a detailed procedure to create a profile, see the Profile Management online help.

Profiles are defined and managed from the site and organization contexts. Profiles that are created in an organization context are peers to the profiles created in the site context, unless they are given identical names – that is, the system will merge the settings from all profiles that are associated with a user, regardless of whether they are defined at the site or the organization contexts in order to determine what will be visible to the user. When a profile is created in an organization context with an identical name to a profile created in the site context, the organization profile overrides the site profile.

## **Profiles as a Visibility Control Mechanism**

Profiles are not an access control mechanism; they are a user interface control mechanism.

The profiles that are associated with a user, group, or an organization control which actions and areas of the user interface are visible by those users. A user could have permission to edit an object based on an object domain-based policy, but not have visibility to that edit action for the object because this action is not included in the user's profile.

The primary goal in defining profiles is simply to hide the user interface elements (e.g., actions, tabs, and attributes) that a user has no need for or interest in. If a user does not have rights to an object or an action through access control rules, a profile will not add visibility to actions or information that are restricted by underlying access control policies. That is, you cannot grant a user visibility through a profile to an action or an area of the user interface for which that user does not already have access control rights. For example, a user could have access control permissions to the Create Part action in the system, but through a profile, that action can be hidden from the user and not visible in the user interface. On the other hand, if the user does not have access control permissions to the Create Part action, that action cannot be made visible to the user via a profile. It is the combination of a user being granted permission to an object or action via access control policies, and the profile granting visibility to those actions and objects, that will enable a user to see and perform an action.

When a profile is associated or disassociated from a user or group, no change in access permissions is implied or enforced.

## **Default Profile Behavior for a New User**

When a user is added to the system and the user is not associated with a profile, the user will inherit the profile that is associated with the organization of which the user or group is a member. If the organization to which the user is added is not associated with a profile, and the user is not associated with a profile, then the user's visibility to actions and areas of the user interface will be determined by the default profile system configuration. The user will not be associated with this default system profile but the system will use this profile for the user in the absence of any other profile association. For more information, see the [Global Default Settings](#) section below.

## Global Default Settings

The out-of-the-box system default profile provides visibility to all functions and areas of the user interface. This system profile may be modified by the site or system administrator to provide visibility to a minimal number of actions and information elements.

A site or system administrator can globally configure the default visibility for actions as well as the standard out-of-the-box roles. This is done by modifying the out-of-the-box system default profile. For more information, see the *Windchill Customizer's Guide*.

## Overriding Profiles in an Application Context

Profiles that are created in both the site and organization contexts can be overridden in a specific application context by the application context manager, such as in a specific project. For example, assume that a user is associated with a profile that is defined in the site context and this profile does not allow visibility to the actions that would allow that user to create folders in any application context. A project manager can override the site-defined profile privileges and allow visibility to the action for creating folders for that user within a specific project. Application context managers override the site-defined profiles from the Team user interface by using the Configure Actions for Roles option in the team in which the user is a member.

For detailed procedures on configuring the actions for roles from the Team user interface, see the Team online help, which can be accessed from the Team page.

## Default Visibility for Application Context Managers

The site and organization administrator can choose to set defaults for and hide visibility over a specific product, library, project, program, or other application context action or user interface element. That is, the site and organization administrator can carefully control which elements that application context managers (project, program, product, and library managers) are able to override in a context instance.

Removing the ability for an application context manager to override a profile setting is done by globally hiding visibility to the Configure Actions for Roles action for a specific context. If this is done, then the visibility of actions and areas of the user interface in a context instance cannot be modified by that application context manager.

If an action or user interface element is allowed to be configured or overridden by the application context manager in a context instance, then the user interface element will be displayed as a row in the Configure Actions for Roles dialog box which is accessed from the Team page.

The site administrator can define a default value (to show or hide) the user interface element but if the user interface element is made configurable in a context, then the application context manager can override the default and choose to show or hide the user interface element per role in the context team.

## Out-of-the-Box Profiles

The following table lists the profiles provided out-of-the-box and their descriptions:

Profile	Description
CAD Author	Provides visibility to the Project, Product, Library, and Change tabs and to the CAD data management functionality.
Consumer	Provides visibility to the Project, Product, Library, and Change tabs, but hides all authoring functionality across projects, products, and libraries.
Customer	Provides visibility to only the Folders subtab on the Projects tab and hides discussions, routings and search functionality, including the CAD data management functionality.
Employee	Provides visibility to the Project, Product, Library, and Change tabs, but hides the CAD data management functionality.
Non-Employee	Provides visibility to only the Project tab and hides global search, project search, and home page search as well as all authoring functionality.
PDM	Provides visibility to the Product, Library, and Change tabs, but hides the Project and Program tabs as well as the CAD data management functionality.
Procurement	Provides visibility to the Project, Product, Library, Change, and Supplier tabs, but hides the CAD data management functionality.
Project	Provides visibility to the Project tab, but hides all other tabs as well as the CAD data management functionality.
Supplier	Provides visibility to only the Folder, Team, and Discussions subtab on the Projects tab.

**Note:** The CAD data management functionality referred to in the out-of-the-box profiles is controlled by the action, **View Related CAD Documents** in the set action visibility list.



## Profile Actions and User Interface Elements

When creating a profile, you can specify visibility at the global level or by making specific actions and user interface elements visible within a specific application context.

The **Set Action Visibility** step displays a list of all the actions and user interface elements that can be hidden or made visible in a profile. The actions and user interface elements are displayed based on the application context in which they reside in the user interface, for example, the Product, Project, Library, and Program tabs.

Within each column, visibility can be given to all or only a subset of actions or user interface elements within a specific application context by selecting the corresponding check box. For detailed procedures for setting visibility of the actions and attributes, see the Profile Management online help.

Visibility can also be specified at the global level using the **Global** column. For example, if you are creating a profile where visibility should be given for everything in the Change and Library contexts, you can select the check boxes in the Global column for **View Library Tab** and **View Change tab**.

The global settings are available for profiles only and cannot be modified in a context instance. Application context managers can only modify the action list that is applicable to their active context.

### Default Settings for Actions

Out-of-the-box, most of the actions and user interface elements are selected by default. That is, they are by default made visible in a profile unless the administrator deselects the corresponding check box. The following table lists the out-of-the-box actions that are by default deselected, unless an administrator specifically grants visibility by selecting the check boxes:

Profile Action Name	Default
Configure Actions for Roles  <b>Note:</b> This action sets the default visibility within a specific tab for the application context manager to override the profile for that specific instance in the context.	deselected (not visible)
Modify Teams  "Modify Team" is not a specific action found in the user interface. The term is used to describe the general task of changing and modifying a team which includes multiple actions in the user interface.	deselected (not visible)



# 9

## Teams

This chapter provides information about context teams, team templates, and roles.

<b>Topic</b>	<b>Page</b>
Overview .....	9-2
Context Teams.....	9-3
Team Templates .....	9-12
Team Best Practices .....	9-30

# Overview

Teams and team templates are used throughout Windchill.

To manage teams, you need to be aware of the different ways that teams are used in Windchill.

- When business objects (such as parts and documents) are created, a unique team is created as its own object and is associated with the business object. This *team* contains all the roles consolidated from the team, life cycle, and workflow templates. The roles get mapped to end users; ad hoc access permissions can be defined for the participants in the life cycle and workflow templates.
- At the context level, the manager can create a context team specific to a particular project, library, program, or product. Each *context team* identifies the members of the specific project, library, program, or product and assigns the members to roles that have been established in the context. Each context team can be made up of a local team, a shared team, or the combination of a local team and shared team. For additional information about context teams, see [Context Teams](#).

The context teams are also used when creating the team that is associated with a specific business object. For additional information, see [Team Roles Resolution](#).

The *team template* is an object managed by administrative users. This object can map participants and actor roles to roles. The team template can be assigned to a life cycle or workflow-managed business object, when it is created, to use as a template for roles resolution for the team.

**Note:** Windchill ProjectLink does not use team templates.

To understand this chapter, you should be familiar with the following terms:

- A *principal* is an individual user, group, or organization.
- A *role* is a function that can be fulfilled by some principal. A role is mapped to participants. A list of predefined roles is available when you define a team.
- An *actor* represents a user who performs a specific action within the context of a specific business object. Currently, Creator is the only actor defined.
- A *participant* is a principal or an actor, which has been mapped to a specific role in a team.

## Context Teams

A context team can consist of one of the following:

- A *local team*. This is the team that is established when the application context is created and is managed from the context.
- A *shared team*. This team is maintained in the organization context and then selected when application contexts are created. You cannot add a shared team to an existing application context.

There can only be one shared team in a context team and the shared team cannot be modified from the application context. Modifications to a shared team affect all context teams using the shared team and are done from the organization context.

- Both a local team and a shared team.

To use a shared team, the shared team must be available before creating the application context. If you select a shared team when creating a context, the shared team attributes determine if you can add local team members to the team:

- If the shared team you select can be extended to allow local team members, then when you create the application context, you can choose to allow roles and members to be added locally.
- If the shared team you select cannot be extended, then you will not be able to choose to allow roles and members to be added locally.

If the context team consists of only a local team or is a team that has both local and shared teams, you can modify the local team from the **Team** page that is available from the application context by adding and removing roles, users, and user-defined groups. You can define the local team using any of the roles that are predefined for your organization or for your context, or you can define new custom roles that are specific to a context. Then, you can invite members to one or more roles in the local team. Users in local teams can have different roles in each context in which they are a participant. In one context, a user can be the manager, while in another context, the user can be a member. Users in shared teams have the same role in all contexts.

If the team established for a specific application context is a shared team that is not extended locally, then the entire context team can only be modified from the organization context.

As part of saving an existing application context as either a new application context or as a context template, you can recreate an established context team (both local and shared teams, if applicable) in another application context. When using a context template that has a shared team named in the template, the shared team named must be available in the organization context before the template can be used to create a new context. For details on the saving an application context as a template, see [Methods of Creating Additional Context Templates](#).

## Local Teams

When creating programs and projects that have a local team, you can add roles and members during the creation step. Additionally, you can change roles and members using the **Team** page that is available from the program or project after the creation steps are complete.

When creating products and libraries that have a local team, the set of roles added to the team is fixed based on the context template that you select. After the product or library is created, you can change the roles and members using the **Team** page from the context.

## Creating Local Teams

When creating an application context, the team you create is a local team if you do not select a shared team.

If you select a shared team and that team is extendable, you can choose to add roles and members to a local team. Select the **Allow roles and/or members to be added locally in this context** option to be able to add local roles and members.

## Local Team Roles

Initially, a local team always has the following roles (but can have additional roles depending on the context template that is used to create the application context):

- Members -- Use this role to allow users general access to all actions in the context.
- Guest -- Use this role to limit users to the ability to view but not change anything in the context.
- A context manager role -- The context manager roles available are as follows:
  - Project context -- Project Manager
  - Program context -- Program Manager
  - Product context -- Product Manager
  - Library context -- Library Manager

Use the context manager role for users who have the responsibility of managing the context. This includes managing the team and all of the data in the context.

The context manager role is automatically assigned to the creator of the context; however, the creator can assign the role to other team members. No other users are automatically added to roles in the team.

Other than the context manager, Members, and Guest roles, there are no explicit means of associating responsibilities with roles; however, you can create a document in the application context that identifies the responsibilities of each of the team roles or members. If you are using team templates, you may want to set up team roles that match the roles used in the team template. For information about team template roles, see [Team Templates](#).

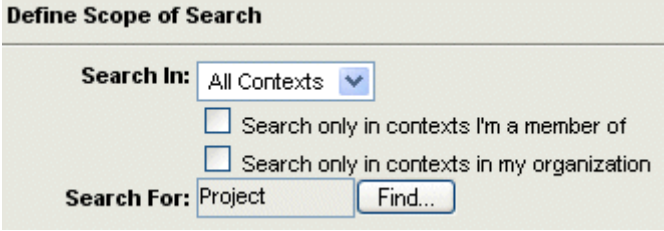
## Modifying Local Teams

Every local team must have one or more participants in the context manager role. You cannot remove this role from a local team.

To modify a local team, select either **Local Team Members** or the **Local Team Roles and Members** from the **Current View** drop-down list on the **Members** table.

Context managers, organization administrators, site administrators, and those granted the **Modify Team** action through a profile or through the team **Configure Actions for Roles** window can add and remove members and team roles from a local team.

**Tip:** To navigate to the **Team** page through the user interface, you must be a member of the context team. Others who are not members of the team but who have been granted the privilege to modify the team can locate the application context of the team through the **Advanced Search** option by searching for the specific context type. For example, you can search for all projects or programs using the **Project** object type as shown in the following:



The image shows a dialog box titled "Define Scope of Search". It contains a "Search In:" label followed by a dropdown menu currently set to "All Contexts". Below this are two unchecked checkboxes: "Search only in contexts I'm a member of" and "Search only in contexts in my organization". At the bottom, there is a "Search For:" label followed by a text input field containing the word "Project", and a "Find..." button to its right.

## Hiding the Guest Role

To hide the Guest role in a local team, a site or organization administrator can set the value of the **Hide Guests Group** preference in the **Display** category to **Yes** in the **Preference Manager** on the **Utilities** page of the **Site** or **Organization** tab.

After the preference is set to Yes, the Guest role is then hidden from users in all existing and new application contexts. The preference value can be changed at any time, however the change affects all application contexts within your site or organization, rather than individual application contexts. Setting this preference does not require a restart of the server; saving your changes automatically refreshes the page and the preference setting is active.



**Caution:** If a user is in the Guest role of an application context when the above preference is set to Yes, then that user's access control rules are not altered, and that user cannot be removed from the now hidden Guest role until and unless the property is set back to No and an administrator manually removes the user from that role.

For more information on preferences and the Preference Manager, see [Administering Preferences](#).

## Additional Local Team Information

For additional information about local teams, see the online help accessible from the **Team** page in an application context.

## Shared Teams

A shared team consists of a set of roles and the members added to those roles. Shared teams are created in an organization context from the **Teams** page and can be used as the context team for multiple application contexts that are created from the organization context. For example, assume you have created a shared team called the Product Demo Team in the Bike Company organization context. This shared team can be selected when users create application contexts (such as products and projects) when working in the Bike Company organization context. If users who are creating application contexts are members of a different organization context, then the Product Demo Team is not available.

**Note:** If there are no shared teams that are enabled when an application context is created, then the creator of the application context does not see the Shared Team field and the team created is a local team. A shared team cannot be added after an application context is created.

## Creating Shared Teams

By default, site and organization administrators can create shared teams. Additionally, from the **Creators** link on the **Organization** tab, you can add users to the Shared Team Creators group. Those users can then create shared teams.

When a shared team is created, an administrative domain is created with the same name as the shared team. The parent domain of this shared team domain is the Default domain that resides in the organization context where the shared team is created. Use this domain to set access control policy rules for the members of the shared team. For the domain structure and an introduction to managing data, see the [Managing User Access to Data](#) section in the Administration Overview chapter.

When creating a shared team, you can set whether the shared team can be extended locally when an application context using the shared team is created. To limit all context teams that select the shared team to only use the shared team, ensure that the **Allow team to be extended locally in associated application contexts** option is not set.



Each shared team has an owner. The owner is established when the team is created but is not automatically added as member of the shared team and has no special team privileges. If you want the owner to be a member of the shared team, be sure to add the user who is the owner to the team. If you have access to the **Shared Teams** table, you can e-mail all shared team owners by using e-mail team owners icon.

## Exporting and Importing Shared Teams

You can export a shared team from the **Shared Teams** table or from the information page of the shared team. The exported file is either a ZIP or JAR file containing a text file that names a business XML file and the business XML file that contains the XML elements that define the shared team.

You can import an exported team into another organization context or another Windchill system. Any roles imported that are not defined in the organization are considered custom roles.

The system does not verify that the members of a shared team that is being imported exist in Windchill nor does it check to determine if you can view the members of the shared team being created. Ensure that appropriate participants are listed in the XML files that you import. Including placeholder e-mail addresses for shared team members is not supported when exporting and importing shared teams.

The XML elements in a shared team business XML file are described in [Creating Business XML Files for Shared Teams](#).

## Shared Team Roles

The initial set of roles established in a shared team are:

- **Shared Team Manager** -- Use this role to identify the participants who are assigned to the context manager role when the shared team is included in a context.

The system group for this role is added as a member of the context manager group created when the shared team is used in a context team. Therefore, the participants in this role inherit the access control rules set for the context manager, but are maintained separately in the **Members** table for the context team.





- **Members** -- Use this role to allow users general access to all actions in the application context.
- **Guest** -- Use this role to limit users to the ability to view but not change anything in the application context.

Participants in the shared team **Members** and **Guest** roles also become members of the local team **Members** and **Guest** roles. The shared team **Members** and **Guest** groups are added as members to the local team **Members** and **Guest** groups.

You can add additional roles to the shared team. If you add other roles to a shared team that are also in the local team, the **Members** table shows the role as a shared team role and includes the members that are from both the shared team and the local team.

If you are using team templates, you may want to set up team roles that match the roles used in the team template. For information about team template roles, see [Team Templates](#).

Shared team roles and members are identified on the **Members** table using the shared team icon at the beginning of each row in the table. For example, the following rows in a **Members** table show the icon:

	Roles/Members
	Guest
	Members
	demo
	Shared Team Manager

When a role is in both the local and shared teams, the shared team icon appears at the beginning of the row for the role. To see any local members that are in the role, expand the row for the role when the view is **Members by Role** or change the view to **Local Team Roles and Members**.

**Note:** Although users in the Shared Team Manager role have the required permissions to modify the shared team, they can only do this if they are members of the organization where the shared team resides.

## Shared Team E-mail Invitation Messages

When creating a shared team, you can specify an e-mail invitation message. This message is only used when the shared team is used in a program or project context and new members are added to the shared team.

## Shared Team Access Control Rules

When you create a shared team, the following access control rules are set in the shared team domain by default:

Object Type	State	Principal	Permissions
WTOBJECT	All	Team Members	Read and Download
WfExecutionObject	All	Team Members	Read, Modify, and Create
WTOBJECT	All	Guest	Read and Download

The shared team domain is a child of the /Default domain in the organization context where you create the shared team and has the same name as the shared team. For additional information on shared team domains, see [Shared Teams](#) in the Administration Overview chapter.

## Additional Shared Team Information

For additional information about shared teams, see the online help accessible from the **Teams** page in an organization context.

## System Groups Associated with Teams

System groups are used to manage team role membership and are maintained only in the Windchill database (and not as entries in your LDAP directory service).

Each role that is included in a context team or in a team template has a corresponding system group with a name that is very similar to the role name. For example, the system role group for the Change Admin II role is CHANGE ADMINISTRATOR II and the system role group for the Members role is MEMBERS. System groups for roles are shown in the interface using either the role name or using all uppercase letters.

Additionally, the following system groups are maintained for each context team:

- The Team Members group has all role groups that are created for the team as members of the group.

In some places in the user interface, this group is shown as Team Members and, in other places, it is shown as teamMembers.

- Organization groups contain the participants in the team grouped by organization.

When a context team is created, a set of team organization groups is also created. There is one system group for each organization represented in the team. As team members are added and removed, the organization groups for the team are automatically updated.

In some places in the user interface, a team organization group is shown using the organization name and, in other places, it is shown as xxxx\_ORG, where xxxx is the internal number associated with the organization.

The system groups associated with teams are used in administrative activities such as defining access control policy and ad hoc rules, life cycle activities, and setting up action visibility for roles in a team.

## Restricting the Visibility of Actions in the User Interface through Teams

From the **Members** table of a context team, you can restrict the visibility of the actions in the user interface for the roles that are in the team.

**Note:** The ability to restrict the visibility of actions by role is not the same as restricting the visibility of actions for individual users, user-defined groups, or organizations. Restricting visibility for individual users, groups, or organizations is done through the use of profiles. For information on profiles, see [Profile Management](#).

You can indicate which set of actions should not be visible to those members of the team who are in selected roles. The visibility is not controlled by access control rules and is not contingent on certain permissions being set. Configuring actions for roles determines what appears in the user interface when users are in a specific team role and are accessing the context in which the visibility has been restricted.

**Note:** You cannot restrict the visibility of actions from a shared team.

Set the **Current View** to **Members By Role** to view the roles currently used. By selecting roles and then clicking the Configure actions for roles icon, you can display the list of actions for which visibility can be controlled. For example, to configure visibility of actions for the members of the Reviewer role, select **Reviewer** and then click the Configure actions for roles icon. From the **Configure Actions for Roles** window, clear the check boxes below the role name to hide the action. Those roles not displayed in the window keep the configuration that is currently set for the roles. If you haven't configured actions for a role, the default set of actions that is displayed when the role is displayed in the window is stored for the role.

**Tip:** The Team Members role always appears in the list of roles. Use the Team Members role to set the action configuration for all new roles added after you have configured actions for roles through the **Configure Actions for Roles** window.

**Note:** If a user is a member of multiple roles, the user has access to the union of the actions. For example, if a user can view the **Team** page because visibility is set in one role, the user keeps that ability even if other roles do not grant this visibility.

If you have configured some roles to restrict the visibility of some of the actions, you can include those visibility restrictions when you export the application context, save the context as a template, or save as a new context. To include the restrictions, be sure to select team roles when exporting, saving as a template, or saving as a new context. For details on exporting or saving a project as a template, see the [Context Templates](#) chapter. The XML tags used for exporting and then importing team roles are described in the [Common WTPPrincipalReference Element](#) section of the Creating XML Files for Templates and Shared Teams appendix.

## Actions that can be Configured

The actions that you can configure are limited to those actions directly related to the context in which the configuration is taking place and only affect the users in the roles configured in the context. For example, restricting the visibility of the **View Team Page** action for the members of the Guest role in a specific project does not affect the visibility of the action for members of the Guest role in another project or other context.

The actions displayed in the **Configure Actions for Roles** table are determined by the context you are in when you are setting the visibility. For example, the **View Plan Page** action is only applicable for programs and projects and, therefore, is only in the table when you are configuring a program or project context team.

The actions displayed in the **Configure Actions for Roles** table are a subset of the actions that can be set when creating profiles. Profiles can be used to restrict the visibility of actions for specific principals (users, user-defined groups, or organizations).

**Note:** Configuring actions for roles from the **Members** table disables the use of the profile settings for any members who have profiles and are members of any of the roles in the context. The profile settings affected are only those that relate directly to the application context.

For information on the actions available in profiles, see [Profile Management](#).

## Additional Action Visibility Information

For additional information about action visibility for roles, see the online help accessible from the **Configure Actions for Roles** table.

# Team Templates

This section describes the following:

- The out-of-the-box team templates
- Team role resolution
- Defining team properties and roles
- Assigning participants to team roles

Teams and team templates make it possible to define a smaller number of life cycles, since the life cycle roles can be mapped to team roles, rather than to specific users and groups. For more information about life cycles, see [Life Cycles](#).

The differences and relationships between teams and team templates are summarized in the following list:

- Team templates are used only in the creation of teams. When a business object that requires a team is created, the necessary team is automatically created.
- Updates to team templates do not affect existing teams that were created from the template. Team template changes are reflected in new teams only.
- Each team created for a specific business object is distinct from other teams created for objects of the same type.

For example, two problem reports in the same library may have associated teams that initially appear to be identical, consisting of the same roles and participants; however, changes made to one of the problem report teams do not affect the other problem report team.

- Teams cannot be manually created from team templates. Team creation is automatic, based on associations to business objects and workflow requirements.

**Note:** For more information about how teams are automatically created based on team templates and object associations, see [Team Templates and Object Types](#) later in this section.

## Out-of-the-Box Team Templates

This section summarizes the team templates that are provided with all Windchill solutions except Windchill ProjectLink. These team templates are designed to work with the Windchill Change Management functions.

In order for the Windchill Change Management functions to operate properly, each of the Change Management objects listed in this section must have an associated team template, and each of the associated team templates must contain certain roles. The following table indicates which team template roles administrators must define, and which are designed to remain empty. A "Yes" entry in the **Modify?** column means an administrator should define the role.

Team Template	Role Name	Modify?	Notes
Problem Report Team	Change Admin I	Yes	Assign this role to the user serving as Change Administrator I for problem reports. This user is the initial reviewer of problem reports.
	Problem Report Author	No	This role is used for workflow notifications, and is automatically set to the creator of the problem report. It should be left blank in the team template.
Change Activity Team	Assignee	No	This role is set through the user interface. It should be left blank in the team template.
	Reviewer	No	This role is set through the user interface. It should be left blank in the team template.

Team Template	Role Name	Modify?	Notes
Change Request Team	Change Admin I	Yes	Assign this role to the user serving as Change Administrator I for change requests. This user is the initial reviewer of change requests.
	Change Admin II	Yes	Assign this role to the user serving as Change Administrator II. This user is responsible for creating change notices.
	Change Review Board	Yes	Assign this role to the users who are responsible for approving full-track change requests.
	Change Request Author	No	This role is used for workflow notifications, and is automatically set to the creator of the change request. It should be left blank in the team template.
	Problem Report Author	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.
Change Notice Team	Change Admin III	Yes	Assign this role to the user serving as Change Administrator III. This user is responsible for auditing a change notice after it has been completed and before the product data is released.
	Change Admin II	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.
	Change Admin I	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.
	Change Request Author	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.
	Problem Report Author	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.
	Change Implementation Board	No	This role is used for workflow notifications, and is automatically set. It should be left blank in the team template.

To associate a team template with a particular object type, use the Object Initialization Rules Administrator.



## Team Templates and Object Types

This section describes how team templates are associated with Windchill object types. Object associations determine how teams are automatically created from team templates.

### Out-of-the-Box Associations

Out-of-the-box team templates are associated with the following object types:

Team Template	Object Types
Change Activity Team	wt.change2.ChangeActivity2
Change Notice Team	wt.change2.ChangeOrder2
Change Request Team	wt.change2.ChangeRequest2 wt.change2.WTChangeProposal
Problem Report Team	wt.change2.ChangeIssue
Default Team	All other objects

With the exception of the Default team, the out-of-the-box team templates are stored in the Windchill PDMLink cabinet. The Default team is stored in the System cabinet.

**Note:** The Default team template, which is stored in the System cabinet, does not contain any roles. If your site uses workflows to manage objects (such as documents and parts) other than change objects, you must add roles to the Default team template and any other team templates you create for use with non-change objects.

The change objects listed in the preceding table are the object names used in Windchill. The following table illustrates the correspondence between Windchill Change Objects as displayed in the Type and Attribute Manager and as displayed in the user interface. ("NA" indicates that the object is not applicable.)

Type Manager (Windchill) Change Object	Windchill Change Object or Field
WTAnalysisActivity	NA
WTChangeActivity2	Change Notice Implementation Task
WTChangeInvestigation	NA
WTChangeIssue	Problem Report (PR)
WTChangeOrder2	Change Notice

Type Manager (Windchill) Change Object	Windchill Change Object or Field
WTChangeProposal	Change Request Proposed Solution
WTChangeRequest2	Change Request

## Team Association

To associate a team template with a particular object type, use the Object Initialization Rules Administrator. For additional information, see the [Object Initialization Rules](#) chapter.

In the Site container, the value Default is set as the default team template for the following object types:

ManagedBaseline  
 WTProductInstance2  
 WTProductConfiguration

You can also create organization-, product-, and library-specific team associations by creating a team template in the corresponding context.

## Team Association Rules

When a workflow activity begins, an appropriate team is automatically created, according to the following rules:

- If you have established a default team template in the product or library context, that team template is used as the basis for any new team.
- If no corresponding team template exists in the product or library context, the default team template defined in the organization context is used as the basis for the new team.
- If no corresponding team template exists in the organization, product, or library context, the team template in the site context is used as the basis for the new team.
- The team is given a name containing the name of the object for which it is created.

**Note:** The System cabinet/domain is used to store document templates. For each product and library, a context domain is automatically created for storing document templates. For more information about domains, see [Administering Domains and Policies](#) in the Contexts chapter.

After a team has been created, users with the necessary permissions can edit the team members by clicking the team name from an object information page. On the **Team** page, click **Edit Team** to make changes.

For example, the out-of-the-box team template association for problem report objects is the Problem Report Team. If you were to create a problem report titled My Problem Report in the library called My Library, the team is created according to the following rules:

- If a team template called Problem Report Team exists in the My Library context, that team template is used as the basis for the new team.
- If no Problem Report Team exists in the My Library context, the Problem Report Team in the organization context is used as the basis for the new team.
- If no Problem Report Team exists in the organization, product, or library context, the Problem Report Team in the site context is used as the basis for the new team.
- The team is named My Problem Report.

Users with the necessary permissions can then modify the My Problem Report team by accessing the **My Problem Report** details page and clicking the team name. On the **Team** page, click **Edit Team** to make changes.

## Team Roles Resolution

The primary task in defining a team is selecting roles and mapping them to participants. To understand the concept and purpose of teams you should understand the relationship between teams and life cycles. (For additional information about life cycles, see [Life Cycles](#) in this guide.)

Business objects are associated with life cycles and teams, and roles are selected within these. The primary purpose of a team is to determine who is assigned the roles that are selected in a life cycle, that is, how life cycle roles are *resolved* to *principals* at runtime. Life cycles are more complex than teams, and they require more resources to create and maintain. Therefore, it is generally more efficient to create a relatively small number of life cycles with abstract roles and a larger number of teams that map roles to specific principals, which may change over time. Your site should have a policy regarding your use of teams with life cycles.

If you can modify a team, you should be familiar with existing life cycles that might be associated with the team you are about to create. You should then select a role for the team for every role that exists in the relevant life cycles.

## Initial Team Creation

The initial team creation is unique for Windchill ProjectLink.

### Windchill ProjectLink

If the object is assigned to a life cycle containing a phase workflow in the initial phase, the object uses the context team as the template. The behavior is dependent on the setting of the wt.property named wt.team.addEntireContainerTeam as follows:

- If the property is set to true, all roles and participants of the context team are copied into the team.
- If the property is false, only the roles and participants of the context roles matching the life cycle templates phase roles are copied into the team.

The default value of the wt.team.addEntireContainerTeam property is false.

If the object is assigned to a life cycle that does not contain a workflow process in the phase of the initial state (which is the case in Basic templates), it is assigned to a default team where the roles and participants are meaningless.

PTC recommends using life cycle templates of Basic type for better performance and scalability.

The team used for workflows and life cycles is created when the object is routed.

### All Windchill Solutions Except Windchill ProjectLink

If the object is assigned to a team template, the team template is resolved into the team. Roles and participants are added, and roles-actor roles are resolved to the users playing the role for the object and added to the team. For additional information, see [Administering Teams](#) in the Products and Libraries chapter.

## Role Resolution Rules

The property wt.team.re-resolveRoles is present in wt.properties. You can specify whether or not to have the life cycle roles resolve again when entering a new phase. If you wish to have the roles resolved, set wt.team.re-resolveRoles=true. The default is set to false and will not resolve roles at the new phase. If you do not wish to resolve roles, you do not have to add this property. More information about the role resolution and this property are as follows.

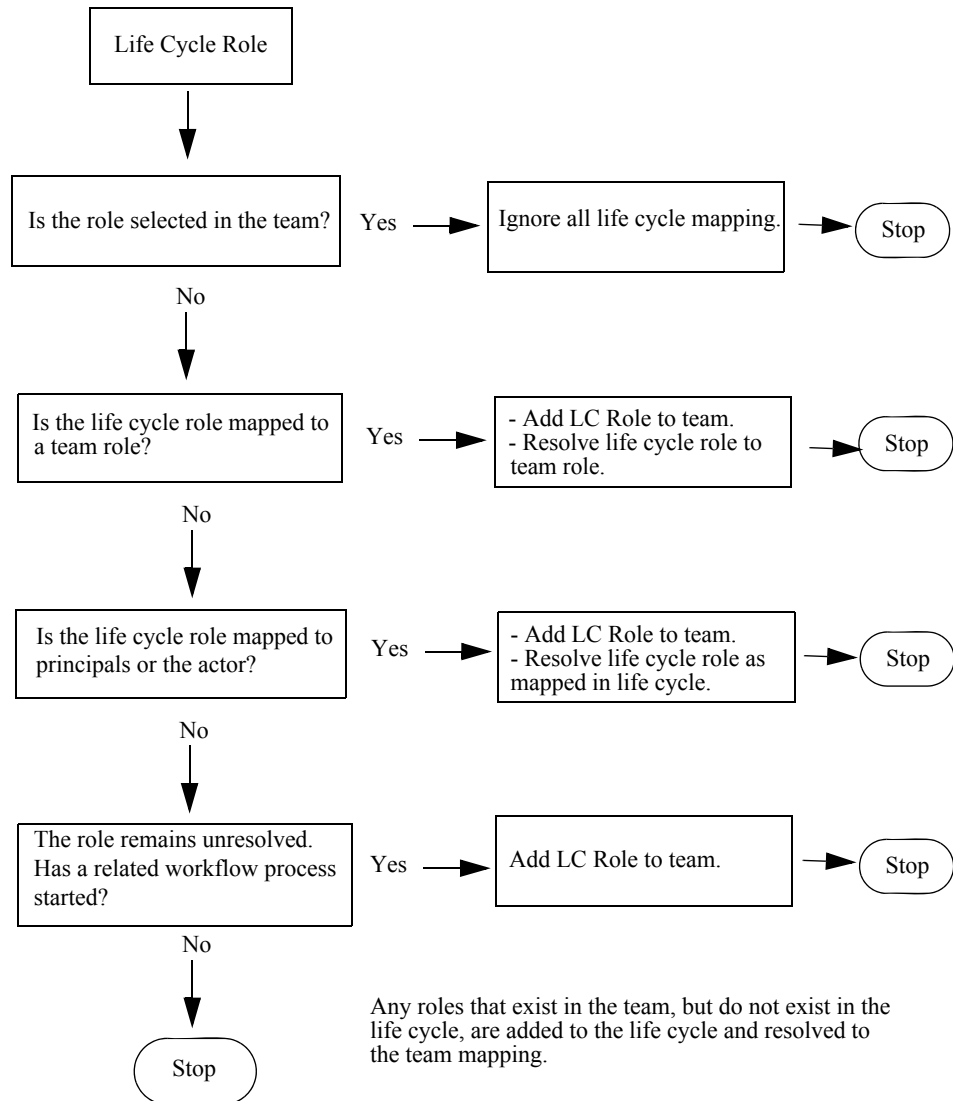
## Default Behavior

For the default, the following list illustrates the order in which Windchill tries to resolve each role in a life cycle:

1. If the life cycle role exists in the team template, the life cycle role is resolved to principals (or the actor for the role), *as defined in the team template*. All life cycle mapping for that role is overridden by the team values.
2. If the life cycle role *does not* exist in the team template (that is, rule 1 does not apply), but the life cycle role is mapped to an existing team template role, then the life cycle role is added to the team and resolved to principals, as defined in the team template role.
3. If the life cycle role does not exist in the team template and is not mapped to a role that exists in the team template (that is, rules 1 and 2 do not apply), then the life cycle role is added to the team and resolved to principals, as defined in the life cycle.
4. If the object's context team contains the role, any participants who play the role in the context team that are not already members of the role in the team are added to the team.
5. All roles that are not defined in the team, but are used in a related workflow process, are added to the team when the workflow process starts.
6. Any roles in the team template that is not used in the life cycle is also added to the team along with the participants in the role.

The following flow chart illustrates the Windchill business rules for resolving life cycle roles:

**Note:** Although it is possible to define a team that does not map roles to principals, or even to define a team with no role mapping, with typical usage, such a team would be useless.



## Property Set to True

If the property `wt.team.re-resolveRoles=true`, the only difference is the re-resolution of the roles when the object enters a new phase. Only the life cycle phase roles are re-resolved, i.e. the phase roles are cleared of their members and then re-resolved using the team template and the context team. All other run-time team roles which are not present in the life cycle phase are only updated using the context team.

If the property is set to false, the role resolution will only happen for the first phase of the first state and not for consequent states.

1. If the life cycle role is mapped to an existing role in the team template, the role is resolved to the members in the team template.
2. If the life cycle role *does not* exist in the team template (that is, rule 1 does not apply), but the life cycle role is mapped to an existing team template role, then the life cycle role is added to the team and resolved to the principals as defined in the team template role.
3. If the life cycle role is not mapped to an existing role in the team template, the life cycle role participants are resolved and added to the team in the role.

Example 1: The life cycle contains Role A and is assigned to Role B. The team contains Role B with member user x. The team template does not contain Role A. Role A is added to the team with user x as the participant.

Example 2: The life cycle contains Role A and is assigned to Role B. The team contains Role B with member user x. The team template contains Role A with user y. Role A is added to the team with user y as the participant.

4. If the object's context team contains the role, any participants who play the role in the context team that are not already members of the role in the team are added to the team.
5. All roles that are not defined in the team, but are used in a related workflow process, are added to the team when the workflow process starts.

## Role Resolution Example

This section contains an example of how roles might be resolved for a document object that is associated with a team and a life cycle.

The life cycle template contains the following roles:

	<b>1st Phase</b>	<b>2nd Phase</b>	<b>3rd Phase</b>	<b>4th Phase</b>
Submitter	creator	Not in this phase	Not in this phase	Not in this phase
Promoter	Not in this phase	Not in this phase	Not in this phase	Not in this phase
Reviewer	Design Engineer	Project Manager Prod Marketing	QA Engineer Pubs	Jane Design Engineer Project Manager QA Engineer
Observer	Not in this phase	Team Leader	Not in this phase	Not in this phase

The team template contains the following roles/participants:

- Design Engineer--Kristin
- Project Manager--Dave, John
- QA Engineer--Sean, Sachin
- Team Leader--Tom, Beth

The context team contains the following roles/participants:

- Design Engineer--Kristin, Flavio, Bill, Galen
- Project Manager--Dave, John
- Prod Marketing--Chris
- QA Engineer--Sean, Sachin, Iyrena
- Pubs--April, Diane, Muriel
- Team Leader--Tom, Beth



## First Phase

An object is created by Jeff and assigned to the life cycle template and team template above. The context team is listed above. The team resolves to the following participants for the first phase when the `wt.team.re-resolveRoles` property is set as shown in the table below:

	<b>Default (property set to false)</b>	<b>Property set to true</b>
Design Engineer	Kristin, <i>Flavio</i> , <i>Bill</i> , <i>Galen</i>	Kristin, <i>Flavio</i> , <i>Bill</i> , <i>Galen</i>
Project Manager	Dave, John	Dave, John
QA Engineer	Sean, Sachin, <i>Iyrena</i>	Sean, Sachin, <i>Iyrena</i>
Team Leader	Tom, Beth	Tom, Beth
Reviewer	Kristin	Kristin
Submitter	Jeff	Jeff
Promoter		

Observations:

- Participants are added during the team template/life cycle role resolution.
- Participants (Flavio, Bill, and Galen) were added to the design engineer role from the context team.
- Roles from the context team that do not exist in the team are not added (in this case, Prod Marketing and Pubs).
- The life cycle Reviewer role is mapped to Design Engineer, but since the context team roles are not added until after the team template/life cycle role resolution is completed, the people in the Design Engineer role are not members of the Reviewer role.

## Second Phase

The object is promoted to the second phase.

	<b>Default (property set to false)</b>	<b>Property set to true</b>
Design Engineer	Kristin, <i>Flavio</i> , <i>Bill</i> , <i>Galen</i>	Kristin, <i>Flavio</i> , <i>Bill</i> , <i>Galen</i>
Project Manager	Dave, John	Dave, John
QA Engineer	Sean, Sachin, Iyrena	Sean, Sachin, Iyrena
Team Leader	Tom, Beth	Tom, Beth
Reviewer	Kristin	Dave, John
Submitter	Jeff	
Promoter		
Observer	Tom, Beth	Tom, Beth

Observations are the same as for the first phase.

### Third Phase

The team templates and context teams are modified.

The team template contains the following roles/participants:

- Design Engineer--Kristin, Flavio
- Project Manager--Dave, John
- QA Engineer--Sean, Sachin, Dan
- Integration--Mark

The context team contains the following roles/participants:

- Design Engineer--Kristin, Flavio, Jeff, Michelle
- Project Manager--Dave, John
- Product Marketing--Chris
- QA Engineer--Sean, Sachin, Iyrena
- Pubs--April, Diane, Muriel
- Team Leader--Tom, Beth
- Observer--Jane, Lynn

	<b>Default (property set to false)</b>	<b>Property set to true</b>
Design Engineer	Kristin, Flavio, Bill, Galen	Kristin, Flavio, Bill, Galen, <i>Jeff, Michelle</i>
Project Manager	Dave, John	Dave, John
QA Engineer	Sean, Sachin, Iyrena	Sean, Sachin, Iyrena
Reviewer	Kristin	Sean, Sachin, Dan
Submitter	Jeff	
Promoter		
Observer	Tom, Beth	Tom, Beth, Lynn, Jane
Team Leader	Tom, Beth	Tom, Beth

## Fourth Phase

A set state sets the object to the fourth phase.

	<b>Default (property set to false)</b>	<b>property set to true</b>
Design Engineer	Kristin, Flavio, Bill, Galen	Kristin, Flavio, Bill, Galen, <i>Jeff, Michelle</i>
Project Manager	Dave, John	Dave, John
QA Engineer	Sean, Sachin, Iyrena	Sean, Sachin, Iyrena
Reviewer	Kristin	Jane, Kristin, Flavio, Dave, John, Sean, Sachin, Dan
Submitter	Jeff	
Promoter		
Observer	Tom, Beth	Tom, Beth, Lynn, Jane
Team Leader	Tom, Beth	Tom, Beth

Observations:

The Design Engineer and Observer do not exist in the life cycle template for the fourth phase, but they do exist in the team and context team. So even if the property is set to true, these roles are not removed from the run-time team.

## Defining Team Properties and Roles

**Note:** The Team Administrator is not accessible from Windchill ProjectLink.

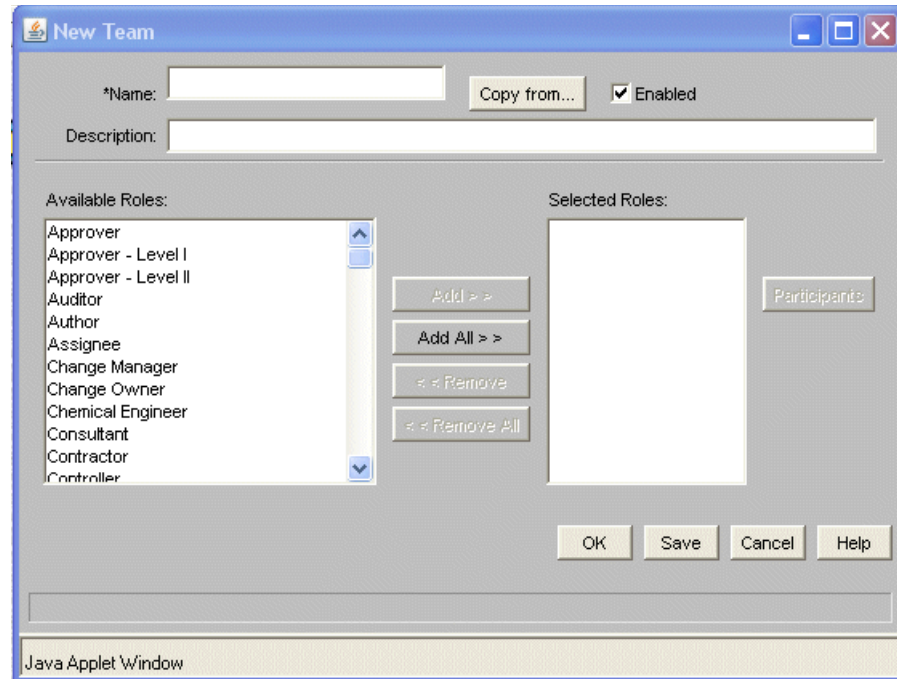
For all other Windchill solutions, you can access the **Team Administrator** page in one of the following ways:

- From the **Product**, **Library**, **Organization**, and **Site** tabs, click **Utilities**. Click **Team Administrator** to access the Team Administrator.
- From the **Product** and **Library** tabs, click **Templates**. On the **Templates** table, select **Team Templates** from the **Current View** drop-down list. Click **Administer Team Templates**.

The **Team Administrator** page displays a list of existing teams. The Team Administrator refers to team templates as teams. Use the buttons on this page to create, update, view, delete, rename, and save as a new team. Click **Help** on the Team Administrator page for a description of the buttons and their functions.

Click **New** or **Edit** to access the **New/Edit Team** window.

The following figure shows the **New Team** window, with properties entered:



The fundamental task in defining a team is selecting the roles that compose the team. The **Available Roles** list is populated with the names of all roles available in the system.

To add a role to a team, select it and click **Add** to move it to the **Selected Roles** list. You can also click **Add All** to move all the available roles to the **Selected Roles** list. Click **Remove** or **Remove All** to delete roles from the team.

Leave the **Enabled** check box selected to make the team selectable. Typically, a team is disabled only when you want to remove it. A disabled team remains in effect for all current usages, but cannot be selected for new objects.

## Predefined Roles

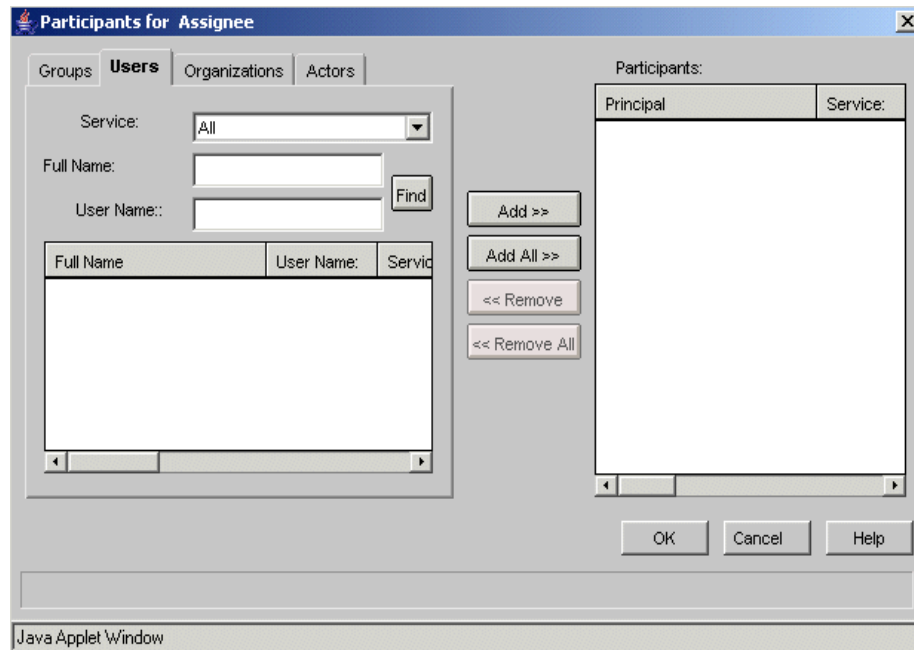
Windchill includes many predefined life cycle states and roles. You can define additional states and roles by adding them to the RoleRB.rinfo resource files. Defined states and roles are added to this list when you recompile the resource files and deploy the class files to your production environment. For additional information, see the enumerated types information in the *Windchill Customizer's Guide*.



**Caution:** When you add a value to an enumerated type (for example, by adding a role in the RoleRB.info resource file), removing that value can result in a serious runtime error. Do not remove a role unless you are certain there is no reference to it.

## Assigning Participants to Team Roles

To associate principals (users, groups, organizations) or actors with a team role, select a role from the **Selected Roles** list in the **Edit Team** window, and click **Participants**. This opens the **Participants** selection window, shown in the following figure, where you can map users, groups, organization, or actors to the selected role:



**Note:** The **Group Members Display** preference found in the site context under the **Promotion Process** category, controls the display of all the group members in the Participants table. When the preference is set to Yes, the **Participants** table shows all the members (users and sub-groups) of the existing groups in the table. The default for this preference is set to No.

Use the following information to search for groups, users, organizations, and actors:

- To search for groups, click the **Groups** tab. Select **All** or a directory service from the **Service** drop-down list, to narrow your search. Enter the name of the group in the **Group** field, and click **Find**. You can use the \* wild card to search for groups with similar names. The specified services are searched and the matching results display in the table.
- To search for users, click the **Users** tab. Select **All** or a directory service from the **Service** drop-down list to search the entire system. To search for a specific user, enter information in the **Full Name** or **User Name** fields and click **Find**. You can use the \* wild card to search for users with similar names. The specified services are searched and the matching results display in the table.

- To search for organizations, click the **Organizations** tab. Select **All** or a directory service from the **Service** drop-down list to search the entire system. Enter the name of the organization in the **Organization** field, and click **Find**. You can use the \* wild card to search for organizations with similar names. The specified services are searched and the matching results display in the table.
- To search for actors, click the **Actors** tab. To assign a role, select an actor. Currently, Creator is the only actor defined. The Creator is resolved at runtime to the user who created the selected object.

To map a participant to a role, select it and click **Add** to move it to the **Participants** list. You can also click **Add All** to move all the displayed users, groups, organizations, or actors to the **Participants** list. Click **Remove** or **Remove All** to delete participants from the list.

From the **Change Request** information page, users can edit the team using the **Participants** selection window described previously. The current team is displayed and members of the context team cannot be removed.

For detailed instructions for selecting participants, see the online help by clicking **Help**.

## Manually Selecting and Resetting Team Templates

In Windchill PDMLink, during the creation of a part and a document, the team template can be manually selected. It is also possible to reset a team template on an existing part or document. In most cases, the manual selection can be done from either a drop-down list or a Search function.

### Manual Selection of Team Templates

The action to manually select the team template during the creation of a part and a document is available to users that have administrative privileges set on the object. For more information, see the [Access Control](#) chapter.

### Resetting Team Templates

The action to reassign team templates on existing parts and documents is only available in Windchill PDMLink. Users must have administrative privileges set on the object for either of these actions to be available. For more information, see the [Access Control](#) chapter.

Additionally, resetting the team template is only available for objects with advanced life cycles.

# Team Best Practices

The following sections provide some best practices for teams.

## Using Local and Shared Teams

To manage your user community most effectively, analyze how best to organize your users at the organization level. Where possible, create user-defined groups and shared teams in each organization context and then use those groups and teams when creating application contexts.

When deciding which roles should be used in shared and local teams, be sure to check the team templates that will be used so you can match the roles in the context teams with the roles used in the team templates as appropriate.

## Using Team Templates

Site, organization, and product and library administrators manage team templates.

- Site administrators create, modify, delete, and view team templates in the site context.
- Organization administrators create, modify, delete, and view team templates in the given organization context. Organization administrators can view team templates from the site context.
- Product and library managers create, modify, delete, and view team templates in the given application context. They can view team templates from the parent organization context and the site context. The Team Administrator is not available to administrators of program or project contexts.

The Team Administrator client displays a table that lists all team templates belonging to the given context plus those belonging to its ancestor contexts. A column in the table identifies the context owning each team template.

**Note:** Although you are allowed to create team templates using users, it is best to create them with groups.

## Refreshing User-Defined Groups

User-defined groups can be created and maintained in organization contexts (using the **Groups** link or the Principal Administrator) and in the site context (using the Principal Administrator), or created and maintained through an enterprise directory server.


When a user-defined group is added as a participant of a context team role, the group's members (users) are added individually to the role and tracked independently as participants. This means that if the group membership is changed or the group is deleted, the context teams must be refreshed so that the role participants properly reflect the current members of the group.



In order to effectively refresh the groups that are members of teams, you should consider doing the following:

- Define user-defined groups that are common to products, libraries, and projects at the organization level, rather than at the product, library, or project instance level.
- Reference the user-defined groups in the definition of each of the context teams, where appropriate.
- When user-defined groups are referenced in context teams, refresh the teams periodically to reflect the dynamic group membership.

A site or organization administrator can refresh groups in one of the following ways:

- Manually refresh user-defined groups by:
  - Accessing the **Members** table from every project, program, product, or library in which the group is participating, and clicking the synchronize team with groups icon  .

This is the only manual refresh method available if user-defined groups are defined in the site context, and it can also be used to refresh groups defined in an organization context.

This icon is only available to site and organization administrators, which means that at least one site or organization administrator must be a participant in every context team. When the membership in a group is changed, an administrator then goes to each context team and synchronizes the groups.

- Accessing the **Groups** page of the organization and clicking the **Refresh Teams** action for an updated group.

This refresh method is only available for groups that are defined within an organization context.

- Schedule the automatic update of teams through the use of a queue, where teams for which the group membership has changed can be updated on a daily basis in an off-peak time of your choice. Additionally, a schedule can be set up to recompute the membership of all context teams once a week.

Schedule the refreshing of groups by:

- Enabling the scheduled refresh functionality.
- Setting up a schedule for refreshing the group memberships in context teams.

Using the xconfmanager utility, you can set the following properties in the wt.properties file to turn on the scheduling of team updates:

`wt.inf.team.useScheduledRefreshGroups`

Specifies whether or not to use a queue to refresh container teams that contain groups that have been modified. By default, this property is set to false and daily automatic team updates do not happen. Set this property to true to turn on this feature.

When this feature is turned on, all teams containing a modified group get marked as needing to be refreshed and then, at the designated time each day, only those teams are refreshed. By default, the designated time of day is 1 AM (formatted as 01:00:AM). You can use the `wt.inf.team.refreshGroupsDailyQueueTime` property to change the time of day.

`wt.inf.team.useScheduledRecompute`

Specifies whether or not to use a queue to recompute the membership of all container teams. By default, this property is set to false and weekly team updates do not happen. Set this property to true to turn on this feature.

When this feature is turned on, all teams are updated at a specified day and time each week. By default, the designated day is Sunday (indicated by 1) and the time of day is 1 AM (formatted as 01:00:AM). You can use the `wt.inf.team.recomputeMembershipDayOfWeek` property to change the day and the `wt.inf.team.recomputeMembershipWeeklyQueueTime` property to change the time of day.

For additional information about the properties, see the `properties.html` file.

**Note:** Because recomputing team membership uses a lot of system resources that can slow down system performance, scheduling team updates is the recommended method to use.

# 10

## Context Templates

This chapter provides details on how to create and manage context templates.

<b>Topic</b>	<b>Page</b>
Creating Context Templates .....	10-2
Methods of Creating Additional Context Templates .....	10-4
ZIP File Contents of an Exported Context Template .....	10-6
Required Contents of ZIP File Used for Importing a Context Template .....	10-7
Creating Business XML Files for Context Templates .....	10-9
Managing Context Templates .....	10-10

## Creating Context Templates

An initial set of context templates was loaded when your Windchill solution was installed. For a list of the loaded templates, see [Installed Site Templates](#) in the Contexts chapter. For details about the loaded templates, see the following:

- For organization templates, see [Out-of-the-box Organization Templates](#).
- For product and library templates, see [Out-of-the-box Product and Library Context Templates](#).
- For program and project templates, see [Out-of-the-box Project and Program Templates](#).

The XML elements in the template load files are basically the same as those used for the import and export functionality. However, there are some variations and the load files are not meant to be used directly when you create new context templates.

Only site administrators can create organization templates. Site and organization administrators can create product, library, program, and project templates.

The following sections describe the contents of the files you can use as input when you create context templates. See the [Methods of Creating Additional Context Templates](#) section for creation details.

**Note:** The descriptions in the following sections assume you are familiar with XML. The XML files that are used must be coded against the standardX10.dtd. You can download the standardX10.dtd by downloading any of the context templates. The DTD is contained in the downloaded ZIP.

### Organization Templates

The only option available for creating an organization template is to create the input file manually and then, from the **Site** tab, use the **Create Template** icon from the **Templates** table that shows organization templates to create the template.

The input file used when creating an organization template can be just the business XML file if no additional content is needed or a ZIP (or JAR) file that has both the business XML file and other content files named in the XML file.

You can include the following administrative items in a new organization template:

- Additional domains, folders, user-defined groups, and access control rules for the organization
- Additional product, library, program, and project templates that are available from the organization

- Object initialization rules
- Project roles (if the organization is used with Windchill ProjectLink)

For details on the XML elements required in the business XML file for organization templates, see [Creating Business XML Files for Organization Templates](#) in the Creating XML Files for Templates and Shared Teams appendix.

## Product and Library Context Templates

When using any of the options for creating templates, you can include the following administrative items in a new product or library template:

- Domains
- Folder structure (not including the folder contents)
- Team roles (including role access to actions)
- Team members
- Document templates
- Object initialization rules
- Access control policy rules
- Template visibility
- Preferences

For information on the input file used when creating a context template, see [Required Contents of ZIP File Used for Importing a Context Template](#).

## Program and Project Context Templates

When using any of the options for creating templates, you can include the following administrative items in a new program or project template:

- Domains
- Folder structure (not including the folder contents)
- Folder links and structure
- Project or program plan
- Deliverables
- Team roles (including role access to actions)
- Team members
- Documents
- Document templates

- Discussion specification
- Discussion template
- Object initialization rules
- Template visibility
- Shared objects in a project (if any have been created in an existing project that is exported)
- Preferences

Additionally, you can save an existing project (or program) as a new project (or program). Using this option creates a duplicate of a project (or program) under a new name.

For information on the input file used when creating a context template, see [Required Contents of ZIP File Used for Importing a Context Template](#).

## Methods of Creating Additional Context Templates

You can create additional context templates in the following ways:

- By creating a context template input file and then using the **Create Template** icon from the **Templates** table to create the template. This option is available for all contexts.
- By saving the contents of an existing context as a new context template. This option is only available for application contexts.
- By exporting an existing application container as a template, updating the template contents (if needed), and then importing the new template. This option is only available for application contexts.

The following actions are available from the information page for site and organization administrators:

- The **Save as Template** action provides a way to create a new template from the current context.

A newly created product, library, program, or project template is saved in the application container's organization context and appears in the **Templates** table of that context when the **Current View** is **<context> Templates**.

- The **Export as Template** action allows you to export the current product, library, program, or project context as a template, creating a ZIP file on your system.

For the details on the contents of the exported ZIP file, see [ZIP File Contents of an Exported Context Template](#).

You can import an exported template by navigating to either the Site or the organization under which you want the template imported and clicking the **Templates** link. From the **Templates** table that appears, select the type of template from the **Current View** and click **Import Template**. For the details on the required contents of the ZIP file that is imported, see [Required Contents of ZIP File Used for Importing a Context Template](#).

The **Import Template** action creates a template with the same name as the exported template. If you want to specify a different name for the template, you can do either of the following:

- Choose **Create Template** instead of **Import Template** to import the exported template. On the window that displays, you specify the name and description of the template. This information is used in place of the name and description in the ZIP file.
- Modify the contents of the TAG-0.xml file that is in the exported ZIP file to change the name and description of the template and then use **Import Template** to import the exported template. The following section describes the contents of the exported ZIP file.

You can use **Create Template** to create a context template. The input file you provide in the create operation is not the same as the file used for importing. You can use a file that you manually create; however, PTC recommends that you always start with a business XML file that you have either downloaded or located in the Windchill installation directory. If you do use one of the Windchill load files as your starting point, PTC recommends that you do not modify the ENTITY tags in the file nor modify the content of any of the XML files referenced in the ENTITY tags.

The import and creation functions do not validate that the input XML conforms to the DTD; you must do this before doing the import or create function. The [Creating XML Files for Templates and Shared Teams](#) appendix describes the elements that you can use in each business XML that you can use as input.

## ZIP File Contents of an Exported Context Template

The ZIP file created on your system from an **Export as Template** action contains the framework options you selected. Depending on your system configuration, you may be prompted to save the ZIP file or the file may be saved on your local drive with the name `<context_name>_generated_template<integer>.zip`, where `<context_name>` is the name of the product, library, program, or project being exported and `<integer>` is an integer that ensures that the file has a unique name when the download occurs. The name used is not significant and can be changed.

The ZIP file contains the following:

- The `importList.txt` file, which is not currently used during an import and can be ignored.
- The `TAG-ContainerTemplate_<integer>.xml` file, which contains information about container template itself, such as name and container class of the template. Here also, `<integer>` is used to ensure that the files are unique.
- The `TAG-0.xml` file (which is in the `CONTENTS` directory) lists the contents of the exported template. This file is also known as the business XML file and contains information that is used when creating a container from this template (for example, folder structure, access control policy rules, and so on).
- Additional files that make up the contents of the exported template. If unzipped, these files are put in the `CONTENTS` directory. For example, the ZIP file of a program or project template could contain documents and document templates that then belong in the `CONTENTS` directory.



## Required Contents of ZIP File Used for Importing a Context Template

The ZIP file you use to import a context template must have the same basic structure as the ZIP created when exporting a context template (as described in the previous section). The ZIP file must contain the following:

- An XML file at the top level of the ZIP that contains the definition of the context template itself, such as name and container class of the template. This corresponds to the TAG-ContainerTemplate\_<integer>.xml file that is generated in an export. For details on the content of this file, see [Contents of Top-level XML File for Imported Templates](#).
- The business XML file (which is defined as coming from the CONTENTS directory) that lists the contents of the template you want to import and contains information that is used when creating a container from this template (for example, folder structure, access control policy rules, and so on). This corresponds with the TAG-0.xml file that is generated on export. The name of this file must match the file named in the zipEntry element in the top-level XML file. For details on the content of the business XML file, see the [Creating Business XML Files for Context Templates](#) appendix.
- Additional files to include in the contents of the template you want to import. These files are named in the business XML to be added as content items through the use of the contentItem element. Before creating the ZIP file, put these files in the CONTENTS directory. Examples of files that can be in this directory are documents and document templates that you want to import as part of a Windchill ProjectLink project template.

**Note:** The file used could also be a JAR file rather than a ZIP file.

**Note:** The ZIP file created by selecting the **Download** action from a context template row on the **Templates** table cannot be used as the template data file for creating a context template using the **Import Template** action; it does not contain all of the required information that is needed to create a new context template. The downloaded ZIP file can be used to update an existing context template or create a new template using **Create Template**.

## Contents of Top-level XML File for Imported Templates

The top-level XML file is used to define the template that you are importing. The file must begin with the following lines:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ContainerTemplate SYSTEM "standardX10.dtd">
```

The content of the top-level XML file is defined by the following DTD element:

```
<!ELEMENT ContainerTemplate (name,
                             description?,
                             containerClassName,
                             defaultValuesRule?,
                             locale?,
                             enabled?,
                             (xmlText | loadPath | zipEntry)?)>
```

The following example illustrates how to use the ContainerTemplate element to define the template you want to import. The example is for a Windchill PDMLink product template as signified by the use of wt.pdmlink.PDMLinkProduct in the containerClassName element.

In the example, the name of the template is DEMO Product Template and, when the template is imported, it is enabled and visible. The business XML file that contains the content of the template is contained in the CONTENTS/DEMOProductTemplate.xml file that is in the imported ZIP file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ContainerTemplate SYSTEM "standardX10.dtd">
<ContainerTemplate>
  <name>DEMO Product Template</name>
  <description>A description of the DEMO template</description>
  <containerClassName>wt.pdmlink.PDMLinkProduct</containerClassName>
  <enabled>true</enabled>
  <zipEntry>CONTENTS/DEMOProductTemplate.xml</zipEntry>
</ContainerTemplate>
```

**Note:** Although the DTD element shows that you can specify the business XML file in multiple ways (through xmlText | loadPath | zipEntry), only the use of the zipEntry element is supported when you are importing a template.

To specify a project template instead of a product template, use wt.projmgmt.admin.Project2 in the containerClassName element. For a library template, use wt.inf.library.WTLibrary. For a program template, use wt.projmgmt.admin.Project2.Program. For another example, see [ContainerTemplate Element](#) in the Creating XML Files for Templates and Shared Teams appendix.

# Creating Business XML Files for Context Templates

Business XML files are used as input for the **Import Template** and **Create Template** actions (see [Methods of Creating Additional Context Templates](#)):

- The **Import Template** requires that the business XML file and additional files (as described in [Required Contents of ZIP File Used for Importing a Context Template](#)) be packaged in a ZIP or JAR file.
- For the **Create Template** action, the business XML file can be used alone as the input when no additional content files are referenced in the business XML file, or it can be used in a ZIP or JAR file when additional content files are referenced in the XML file.

All business XML files must start with the following lines:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE context_type SYSTEM "standardX10.dtd">
```

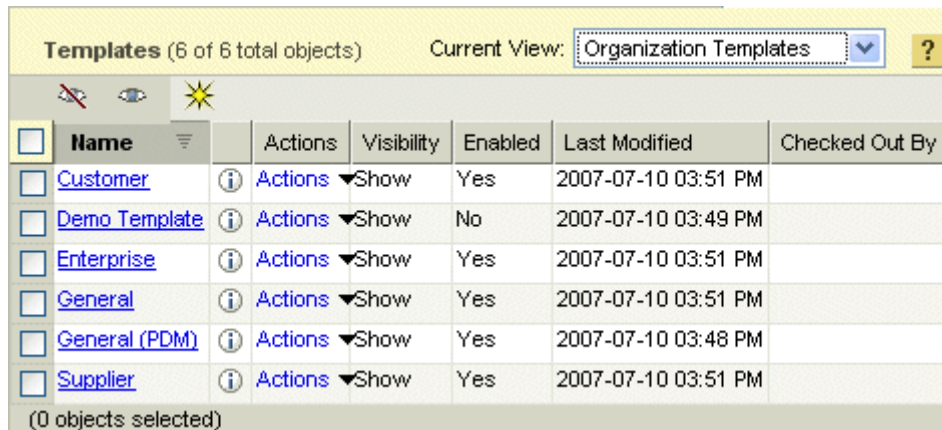
where *context\_type* is one of the following:

- **OrganizationConfig** for organization templates
- **ProductConfig** for product templates
- **LibraryConfig** for library templates
- **ProjectConfig** for project and program templates

For the details on the XML elements supported in the business XML files, see the [Creating XML Files for Templates and Shared Teams](#) appendix.

## Managing Context Templates

To view the set of context templates that are available from a context, click **Templates** from the links located below the tab. The **Templates** table displays in the window. From the Current View drop-down list, select the type of template. For example, the following picture shows the resulting **Templates** table when you click **Templates** from the **Site** tab and the select **Organization Templates** from the drop-down list:



The screenshot shows a web interface for managing templates. At the top, it says "Templates (6 of 6 total objects)" and "Current View: Organization Templates". Below this is a table with columns: Name, Actions, Visibility, Enabled, Last Modified, and Checked Out By. The table lists six templates: Customer, Demo Template, Enterprise, General, General (PDM), and Supplier. Each row has a checkbox, an information icon, and a link to "Actions". The "Visibility" column shows "Show" for all templates. The "Enabled" column shows "Yes" for all templates. The "Last Modified" column shows dates and times. The "Checked Out By" column is empty for all templates. At the bottom left, it says "(0 objects selected)".

<input type="checkbox"/>	Name	Actions	Visibility	Enabled	Last Modified	Checked Out By
<input type="checkbox"/>	<a href="#">Customer</a>	<a href="#">Actions</a>	Show	Yes	2007-07-10 03:51 PM	
<input type="checkbox"/>	<a href="#">Demo Template</a>	<a href="#">Actions</a>	Show	No	2007-07-10 03:49 PM	
<input type="checkbox"/>	<a href="#">Enterprise</a>	<a href="#">Actions</a>	Show	Yes	2007-07-10 03:51 PM	
<input type="checkbox"/>	<a href="#">General</a>	<a href="#">Actions</a>	Show	Yes	2007-07-10 03:51 PM	
<input type="checkbox"/>	<a href="#">General (PDM)</a>	<a href="#">Actions</a>	Show	Yes	2007-07-10 03:48 PM	
<input type="checkbox"/>	<a href="#">Supplier</a>	<a href="#">Actions</a>	Show	Yes	2007-07-10 03:51 PM	

(0 objects selected)

From the **Templates** table, you can perform the following tasks:

- Create new templates.
- Filter the visibility of one or more of the templates.
- Enable and disable templates.
- Edit template information.
- Check out and then check in updated templates.

For a template to be available for use, it must be both visible and enabled.

The following sections provide some additional information about the tasks. For task details, see the online help that is available from the **Templates** table.

## Filtering Template Visibility

You can determine a user's ability to see the templates that are listed in the **Templates** table using the show and hide icons that are located above the table rows or using the **Show** and **Hide** actions in individual rows of the table.

When you hide a template's visibility, the template is no longer displayed when creating a context of that type in the applicable context. For example, assume that you hide the Enterprise organization template from the site **Templates** table. Then when creating an organization context, the Enterprise template is not listed in the **Template** drop-down list.

If a template is hidden in a parent context, it is also hidden in all child contexts unless you explicitly change the visibility to show the template in the child context. For example, assume you have hidden a product or project context template from the site **Templates** table. Then all organization contexts will also show the template as hidden. However, you can navigate to an organization context and show the template in that organization context.

## Enabling Templates

Enabling a template means that the template is ready for use.

All loaded templates are, by default, enabled.

When you create a new template, you can choose to enable or disable it.

Enabling or disabling a template enables or disables the template in all contexts where it is available; you cannot enable a template in one context and disable it in another.



# 11

## Object Initialization Rules

This chapter introduces the concepts for object initialization rules and provides reference information that you can use when creating and managing the rules, including rules for numbering and versioning.

For details on the XML syntax that is used in the rules, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

Topic	Page
Overview .....	11-2
Loaded Object Initialization Rules.....	11-3
Adding and Changing Object Initialization Rules .....	11-3
Accessing the Object Initialization Rules Administrator .....	11-3
How Object Initialization Rules Work.....	11-4
Determining the Composite Rule.....	11-5
Specifying Rules in the Object Initialization Rules Administrator .....	11-7
Defining the Content of XML Documents used for Object Initialization Rules.....	11-8
Specifying Object Initialization Rules in a Context Template.....	11-20
Customizing Rules .....	11-20
Changing Numbering Schemes .....	11-21
Changing Versioning Schemes .....	11-24

# Overview

The use of object initialization rules provides a way to specify the following:

- Attribute value generation -- The value defined for a specific attribute associated with an object type can be a constant or a value that is generated at runtime.

Some clients, such as the Windchill workgroup managers, use attribute value generation to request a value for an attribute (such as a number) before an object is stored in the database. For the additional information, see [CAD Document Numbering](#).

- Default values for object initialization -- When objects are being stored in the database, Windchill uses the object initialization rules that have been set to set the values for the attributes that do not already have a value.

The default values are used only when the user interface creating the object does not set a value or when the user interface requests a server generated value. For example, you can set the default values for folder paths, life cycle templates, numbering, and versioning that are then used in generating the values for the user interface. The functionality in the current user interface for creating objects provides the intended results; modifying the out-of-the-box rules that are set may affect the functionality if the value is not set by the user interface.

- Display characteristics that constrain attribute values -- If the user interface checks for the constraints, they are then used when the attribute value is displayed in the user interface. For example, the constraints for numbering is honored in the user interface that creates parts (object type: WTPart).

The out-of-the-box constraints include the ability to:

- display a value that the user cannot change (immutable).
- obtain a value that can be displayed in the interface as the default value (server pregenerated).
- not display a generated value, but only display the word generated (server assigned).
- not display the attribute label or value (hidden).

These specifications for the attributes are called rules. Each rule can contain the specification of both attribute values and constraints for one object type. The rules are created using XML elements.

Additionally, you can qualify when values are set or when constraints are used for Windchill attributes by integrating the use of conditional logic algorithms in object initialization rules.



The following sections provide information on rules that are loaded during installation, how to add rules, how rules work, and the content of the rules (including information on numbering and versioning).

## Loaded Object Initialization Rules

A set of rules is loaded during installation and additional sets are included in the context templates that are loaded.

The rules that are loaded are named according to the object type to which the rule applies and appear with a site context. For additional information, see [Installed Site Object Initialization Rules](#). The user interface supports rules that are set during the load.

**Note:** PTC recommends that you fully understand the out-of-the-box object initialization rules that are set through the load files before you make any rule changes. You can view the rules that are loaded from the Site **Utilities** page. For more information, see [Object Types and Attributes Used with AttrValue and AttrConstraint](#).

## Adding and Changing Object Initialization Rules

You can add and change rules as follows:

- By using the Object Initialization Rules Administrator.
- By creating additional context templates or updating existing templates. See [Specifying Object Initialization Rules in a Context Template](#).

## Accessing the Object Initialization Rules Administrator

You can access the Object Initialization Rules Administrator, as follows:

- From the program, project, organization, and Site **Utilities** pages (if Windchill ProjectLink is installed).
- From the product, library, organization, and Site **Utilities** pages (for all other Windchill solutions).

If you have access to the **Object Initialization Rules Administrator** link, then you can use it. Clicking the link displays the **Object Initialization Rules** table where all rules that have been added in the current context and ancestor contexts are listed.

## How Object Initialization Rules Work

Where a rule is created determines which objects are affected by the rule:

- Rules set during installation affect all objects of the types specified in the load file.
- Rules set from the site **Utilities** page affect all objects of the types specified that are initialized in the site context after the rule is set.
- Rules set from the organization **Utilities** page affect only those objects of the types specified that are initialized by a user from the organization under which the rule was set.
- Similar restrictions are in place for those rules set for program, project, products, and libraries, where the rules only apply to the program, project, product, or library under which the rules are set.

Rules for an object type that are set at one context do not replace other rules that are set at a higher context, but all rules are merged to create a composite rule that is based on the object type hierarchy and the context hierarchy. For example, a rule for WTPart numbering and versioning can be set at the Site context and a rule for WTPart folder paths can be set at the product or organization context. Then the composite rule for WTPart objects created under the product or organization includes both the setting for numbering and versioning, and the setting for folder paths. If the product rule had included setting the numbering scheme, then this rule setting would take precedence over the setting made at the Site context.

If a default value is not set for an object attribute in the composite rule that is in place and the user creating the object does not specify a value for the attribute, then one of the following occurs:

- If the underlying code provides a default, then it is used. For example, if the rule does not set the default life cycle state, then the Life Cycle service would use its property value to set a default state.
- If there is no underlying code that provides a default, then either the attribute value is set to NULL or an exception is thrown.

## Determining the Composite Rule

To determine what object initialization rule is in effect for an object type in a specific context, you can generate the composite rule for the object type.

Use the following steps to generate and display the composite rule:

1. Access the Object Initialization Rules Administrator from the context in which you want the composite rule generated, as described in [Accessing the Object Initialization Rules Administrator](#).

2. Click **Download Composite** .

The **Download Composite Object Initialization Rule** window opens.

3. Enter a valid object type. Valid object types are of any of the object types listed in the **Object Type** column of the **Object Initialization Rules** table.
4. Click **OK** to initiate the download process.

or

Click **Cancel** to cancel the process.

How the download occurs depends on how your system is configured. Your system may be set up to display the XML in an XML editor or a browser. From the display, you can save the XML; otherwise, you may be prompted to save the XML in a file.

The following XML shows a sample composite rule for the wt.fc.WTObject object type (formatted to fit on this page).



**Caution:** Do not modify or remove the wt.fc.WTObject rule.

```
<AttributeValues objType="wt.fc.WTObject">

  <AttrValue id="lifeCycle.id"
    algorithm="com.ptc.core.foundation.lifecycle.server.impl.
      LifecycleTemplateAttributeAlgorithm"
    ignore="false"
    force="false"
    final="false">
    <Arg>Basic</Arg>
  </AttrValue>

  <AttrConstraint id="lifeCycle.id"
    algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints"
    ignore="false"
    force="false"
    final="false">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint">
    </Value>
    <Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint">
    </Value>
  </AttrConstraint>

  <AttrConstraint id="lifeCycle"
    algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints"
    ignore="false"
    force="false"
    final="false">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint">
    </Value>
    <Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint">
    </Value>
  </AttrConstraint>

</AttributeValues>
```

In this rule, the following wt.fc.Object attribute default value is set:

- The lifecycle.id default value is set to Basic.

The composite rule also sets the following constraints on the lifecycle.id and lifecycle attributes that are displayed in the user interface where the wt.fc.WTObject is used:

- The attribute is assigned by the server when the user clicks OK.
- The attribute cannot be changed by the user.

## Specifying Rules in the Object Initialization Rules Administrator

An object initialization rule is specified in an XML document that is formatted according to the object initialization rules DTD.

You can download this DTD from the **Object Initialization Rules** table, as follows:

1. Access the Object Initialization Rules Administrator from the context in which you want to create the rule, as described in [Accessing the Object Initialization Rules Administrator](#).

2. From the **Object Initialization Rules** table, click **Download DTD** .

Your system may be set up to display the DTD in an XML editor or a browser. From the display, you can save the DTD; otherwise, you may be prompted to save the DTD in a file.

To specify rules through the Object Initialization Rules Administrator, complete the following steps:

1. Create the XML document for the rule you want to create, validating it against the DTD you can download, as described in the earlier procedure.

The content of the XML document is described in the next section.

2. Access the Object Initialization Rules Administrator from the context in which you want to create the rule, as described in [Accessing the Object Initialization Rules Administrator](#).

3. Click **Create Rule** .

The **Create Object Initialization Rule** window opens.

4. Enter values in the fields provided.

For help on what to enter in the fields, click the help icon; browse to the XML document you created in Step 1 for the value to specify in **XML File** field.

5. Click **OK** to create the rule.

## Defining the Content of XML Documents used for Object Initialization Rules

This section describes the content of XML documents used when specifying rules through the Object Initialization Rules Administrator, as described in the previous section.

Object initialization rules can also be set through the use of templates; however, additional elements are needed before and after the rules to define import information that is needed in an XML document that can be imported as a template. The format of the rules themselves is the same everywhere. For information on the elements required in XML documents for importing templates, see the [Specifying Object Initialization Rules in a Context Template](#) section.

For each object initialization rule, the XML document must contain the following:

- The AttributeValue tag which identifies the type of object. Use the logical identifier of the type as the object type. See [Windchill Types in AttributeValues](#).
- Nested inside the AttributeValues tag must be at least one of the following:
  - An AttrValue tag that identifies an attribute for which a default value is being set and specifies the algorithm that is used to set the default value.
  - An AttrConstraint tag that identifies an attribute for which a constraint is defined and specifies the algorithm that is used to apply the constraint.

**Note:** The constraint definitions provided through the AttrConstraint tag do not impact the behavior of setting default values provided through the AttrValue tag. The constraints are only used to control the behavior of the user interface.

For details on the syntax required for the tags and optional elements that can be included in the tags, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

An example wt.doc.WTDocument rule specification that defines a default folder path and displays the folder path value as a server generated value that can be manually changed is as follows:

```
<AttributeValues objType="wt.doc.WTDocument">
  <AttrValue id="folder.id"
    algorithm="com.ptc.core.foundation.folder.server.
      impl.FolderPathAttributeAlgorithm">
    <Arg>/Default</Arg>
  </AttrValue>
  <AttrConstraint id="folder.id"
    algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue"/>
  </AttrConstraint>
</AttributeValues>
```

The object types, attributes, and algorithms that you can include in rule tags are described in the following sections.

## Windchill Types in AttributeValues

Use the logical identifier of an object type when specifying an object type in the AttributeValues tag. To determine what the logical identifier of a type is, view the type information from the Type and Attribute Manager. The logical identifier of a type can be set when out-of-the box types are loaded or can be set when you create (or update) a type through the Type and Attribute Manager. For example, the logical identifier of the soft type named Dynamic Document is defined when the type is loaded and, when viewed from the Type and Attribute Manager, has the following format:

```
<prefix>.DynamicDocument
```

where *<prefix>* is generated by reversing the internet domain specified for the default organization during the installation. In this case, the logical identifier is the same as the type name. For additional information on types and the use of the logical identifier, see the [Type and Attribute Manager](#) chapter.

## Object Types and Attributes Used with AttrValue and AttrConstraint

The list the out-of-the-box combinations of object types and attributes for which default values are provided using the AttrValue tag and constraints are provided using the AttrConstraint tag can be determined by accessing the Object Initialization Rules Administrator from the site and organization contexts established after your Windchill solution is installed and configured (see [Accessing the Object Initialization Rules Administrator](#)). Additionally, some out-of-the-box context templates set object initialization rules. After creating a context, the out-of-the-box object initialization rules that are set are listed in the **Create Object Initialization Rule** window that is accessible from the context.

The default values and constraints specified in the out-of-the-box object initialization rules can be used without any customization of the user interface. Any additional default values and constraints that you want to add can require customization. See [Customizing Rules](#).

**Note:** The object initialization rules that are set for default values only apply when the Windchill solution that is used to create an object does not set a corresponding value. The default values set in the rules can also be displayed in the user interface if the rule includes the use of a constraint that gets the default value before the user interface displays. For example, the user interface that is used to create a document displays the default value for the folder.id attribute in the **Location** field. See [Folder Constraints Example using GetServerPreGeneratedValue](#).

For details on the specific syntax to use in the AttrValue and AttrConstraint tags, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

## Algorithms Used with AttrValue

You specify an algorithm for each attribute specified in the AttrValue tag. Out of the box, Windchill provides the following algorithms (names are shown on multiple lines in the table; enter the tag, including the name of the algorithm, on one line):

Algorithms for Use with AttrValue Tag	Description
com.ptc.core.foundation.folder.server.impl. FolderPathAttributeAlgorithm	Converts folder path into a type instance identifier of the specified folder.
com.ptc.core.foundation.lifecycle.server.impl. LifeCycleTemplateAttributeAlgorithm	Converts the specified life cycle name into a type instance identifier of the life cycle.
com.ptc.core.foundation.team.server.impl. TeamTemplateAttributeAlgorithm	Converts the specified team template name into a type instance identifier of the team template.
com.ptc.windchill.enterprise.revisionControlled. server.impl.NumberGenerator	Implements the numbering scheme identified in the generator function that is specified in the argument.
com.ptc.core.foundation.vc.server.impl. VersionInfoGenerator	Implements the versioning scheme that is specified in the argument.
wt.rule.algorithm.StringConstant	Converts the specified value into a string. This algorithm is not currently used, but is provided as a convenience for those doing customizations.
wt.rule.algorithm.EnumTypeConstant	Converts the specified value into an enumerated value. This algorithm is not currently used, but is provided as a convenience for those doing customizations.

For the specific syntax to use in the AttrValue tag, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

Numbering attributes require that you specify a generator function as the argument. The function is used to generate the numbers. For additional information on numbering, see [Changing Numbering Schemes](#).

Versioning attributes require that you specify a series as the argument. For additional information on versioning, see [Changing Versioning Schemes](#).

The following sections provide some examples that use the other algorithms.



## Default Folder Path Example using the FolderPathAttributeAlgorithm

The following example shows the out-of-the-box XML that sets the default folder path that is used when creating a document:

```
<<AttributeValues objType="wt.doc.WTDocument">
  <!-- set the folder -->
  <AttrValue id="folder.id" algorithm="com.ptc.core.foundation.folder.server.impl.
    FolderPathAttributeAlgorithm">
    <Arg>/Default</Arg>
  </AttrValue>
</AttributeValues>
```

This example shows the use the following XML to specify the default document folder path:

```
<Arg>/Default</Arg>
```

/Default in this argument is used to indicate the top-level folder that is the listed in the folder browser on the **Folders** page of any context. The name of the top-level folder is the same as the name of the context. When setting the default value for folder.id, always start the path with /Default. To name a folder under the top-level folder, add a slash and the folder name after /Default. For example, to set the default value to the Design folder, include the following argument:

```
<Arg>/Default/Design</Arg>
```

## Default Life Cycle Example using the LifeCycleTemplateAttributeAlgorithm

The following example shows the out-of-the-box XML that sets the default life cycle template to the localized key of "BASIC\_LIFECYCLE\_NAME" (which is Basic in English) for a document. The locale used to determine what is displayed is the locale set on the server (not the locale set on the client).

This example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
  <!-- set the lifecycle -->
  <AttrValue id="lifeCycle.id" algorithm=
"com.ptc.core.foundation.lifecycle.server.impl.LifeCycleTemplateAttributeAlgorithm"
>
  <Arg>
  <!-- Translation of the word "Basic" must be the same as the translation -->
  <!-- done in commonLifeCycles.xml -->
  <?loc-begin key="BASIC_LIFECYCLE_NAME" maxlen="30"?>Basic<?loc-end?>
  </Arg>
</AttrValue>
</AttributeValues>
```

## Default Team Template Example using the TeamTemplateAttributeAlgorithm

The following example show the out-of-the-box XML that sets the default team template to Default for a document.

This example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
  <!-- set the team template -->
  <AttrValue id="teamTemplate.id" algorithm=
"com.ptc.core.foundation.team.server.impl.TeamTemplateAttributeAlgorithm">
    <Arg>Default</Arg>
  </AttrValue>
</AttributeValues>
```

## Document Attributes Example using the EnumTypeConstant

The following example shows the out-of-the-box XML that sets the docType and department attributes for a document. These attributes are not displayed in the user interface but must be supplied with values. PTC provides object initialization rules that supply values for these attributes. To display these attributes, customization of the user interface is required.

The wt.rule.algorithm.EnumTypeConstant algorithm requires the following arguments:

- The class name of the class that defines the enumerated list of constants
- The default constant

This example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
  <!-- set the doc type - enumerated type, this field is no longer displayed
in the UI so just default it -->
  <AttrValue id="docType" algorithm="wt.rule.algorithm.EnumTypeConstant">
    <Arg>wt.doc.DocumentType</Arg>
    <Arg>$$Document</Arg>
  </AttrValue>

  <!-- set the department type - enumerated type, this field is no longer displayed
in the UI so just default it -->
  <AttrValue id="department" algorithm="wt.rule.algorithm.EnumTypeConstant">
    <Arg>wt.doc.DepartmentList</Arg>
    <Arg>ENG</Arg>
  </AttrValue>
</AttributeValues>
```

## Algorithms Used with AttrConstraint

You specify one or more algorithms for each attribute specified in the AttrConstraint tag. Out of the box, Windchill provides the following constraint algorithms (names are shown on multiple lines in the table; enter the tag, including the name of the algorithm, on one line):

Algorithms for Use with AttrConstraint Tag	Description
com.ptc.core.rule.server.impl. GetHiddenConstraint	The user interface does not display a label or value for the attribute.
com.ptc.core.rule.server.impl. GetDiscreteSetConstraint	The user interface allows picking from a list of discrete values for the attribute. The values are specified as arguments to the algorithm.
com.ptc.core.rule.server.impl. GetImmutableConstraint	The user cannot change the attribute value that is being displayed. If an attribute that has another constraint does not have this constraint, then the user is allowed to manually enter a value.
com.ptc.core.rule.server.impl. GetServerAssignedConstraint	The user interface does not display a value for the attribute; the value is generated when the user clicks OK to save the object instance. The text displayed in place of the value is similar to (Generated).
com.ptc.core.rule.server.impl. GetServerPreGeneratedValue	<p>The user interface displays a value for the attribute. The value is generated before the user interface is displayed. The value displayed is specified as an argument to this algorithm, or if not specified, is obtained by executing the corresponding AttrValue tag for the given attribute.</p> <p><b>Note:</b> For number attributes where the default number value is sequentially generated, using this constraint can cause numbers in the sequence to be unused. The skipping of numbers occurs when a user accesses the interface and then cancels the operation before saving. There is no way to reclaim the numbers skipped.</p>

Using the com.ptc.core.rule.server.impl.GatherAttributeConstraints algorithm, you set up the structure to gather the constraints. You can specify zero, one, or more of the constraint algorithms to set the display characteristics that you want. For details on how the constraint algorithms work together and for the specific syntax to use in the AttrConstraint tag, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

The following sections provide examples that use some of the algorithms.

## Folder Constraints Example using GetServerPreGeneratedValue

The following example XML shows the out-of-the-box XML that sets the attribute display constraint for the folder location of a document.

The example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
  <AttrConstraint id="folder.id" algorithm=
    "com.ptc.core.rule.server.impl.GatherAttributeConstraints">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerPreGeneratedValue"/>
  </AttrConstraint>
</AttributeValues>
```

The constraint is supplied for the folder.id attribute and is used to obtain the default folder path that is then presented when a user creates a document.

From an application context named Demo, the **New Document** window includes the following:

Location ☐ Autoselect Folder (/Demo)

**/Demo in Autoselect Folder (/Demo)** is generated using object initialization rule set for the wt.doc.WTDocument object type. It uses the constraint in the example XML to obtain the value and assumes that /Default is the attribute value is stored in the AttrValue tag for wt.doc.WTDocument.

## Examples using GetServerAssignedConstraint and GetImmutableConstraint

The following examples use both the GetServerAssignedConstraint and GetImmutableConstraint algorithms.

### Life Cycle Constraints Example

The following example XML shows the out-of-the-box XML that sets the attribute display constraints for the life cycle template associated with a document.

The example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
  <AttrConstraint id="lifeCycle.id" algorithm=
    "com.ptc.core.rule.server.impl.GatherAttributeConstraints">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint"/>
    <Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>
  </AttrConstraint>

  <AttrConstraint id="lifeCycle" algorithm=
    "com.ptc.core.rule.server.impl.GatherAttributeConstraints">
    <Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint"/>
    <Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>
  </AttrConstraint>
</AttributeValues>
```

The constraints are supplied for two attributes:

- lifeCycle.id (for fields from which users can search for a life cycle template)
- lifeCycle (for life cycle template fields that do not provide the search capability)

Using the constraints in the example, the **New Document** window includes the following:

Life Cycle Template (Generated)

From this interface, the life cycle template assigned is the one set by the server (in the AttrValue rule) and the user cannot change the template that is used.

### Team Template Example

The following example shows the out-of-the-box XML that sets the attribute display constraints for the team template associated with a document.

The constraints are supplied for two attributes:

- teamTemplate.id (for fields from which users can search for a team template)
- teamTemplate (for team template fields that do not provide the search capability)

Using the constraints in the example, the **New Document** window includes the following:

Team Template (Generated)

From this interface, the team template assigned is the one set by the server (in the AttrValue rule) and the user cannot change the template that is used.

The following example is formatted to fit the page. Always enter each tag on one line:

```
<AttributeValues objType="wt.doc.WTDocument">
<AttrConstraint id=teamTemplate.id" algorithm=
"com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint"/>
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>
</AttrConstraint>

<AttrConstraint id=teamTemplate" algorithm=
"com.ptc.core.rule.server.impl.GatherAttributeConstraints">
<Value algorithm="com.ptc.core.rule.server.impl.GetServerAssignedConstraint"/>
<Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>
</AttrConstraint>
</AttributeValues>
```

## Conditional Logic Algorithms

To qualify when values are set or constraints used for Windchill attributes in object initialization rules, you can integrate the use of conditional logic algorithms into both the AttrValue and AttrConstraint tag syntax.

PTC provides the following types of conditional logic algorithms:

- Test algorithms - always return either a TRUE or FALSE value.
- Branch algorithms - always branch between the supplied <Arg> elements.

**Note:** Values for any attributes used within the algorithms for the conditional logic must be set before the object initialization rule is evaluated.

The note means that the attribute value used in the conditional logic must be set before the field used for the attribute specified in the AttrConstraint tag is displayed. For example, if fields for the document template and the document number appear on the same wizard step that is used to create a document, then the value of the document template cannot be used to determine the generation scheme for the number.

Also, any attribute used within default value algorithms for conditional logic evaluation purposes must be set before the object initialization code executes.

**Note:** The use of conditional logic requires that you understand the sequence of how values are set for attributes through the user interface. For example current user interface implementation for selecting folders does not support the use of conditional logic for the folder.id attribute.

For details on the syntax required for the conditional logic algorithms, access the Object Initialization Rules help by clicking the help icon on the **Object Initialization Rules** table.

The following sections describe the conditional logic algorithms and provide some examples.

## Test Algorithms

The out-of-the-box test algorithms are as follows:

Algorithm	Description
wt.rule.algorithm.EqualsTest	<p>Given two objects, determine the equality. This is an object to object comparison.</p> <p>Return TRUE if there is a match; otherwise, return FALSE.</p>
wt.rule.algorithm.StringEqualsTest	<p>Given an attribute and value determine the equality. The algorithm forces everything to be a string using toString() before the comparison. This string to string comparison is case insensitive.</p> <p>Return TRUE if there is a match; otherwise, return FALSE.</p>
wt.rule.algorithm.StringRegexEqualsTest	<p>Given an attribute and a value, with "*" in the value, determine the equality using regular expression-related concepts. This algorithm uses the java.util.regex implementation of regular expression matching. The algorithm forces the first &lt;Arg&gt; tag value to be a string using toString() before doing the comparison.</p> <p>The return is TRUE if it finds a match and the match is the whole string argument.</p>
wt.rule.algorithm.IfNullTest	<p>Given the value of an attribute, determine if the value is a null value.</p> <p>Return TRUE if the first argument is a null value; otherwise, return FALSE.</p> <p>To return a null value, use the wt.rule.algorithm.GetNullValue algorithm.</p>
wt.rule.algorithm.IfNotNullTest	<p>Given the value of an attribute, determine if the value is not a null value.</p> <p>Return TRUE if the first argument is not a null value; otherwise, return FALSE.</p> <p>To return a null value, use the wt.rule.algorithm.GetNullValue algorithm.</p>

<b>Algorithm</b>	<b>Description</b>
com.ptc.core.rule.server.impl. IfContainerTypeEqualsProjectTest	<p>Given the context in which the object resides, determine if the context is a wt.projmgmt.admin.Project2 context.</p> <p>Return TRUE if the context is an instance of the wt.projmgmt.admin.Project2 class or subclass or subtype of the Project2 class; otherwise, return FALSE.</p> <p>The algorithm returns TRUE if the context is either a project or a program context.</p>
wt.rule.algorithm.AndTest	<p>Given two or more comparisons, returns TRUE if all comparisons return true; otherwise, return FALSE.</p>
wt.rule.algorithm.OrTest	<p>Given two or more comparisons, return TRUE if any comparison returns true; otherwise, return FALSE.</p>
wt.rule.algorithm.StringInListTest	<p>Takes a list of values and tests if the first value defined in an Attr tag is equal to any of the other items in the list that are defined in Arg tags. The algorithm forces everything to be a string using toString() before doing the comparison.</p> <p>Return TRUE if there is a match; otherwise, return FALSE.</p>

## Branch Algorithms

The out-of-the-box branch algorithms are as follows:

<b>Algorithm</b>	<b>Description</b>
wt.rule.algorithm.BooleanBranch	<p>Takes a list of three objects. The first object contained in a Value tag is expected to be an object that returns a Boolean that tells BooleanBranch algorithm which of the other two objects, each contained in an Arg tag, to return.</p>
wt.rule.algorithm.CaseBranch	<p>Is similar to BooleanBranch, except that the branch occurs when a case is TRUE. There is an Arg element for each case plus one additional Arg element that identifies the default if no cases are TRUE.</p>



## Conditional Logic Algorithm Examples

**Note:** In the examples that follow, some XML elements span multiple lines because of the page width. When coding an XML document, enter each XML element on one line.

The following XML uses both the BooleanBranch and EqualsTest algorithms to test for a specific type of part identified in the genericType attribute as "variant" and then to set a V prefix when numbering parts of this type:

```
<AttrValue id="number" algorithm="com.ptc.windchill.enterprise.revisionControlled.  
server.impl.NumberGenerator">  
  <!-- add a V prefix for variant parts -->  
  <Value algorithm="wt.rule.algorithm.BooleanBranch">  
    <Value algorithm="wt.rule.algorithm.EqualsTest">  
      <Attr id="genericType"/>  
      <Arg>variant</Arg>  
    </Value>  
    <Arg>V</Arg>  
  </Value>  
  <!-- the sequence -->  
  <Arg>{GEN:wt.enterprise.SequenceGenerator:WTPARTID_seq:10:0}</Arg>  
</AttrValue>
```

The following example uses the BooleanBranch and IfContainerTypeEqualsProjectTest algorithms to set unique display constraints for numbers in a project or program. In this example, numbers in a project or program are generated and then displayed, but can be changed by the user; numbers in other contexts are generated and cannot be changed by the user:

```
<AttrConstraint id="number" algorithm="wt.rule.algorithm.BooleanBranch">  
  <Value algorithm="com.ptc.core.rule.server.impl.  
IfContainerTypeEqualsProjectTest"/>  
  <Value algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">  
    <Value algorithm="com.ptc.core.rule.server.impl.  
GetServerAssignedConstraint"/>  
  </Value>  
  <Value algorithm="com.ptc.core.rule.server.impl.GatherAttributeConstraints">  
    <Value algorithm="com.ptc.core.rule.server.impl.  
GetServerAssignedConstraint"/>  
    <Value algorithm="com.ptc.core.rule.server.impl.GetImmutableConstraint"/>  
  </Value>  
</AttrConstraint>
```

## Specifying Object Initialization Rules in a Context Template

The recommended method for including a specific set of object initialization rules in a context template is to set the rules in an existing container using the Object Initialization Administrator. Then, either save the current context as a template or export the context to a ZIP file on your system. Be sure to select the **Object Initialization Rules** option when you save or export a context. After you have a ZIP file containing the object initialization rules (along with any other administrative items you want in the template), import the template using the ZIP file as input. For the details on how to create a context template, see the [Context Templates](#) chapter.

The XML file containing the object initialization rules in a context template includes the specification of `TypeBasedRule` elements as described in [Setting File-based or State-based Versioning for Objects](#) as well as additional elements that define what is being set through the template. For details on the XML file and elements needed in the input file used when importing a template, see the [Common SharedTeamDef Element](#) in the Creating XML Files for Templates and Shared Teams appendix.

## Customizing Rules

As part of customizing your Windchill solution, you can:

- Create additional default value, constraint, and conditional logic algorithms that can be used in object initialization rules.
- Specify additional object types or attributes in object initialization rules.

To use additional object types or attributes, you must customize the user interface code so that it checks for any corresponding object initialization rules when presenting the interface for creating an object of the specified type.

**Note:** The use of custom rules involves making changes to the user interface so that the rules are checked. Before customizing, be sure that you understand the interface and follow the customization best practices as described in the *Windchill Customizer's Guide*.

Many of the object initialization rules that are loaded provide specific functionality in the user interface. Do not modify the rules that are loaded without considering the full impact of your changes.

## Custom Rule Algorithms

To create custom algorithms, PTC provides the `RuleAlgorithm` interface upon which all out-of-the-box algorithms have been built. For details on how to use this interface to create custom algorithms, see the Javadoc associated with this interface.

## Additional Types for AttrValue

**Note:** If you want to use default values for attributes other than those supported in object initialization rules, you must add types to the RuleService.properties file and you may need to customize the client code to check for the rules you create.

The RuleService.properties file contains the list of object types that are processed by the rule service during a PRE\_STORE event. This file is located in the codebase/wt/rule/impl directory where your Windchill solution is installed.

Potentially, any object type that implements the WTContained interface could be included in the above RuleService.properties list.

For additional information, see the *Windchill Customizer's Guide*.

## Additional Type and Attribute Pairs for AttrConstraint

**Note:** If you want to use constraints on attributes other than those supported in object initialization rules, you must add the type and attribute pairs to the RuleConfigurableTypeAttribute.properties file and you may need to customize the client code to check for the rules you create. Since checking for constraints can cause problems with performance, be sure to take that into consideration when deciding to constrain additional attributes.

The RuleConfigurableTypeAttribute.properties file contains the list of object and attribute pairs for constraints used. The file is located in the codebase/com/ptc/core/rule/server/delegate/init directory where your Windchill solution is installed. Additionally, you must also make modifications to the user interface so that the rules defining the display constraints are used.

For additional information, see the *Windchill Customizer's Guide*.

## Changing Numbering Schemes

Numbering schemes control how numbers are generated for parts, documents, CAD documents, dynamic documents, and change objects. The out-of-the-box numbering scheme for parts, documents, CAD documents, and change objects are simple Oracle sequences that have been loaded into the Windchill database. Each starts at 1 and increments by 1. Out of the box, there is no numbering scheme set for dynamic documents.

Dynamic documents use the same numbering scheme as CAD documents unless you add an object initialization rule that sets the numbering scheme for the EPM Document subtype associated with dynamic documents. For additional information about the dynamic document object type, see [Managing Types and Attributes for Dynamic Documents](#). For an example of setting autonumbering for dynamic documents, see [Dynamic Document Numbering](#).



**Caution:** Ensure that the numbering scheme that you want used is in place before allowing users to create parts, documents, CAD documents, dynamic documents, and change objects. Changing the numbering scheme after objects have been created can cause problems if the new scheme attempts to reproduce a number that has already been used.

You can view and update the rules that are being used from the Object Initialization Rules table using the Object Initialization Rules Administrator. For details on assessing the Object Initialization Rules Administrator, see [Accessing the Object Initialization Rules Administrator](#).

Removing a numbering scheme rule in a specific context causes the numbering scheme rule at the parent context to be used. If you remove the numbering scheme rules in all contexts in the hierarchy, then the following items are true:

- If the context is a product or library, then manual numbering occurs.
- If the context is the site, an organization, program, or project context, then manual numbering occurs for CAD documents and dynamic documents. Sequences that generate a single-digit integer, starting with 1, are used for documents and parts.

Using the out-of-the-box rules for WTPart and WTDocument, the number of parts and documents that are created in all contexts is always autogenerated. However for parts, users in a project or program context can override the generated number by editing the **Part Number** field in the part creation wizard. You can remove the ability to override the autogenerated part number by adding the following constraint algorithm in the object initialization rule for WTPart from either the site or organization context:

```
com.ptc.core.rule.server.impl.GetImmutableConstraint
```

By including this algorithm in the Site context, the **Part Number** field is no longer editable when the part creation wizard is run from all contexts. By including this algorithm in a specific organization context, only the organization and all contexts created under the organization are affected by the rule change. If your site uses the capability to move parts between projects, between programs, or between a project or program and a product or library and wants to keep the number assigned to the part, then you should consider including this algorithm.

## CAD Document Numbering

The CAD Document object type is a subtype of EPM Document and, by default, the EPM Document object initialization rule autogenerates numbers.

Users may have special requirements for numbering their CAD documents depending on the workgroup manager client that they are using. For information on the numbering requirements for working with CAD documents on a workgroup manager client, see the *Windchill Workgroup Manager Administrator's and User's Guide* for the CAD application used by your users.

## Dynamic Document Numbering

**Note:** To create and publish dynamic documents, the numbering for dynamic documents must be autonumbered. The Dynamic Document object type is a subtype of EPM Document and, by default, the EPM Document object initialization rule autogenerates numbers.

If you turn off autonumbering for the EPM Document object type, then you must create a new object initialization rule for the Dynamic Document object type to autonumber all dynamic documents. Use the logical identifier for the object type when defining object initialization rules. For example, assume that logical identifier for the Dynamic Document object type is com.acme.DynamicDocument. Then the following object initialization rule autogenerates numbers for dynamic documents:

```
<AttributeValues objType="com.acme.DynamicDocument">
<AttrValue id="number" algorithm=
"com.ptc.windchill.enterprise.revisionControlled.server.impl.NumberGenerator">
<Arg>{GEN:wt.enterprise.SequenceGenerator:EPM_seq:10:0}</Arg>
</AttrValue>
</AttributeValues>
```

For additional information about the Dynamic Document object type, see [Managing Types and Attributes for Dynamic Documents](#).

## Additional Information about Numbering

See **Object Numbering** in the Object Initialization Rules Administrator help for additional information about numbering parts, documents, CAD documents, and change objects including the following:

- The details on numbering schemes.

**Note:** By default, the sequence is 10 characters long and the pad character used is 0. Do not change the pad character to another numeric value as doing so would change the sequence ordering of the numbers.

- How to create additional numbering schemes.
- Numbering scheme examples.

The **Numbering Schemes Administration** link takes you directly to this help and can be found on the program, project, product, library, organization, and site **Utilities** page. Additionally, you can view the help from the Object Initialization Rules Administrator.

## Changing Versioning Schemes

Versioning schemes define the labels, or identifiers, that are automatically applied as object versions are created in a Windchill solution and define the order in which the labels are applied.

For all Windchill solutions, a rules load file sets the initial versioning rules for parts, documents, and CAD documents. These rules are in effect for all parts, documents, and CAD documents created from within any context. However, there is no out-of-the-box user interface for revising parts, documents, and CAD documents that are created within a project or program context (if Windchill ProjectLink is installed).

To change an out-of-the-box versioning scheme, you can change a scheme that is defined for you or you can define your own scheme. You specify the versioning scheme in an object initialization rule for a particular object type, such as `wt.doc.WTDocument`.



**Caution:** If you make any changes to a versioning scheme, ensure that the revision labels are unique within the entire versioning scheme. Specifying the same label multiple times will lead to corrupt data. Also ensure that the versioning scheme that you want used is in place before allowing users to create documents, CAD documents, and parts. Changing the versioning scheme after objects have been created can cause problems. If the scheme is changed such a way that when revising an existing object, the current revision is not in the new versioning scheme, then an error occurs. For example, assume that an alphabetic scheme is in place when some objects are created and then the scheme is changed to an integer scheme. When the user tries to revise an object that is at version A, the operation fails because the new scheme does not have the alphabetic character A. For information on accommodating legacy revision labels, see [Accommodating Legacy Characters in an Alphabetic Scheme](#) and [Accommodating Legacy Values in Revision Labels](#).

Object initialization rules can be set in context templates or by using the Object Initialization Rule Administrator from within a specific context.

For a versioning rule, you must include the following:

- `MBA|versionInfo` as the AttrValue id
- `com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator` as the algorithm
- The version series name as the argument to the algorithm

For the details on creating object initialization rules, see [Specifying Rules in the Object Initialization Rules Administrator](#).

For details on creating templates, see [Creating Context Templates](#).

The following sections describe the Harvard versioning schemes that are available and how to set them up for use within a context.

## Harvard Series Versioning Schemes

A Harvard series is made up of one or more subseries and is typically depicted as a stream of values delimited by a period (for example, 1.1, A.B.A, 1.2.1.1, and so on).

There are four Harvard versioning schemes available for your use:

- A standard alphabetic scheme
- A standard integer scheme
- A file-based scheme
- A state-based scheme

The following sections provide details on the versioning schemes that are defined and the properties used to set up the schemes.

### Properties for All Schemes

The format of the version identifier in a Harvard series is defined through following properties that are in the wt.properties file:

Property	Description	Default
wt.series.HarvardSeries.delimiter	Specifies the character that is used to separate version identifiers.	. (period)
wt.series.HarvardSeries.depth	Specifies the number of levels supported by the series.	16

### Additional Properties for Standard and Integer Schemes

Additionally, when using either the simple alphabetic scheme or the standard integer scheme (or a combination of the two), you can specify the scheme to use at each level in the following set of properties:

```
wt.series.HarvardSeries.level.1
wt.series.HarvardSeries.level.2
wt.series.HarvardSeries.level.3
:
wt.series.HarvardSeries.level.<max level number>
```

Where <max level number> is the depth of the series specified in the wt.series.HarvardSeries.depth property.

The value you can use for each level is either of the following:

Level Value	Description
wt.series.MulticharacterSeries	Uses the <a href="#">Standard Alphabetic Scheme</a> to increment the version identifiers.
wt.series.IntegerSeries	Uses the <a href="#">Standard Integer Scheme</a> to increment the version identifiers.

For example, the following line is the default setting. It indicates that the first level uses a multicharacter series:

```
wt.series.HarvardSeries.level.1=wt.series.MulticharacterSeries
```

If you do not specify a series for each level up to the number specified in the depth entry, the levels you have specified repeat.

**Note:** The file-based and state-based schemes do not use the level properties; all levels use the series as defined in the XML file that is loaded. See [File-based Scheme](#) later in this chapter.

## Standard Alphabetic Scheme

When using the standard alphabetic scheme, version identifiers are first assigned as a single letter. After all letters from A through Z have been assigned, the following series are used:

AA–ZZ

- AA, AB, AC, ..., AZ
- BA, BB, BC, ..., BZ
- (and so on)

AAA–ZZZ

- AAB, AAC, ..., AAZ
- BAA, BAB, BAC, ..., BAZ
- (and so on)

### Using the Default Standard Alphabetic Scheme Properties

This standard alphabetic scheme is defined in wt.properties using the following default property settings:

```
wt.series.MulticharacterSeries.min=A
wt.series.MulticharacterSeries.max=Z
wt.series.MulticharacterSeries.delta=1
wt.series.MulticharacterSeries.length=3
```



When a new view version of a part is created, by default a second identifier is added using a period as a delimiter. In subsequent view versions, an additional period and identifier is added. For example:

- A new view version of version B is assigned the identifier B.A
- A new view version of version B.A is assigned the identifier B.A.A
- A new view version of version Z.Z is assigned the identifier Z.Z.A

For a new revision of a part, document, or CAD document, the identifier is incremented according to the simple alphabetic versioning scheme. For example:

- A new revision of version B is assigned the identifier C
- A new revision of version B.A is assigned the identifier B.B
- A new revision of version Z.Z is assigned the identifier Z.AA

As described in the previous section, the use of the simple alphabetic versioning scheme is defined by specifying `wt.series.MulticharacterSeries` in the `wt.series.HarvardSeries.level.<level number>` properties. For example:

```
wt.series.HarvardSeries.level.1=wt.series.MulticharacterSeries
```

The out-of-the-box Harvard series that is defined in `wt.properties` is named `wt.series.HarvardSeries` and uses the property settings described in this section. The `wt.series.HarvardSeries` series is used as the default versioning series in the object initialization rules loaded.

### Accommodating Legacy Characters in an Alphabetic Scheme

In the standard alphabetic scheme, you can add the following property to identify the characters in the scheme to skip when creating a new revision label:

```
wt.series.MulticharacterSeries.legacyList=<list>
```

where `<list>` is the list of characters to exclude. Do not use a separator in the list.

By adding this property, the characters in the list can be in existing revision labels, but are not used when a new revision label is defined. For example, assume the following property is set:

```
wt.series.MulticharacterSeries.legacyList=IX
```

Then the next revision character used for an existing object with the current revision label of A.H is A.J (and not A.I). Also, existing objects with the current revision label of A.I will use A.J as the next revision label.

**Note:** Windchill allows new revisions to use legacy revision labels when a user overrides the default label by typing in the legacy label.

## Accommodating Skipped Characters in an Alphabetic Scheme

In the standard alphabetic scheme, you can add the following property to identify the characters that should never be used in revision labels:

```
wt.series.MulticharacterSeries.skipList=<list>
```

where *<list>* is the list of characters to exclude. Do not use a separator in the list.

By adding this property, the characters in the list cannot be used in either existing or new revision labels. For example, assume the following property is set:

```
wt.series.MulticharacterSeries.skipList=IX
```

Then valid revision labels will be A-H, J-W, Y, Z, AA-AH, AJ-AW, AY, AZ, (and so on).

## Standard Integer Scheme

When using the simple integer scheme, version identifiers are assigned starting with 1, incremented by 1, and can reach a maximum of 2147483647.

### Using the Default Standard Integer Scheme Properties

This standard integer scheme is defined in `wt.properties` using the following default property settings:

```
wt.series.IntegerSeries.min=1
```

```
wt.series.IntegerSeries.delta=1
```

```
wt.series.IntegerSeries.max=2147483647
```

When a new view version of a part, document, or CAD document is created using the integer series, by default a second identifier is added using a period as a delimiter. In subsequent view versions, the second identifier is incremented according to the standard integer versioning scheme. For example:

- A new view version of version 2 is assigned the identifier 2.1
- A new view version of version 2.1 is assigned the identifier 2.1.1
- A new view version of version 235.567 is assigned the identifier 235.567.1

As described in the previous section, the use of the standard integer versioning scheme is defined by specifying `wt.series.IntegerSeries` in the `wt.series.HarvardSeries.level.<level number>` properties. For example:

```
wt.series.HarvardSeries.level.1=wt.series.IntegerSeries
```

There are no additional properties needed for using the integer series.

## Changing the Harvard Series to Use the Integer Scheme

To change the out-of-the-box Harvard series (named `wt.series.HarvardSeries`) to use the integer scheme, you need only change the `wt.series.HarvardSeries.level.<level number>` property values to `wt.series.IntegerSeries`. Or you can create an alternate Harvard series that only uses integers or uses a combination of integers and alphabetic characters. For details on how to create an alternate series, see **Object Versioning** in the Object Initialization Rules Administrator help. The **Versioning Schemes Administration** link takes you directly to this help and can be found on the program, project, product, library, organization, and site **Utilities** page. Additionally, you can view the help from the Object Initialization Rules Administrator.

## File-based Scheme

When using a file-based scheme for versioning, version identifiers are assigned from an XML file that is loaded into the database. The values in the XML file can establish the version identifiers that are used when a part, document, or CAD document is revised.

The XML file that is loaded must have the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<scheme>
  <series name = "name1">
    <value>P1</value>
    <value>P2</value>
    <value>P3</value>
  </series>
</scheme>
```

## Accommodating Legacy Values in Revision Labels

All values that have been set in a revision label for any object that can be revised should remain in your scheme.

To allow values in revision labels on existing objects, but exclude the values when creating new revision labels, add the following attribute to the value tag that defines each value you want to exclude:

```
legacy="true"
```

For example, to exclude the P2 value shown in the previous section, use the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<scheme>
  <series name = "name1">
    <value>P1</value>
    <value legacy="true">P2</value>
    <value>P3</value>
  </series>
</scheme>
```

Use the legacy attribute when you are updating your versioning scheme and want to exclude the use of one or more of the existing values in the updated scheme.

**Note:** Windchill allows new revisions to use legacy revision labels when a user overrides the default label by typing in the legacy label.

### Defining the XML File Content

You can only load one XML file. However, you can include multiple series in the one file. When there are multiple series in the file, each series name must be unique. For example, the following XML file has two series defined:

```
<?xml version="1.0" encoding="UTF-8"?>
<scheme>
  <series name = "name1">
    <value>P1</value>
    <value>P2</value>
    <value>P3</value>
  </series>
  <series name = "name2">
    <value>A_1</value>
    <value>A_3</value>
    <value>A_5</value>
  </series>
</scheme>
```

There are no out-of-the-box object initialization rules loaded that set versioning to file-based versioning, as defined in this section. You can create a file-based Harvard series that uses the file that you load and name the series in one or more versioning object initialization rules.

In the XML file, the series name can contain any alphanumeric characters other than the period (.) or the at symbol (@). The series name you specify in the XML file is the name you must also append to `wt.series.HarvardSeries` to form the name used in the argument for the versioning rule. For example, if the series name is "name1", then the argument to include is "wt.series.HarvardSeries.name1".

The values contained in the `<value>` elements can be any set of unique values with the following qualifications:

- There is no upper limit to the number of characters that can be specified in each value; however, if you use lengthy values, the complete object name can be longer than the user interface field in which it is displayed.
- A value cannot be empty; it must contain at least one non-white space character.
- A value cannot contain the series delimiter that is defined in the `wt.properties.wt.series.HarvardSeries.delimiter` property. The default series delimiter is the period (.).

- Any white space before or after the value is removed before the value is used.
- The values you specify for the revision labels must be unique within the entire series and specifying the same value multiple times is not allowed; however, you can use the values from one series in another series.

The order of the values in the file determines the order in which the values are used as the version designator when the object is revised.



**Caution:** You can modify the file-based versioning values that are established through the XML versioning file; however, all changes should be made before the versioning is used. Changing values that have already been used can cause unpredictable results when versions are being updated. If you must make a change to a versioning scheme, ensure that the values you specify for the revision labels are unique within the entire versioning scheme and that you do not remove any values from a series that are already in use; however, you can include the legacy attribute for those values that you do not want used for new revision labels as described in [Accommodating Legacy Values in Revision Labels](#). Specifying the same value multiple times is not allowed. Also, removing a series name after the series has been in use causes problems for any existing objects that use the series. For example, those objects could not be revised and the object iteration history could not be displayed.

For details on how to create this series, see [Setting Up a File-based or State-based Versioning Scheme](#).

## State-based Scheme

The state-based versioning scheme is an extension of file-based versioning. It involves the use of <seed name> elements within the file-based versioning XML file.

In the out-of-the-box XML file (StateBasedVersioning.xml), the seed names correspond to the items in the Life Cycle Administrator **Version Series** drop-down list and are used in the One Phase Development and Two Phase Development out-of-the-box life cycle templates. With this setup, the values are used for versioning when the object is revised in a particular state. If an object moves forward in the state phases (such as from Prototype to Released), then the versioning series being used changes as defined in the life cycle. If a revision or set state action moves the object back to an earlier state (such as from Released back to Prototype), then the versioning series used continues to be the series in use in the most forward state. A new version of an object always uses a higher revision value than its predecessor, even if it is moved to an earlier lifecycle state.

The XML file must have the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<scheme>
  <series name = "name1">
    <seed name = "seed_name1">
      <value>P1</value>
      <value>P2</value>
      <value>P3</value>
    </seed>
  </series>
</scheme>
```

The same general information listed under [File-based Scheme](#) holds for a state-based scheme. In addition, when using the out-of-the-box Life Cycle Administrator **Version Series** drop-down list and using the Two Phase Development and One Phase Development life cycle templates (which are the default life cycle templates defined in the Product Design template), the seed names you can include are limited to the following:

NUMERIC

MILSTD

The loaded XML file that is used in the Product Design template contains the seed names and corresponding values, as follows:

Seed Name	Values in the Seed
NUMERIC	The integers 1 through 99.
MILSTD	<p>The hyphen character (-) followed by the alphabetic characters A through Y and AA through AY, but not including I, O, Q, S, X, Z, AI, AO, AQ, AS, AX, and AZ.</p> <p>You can add BA, BB, ... BY (skipping BI, BO, BQ, BS, BX, and BZ) if you want to make these values available in the series.</p> <p><b>Note:</b> This seed conforms to the U.S. military specification (ASME Y 14.35M revision standard).</p>

The loaded XML versioning file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<scheme>
  <series name = "StateBased">
    <seed name = "NUMERIC">
      <value>1</value>
      <value>2</value>
      <value>3</value>
      :
      <value>98</value>
      <value>99</value>
    </seed>
    <seed name = "MILSTD">
      <value>-</value>
      <value>A</value>
      <value>B</value>
      <value>C</value>
      :
      <value>W</value>
      <value>Y</value>
      <value>AA</value>
      <value>AB</value>
      <value>AC</value>
      :
      <value>AW</value>
      <value>AY</value>
    </seed>
  </series>
</scheme>
```

The load file that loads the Product Design template loads the XML versioning file described earlier in this section and sets the versioning for parts, documents, and CAD documents to use the values in the file. For details on the out-of-the-box versioning in the Product Design template, see [Out-of-the-box Product and Library Context Templates](#) in the [Products and Libraries](#) chapter.



**Caution:** You can modify the state-based versioning values that are established through the XML versioning file; however, all changes should be made before the versioning is used. Changing values that have already been used can cause unpredictable results when versions are being updated. If you must make a change to a versioning scheme, ensure that the values you specify for the revision labels are unique within the entire versioning scheme and that you do not remove any values from a series that are already in use; however, you can include the legacy attribute for those values that you do not want used for new revision labels as described in [Accommodating Legacy Values in Revision Labels](#). Specifying the same value multiple times is not allowed. Also, removing a series name after the series has been in use causes problems for any existing objects that use the series. For example, those objects could not be revised and the object iteration history could not be displayed.

The out-of-the-box XML versioning file lists the NUMERIC seed first and then the MILSTD seed. When assigning seeds to life cycles, the seeds must be assigned in the order that they are defined. To assign the MILSTD seed before the NUMERIC seed, you must modify the Xml file so that the MILSTD seed is before the NUMERIC seed.

**Note:** You can create additional state-based versioning schemes by adding additional seed names and values to the out-of-the-box file loaded into the database. However, the out-of-the-box Life Cycle Administrator **Version Series** drop-down list and the One Phase Development and Two Phase Development life cycle templates are not automatically updated to use the additional seed names. To use the additional seed names, you must customize the Life Cycle Administrator and modify the life cycle template you want to use.

The details on how to create or modify a state-based series are the same as those for a file-based series. See the following section for details.

## Setting Up a File-based or State-based Versioning Scheme

To set up a file-based or state-based versioning scheme:

1. Create the XML file containing the series name and the version identifier values for each file revision.
2. Load the XML file into the database.
3. Set the versioning for an object type to the versioning you have set up.

For details on each step, see the following sections.

## Creating the XML File-based or State-based Versioning File

Use an XML editor to create the XML file. For the format of the file, see either of the following:

[File-based Scheme](#)

[State-based Scheme](#)

The file you create can have any name and be located in any accessible directory.

The out-of-the-box XML file that is loaded can be retrieved by entering the following from within a windchill shell:

```
java wt.series.LoadFileBasedSeries <XML_file> -read
```

where <XML\_file> is the absolute file path of the file in which the loaded content is placed.



## Loading the XML Versioning File

To load the file-based or state-based versioning XML file, enter the following from within a windchill shell:

```
java wt.series.LoadFileBasedSeries <XML_file> -load
```

where *<XML\_file>* is the absolute file path to the existing XML file.

**Note:** Only one file can be loaded at a time. Loading a second file overwrites the first file.

To retrieve the versioning file that is currently loaded, enter the following:

```
java wt.series.LoadFileBasedSeries <XML_file> -read
```

where *<XML\_file>* is the absolute file path of the file in which the loaded content is placed.

## Setting File-based or State-based Versioning for Objects

The default Harvard series versioning scheme is set at the site level and then used at the organization, product, and library levels unless it is overwritten. One context template that is loaded (the demo Product Design template) sets the versioning for parts, documents, and CAD documents to state-based versioning. Additionally, new context templates that are created can set file-based or state-based versioning; however, PTC recommends that you set the same versioning scheme for all organizations, products, and libraries by setting the scheme at the site level.

In a context template XML file, you can specify the versioning you want used for each object type as an element in the corresponding *<TypeBasedRule>* tag for the object type. For example, in the Product Design template XML file, the following sample *<AttrValue>* tag sets the versioning for parts to StateBased (which is the series set in the out-of-the-box versioning XML file that is loaded):

```
<TypeBasedRule>
  <EngineRule isDefault="false" enabled="true">
    <ruleName>Part</ruleName>
    <ruleSpecification><![CDATA[
      <AttributeValues objType="wt.part.WTPart">
:
:
<!-- set the version info to a generated version info -->
      <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
        <Arg>wt.series.HarvardSeries.StateBased</Arg>
      </AttrValue>

    </AttributeValues>
  ]]></ruleSpecification>
  <ruleType type="INIT"/>
</EngineRule>
  <className>wt.part.WTPart</className>
</TypeBasedRule>
```

In this example, versioning for parts is set to `wt.series.HarvardSeries.StateBased`. The versioning code checks for `wt.series.HarvardSeries.StateBased` properties and, when they are not found, it looks in the loaded versioning XML file for a series named `StateBased`. If you have named the series you want to use something other than `StateBased`, then use that name in place of `StateBased` in the argument.

Additionally, you can change versioning to file-based or state-based versioning by using the Object Initialization Rules Administrator from within the context where you want the change to take place. However, all changes should be made before a product or library is used. Changing values that have already been used can cause unpredictable results when versions are being updated.

The format of the `<attrValue>` tag that is used for setting file-based or state-based versioning in an object initialization rule is the same as the format shown previously in [Defining the Content of XML Documents used for Object Initialization Rules](#). For additional information about using the Object Initialization Rules Administrator, see [Accessing the Object Initialization Rules Administrator](#).

## Using Versioning Schemes that Conform to the U.S. Military Specification

PTC supports the use of the ASME Y 14.35M revision standard for revision labels by supplying two schemes that conform to this standard:

- The out-of-the-box state-based scheme has the MILSTD seed values that conform. For details on this scheme, see [State-based Scheme](#).
- A variation of the standard alphabetic scheme that is named `MilSpec` conforms. For details on this scheme, see [Setting up the MilSpec Multicharacter Series](#).

## Setting up the MilSpec Multicharacter Series

The `MilSpec` multicharacter series conforms to the U.S. military specification (ASME Y 14.35M revision standard). This series has the hyphen (-) as the starting character and skips the characters I, O, Q, S, X, and Z.

## Defining the MilSpec Multicharacter Series

The `MilSpec` multicharacter series is defined through the following properties:

```
wt.series.MulticharacterSeries.seriesNames=MilSpec
wt.series.MulticharacterSeries.MilSpec.min=A
wt.series.MulticharacterSeries.MilSpec.max=Y
wt.series.MulticharacterSeries.MilSpec.seed=-
wt.series.MulticharacterSeries.MilSpec.skipList=IOQSZ
wt.series.MulticharacterSeries.MilSpec.delta=1
wt.series.MulticharacterSeries.MilSpec.length=3
```

For additional details on the `wt.series.HarvardSeries.MilSpec` properties, see Versioning Scheme Properties in the Object Initialization Rules Administrator help.

If you had used a scheme that allowed some of the characters now skipped in the MilSpec multicharacter series, you can add the following property to specify those characters:

```
wt.series.MulticharacterSeries.MilSpec.legacyList=<list>
```

where *<list>* is the list of characters to exclude from new revision labels. Do not use a separator in the list.

For example, to skip both I and X when creating new revision labels but allow them in existing labels, add the following property:

```
wt.series.MulticharacterSeries.MilSpec.legacyList=IX
```

## Using the MilSpec Multicharacter Series

To use the MilSpec multicharacter series:

- Use the Harvard series that is defined through the following properties:

```
wt.series.HarvardSeries.seriesNames=MilSpec
wt.series.HarvardSeries.MilSpec.delimiter=.
wt.series.HarvardSeries.MilSpec.depth=16
wt.series.HarvardSeries.MilSpec.level.1=wt.series.MulticharacterSeries.MilSpec
```

- Define your object initialization rules so the MilSpec versioning scheme is used. For example, to use MilSpec as the versioning scheme for WTPart, set the following rule:

```
<AttributeValues objType="wt.part.WTPart">
<!-- set the version using the alternate MilSpec HarvardSeries -->
  <AttrValue id="MBA|versionInfo"
algorithm="com.ptc.core.foundation.vc.server.impl.VersionInfoGenerator">
    <arg>wt.series.HarvardSeries.MilSpec</arg>
  </AttrValue>
</AttributeValues>
```



# 12

## Access Control

This chapter provides an overview of access control and background information about access control rules and access control lists. Included in the chapter are descriptions of access control permissions, descriptions of the interfaces you can use to create and manage access control policy and ad hoc rules, and information about the issues and strategies related to managing access to your enterprise information.

<b>Topic</b>	<b>Page</b>
Overview .....	12-2
About Access Control Policy Rules .....	12-3
Creating and Managing Access Control Rules.....	12-4
Setting Permissions .....	12-6
About Access Control Lists.....	12-10
About Default Access Control Policy Rules .....	12-16
Managing Access to Enterprise Information.....	12-17
Access Control Strategies for Life-Cycle Managed Objects.....	12-29
Combining Access Control Strategies for Cabinets and Life-Cycle Managed Objects.....	12-29

# Overview

As an administrator, you must ensure that only the appropriate participants have access to objects. Decisions about access rights are expressed as access control rules. There are two types of access control rules:

- Policy rules are set in domains and the rules determine the types of interactions participants can have with objects of a specific object type and a specific life cycle state. Policy rules form an access control policy for the domain.
- Ad hoc rules are set on an instance of an object and grant access control permissions on the specific object instance. These rules determine the types of interactions participants can have with a specific object instance.

For example, you can create a policy rule that gives the Publication group permission to modify objects of type WTDocument when they are in the Under Review state of their life cycle. Additionally, an ad hoc rule could be added to a specific instance of a WTDocument, such as a Publications Plan, allowing a select group of users who are not in the Publications group permission to modify this specific document.

Access control lists (ACLs) are derived from the access control rules. There are two types of ACLs:

- *policy ACLs*, which apply to an object type.
- *ad hoc ACLs*, which apply to a specific instance of an object.

The ACL is the basic mechanism for enforcing access control decisions when a user attempts to interact with an object. ACLs are created upon demand and are cached to maximize system performance.

When users are viewing the attributes of an object where some of the attributes reference access controlled objects, such as participants, then whether the user sees the value of the attributes is determined by whether the user has Read permission for the referenced objects. Typically, when a user does not have Read permission for a referenced object, the field shows **(Secured information)** instead of the attribute value. For example, assume that a user displays information about a product. On the page displayed, one of the product attribute fields is **Created By** and the value is the name of the user who created the product. If the user displaying the product information does not have Read permission for the user who created the product, then the name of the user will not appear. Instead of the name, the user sees **(Secured information)**.

This chapter discusses Windchill access control concepts, explains the relationship between domain and instance-based access control rules, and presents strategies for developing useful rules.

## About Access Control Policy Rules

One common administration task is specifying policy rules for controlling access to objects governed by a domain. When you create these rules, you customize the domain's access control policy. Subsequently, access control lists (ACLs) are derived from the policy for a domain and the policies of all its ancestor domains and used along with ad hoc ACLs to enforce your access decisions. For more information on ad hoc ACLs, see [Rules Governing Domain-based ACLs and Ad Hoc ACLs](#) later in this chapter.

An *access control rule* for a domain is a mapping between an object type, life cycle state, and a participant and their associated permissions. For an object type and a specific state, an access control rule specifies rights of a participant concerning access to objects of that type, in that state. For example, an access control rule might state that everyone in the Publications group has permission to read all objects of type WTDocument in the Engineering domain when they are in the Under Review state.

An *object type* specifies a category of objects that share the same attributes and functions. For example, WTDocument is an object type, and instances of that type may be found in some of the domains you have created. Since Windchill domains are hierarchical, access control rules defined for a domain are inherited by descendent domains. For example, access control rules defined for the WTDocument object type in all states within the Design domain apply to instances of the type within that domain or any descendent domains. Because Windchill types are also hierarchical, an object inherits rules defined from its ancestor types. Therefore, more than one rule may apply to a given object. For example, a rule that applies to the type AnnotationSet also applies to the type StructuredAnnotationSet. Additionally, there can be access control rules specific to StructuredAnnotationSet.

**Note:** Not all business objects are subject to access control, nor must all object types exist in a domain.

A *participant* is a principal and can be one of the following:

- An individual user
- A user-defined group
- A system group
- An organization

For details, the [Principals \(Users, Groups, and Organizations\)](#) chapter.

Most often, you define access control rules for groups or organizations. Both system groups and user-defined groups appear together on the **Groups** tab. Dealing with groups or organizations helps reduce administrative overhead by enabling you to apply rules to more than one user at a time. Sometimes, however, you need to create rules for a specific user. For example, an access control rule can explicitly deny one group member a permission that is granted to the entire group by another rule.

*Permissions* represent operations that apply to an object. Permissions are described in more detail in the following sections.

## Creating and Managing Access Control Rules

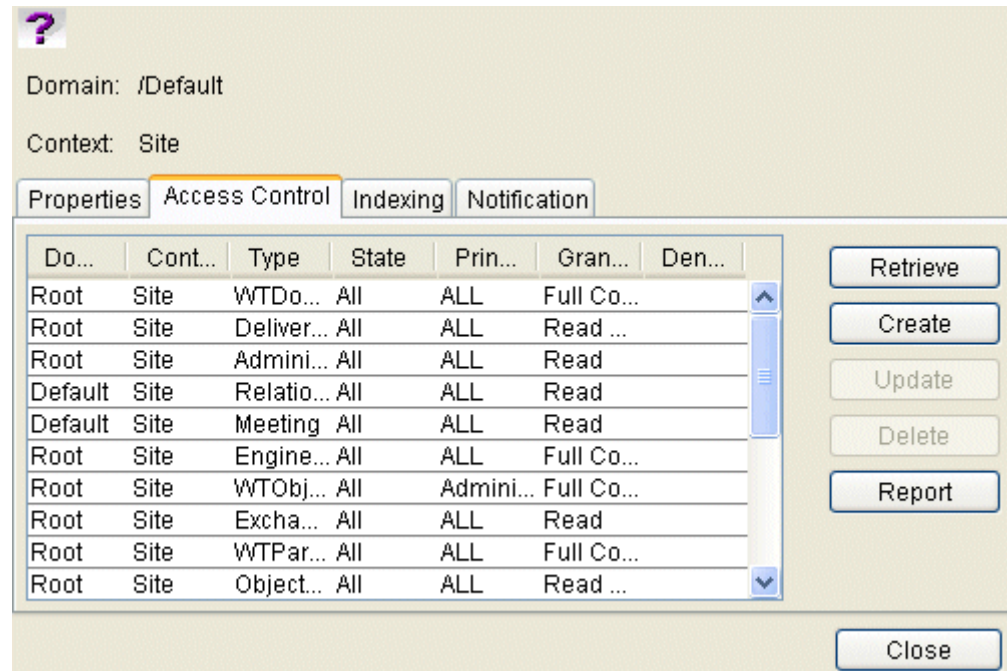
Access control rules can be created and managed in the following ways:

- Authorized administrators can create and manage policy rules that are based on a specific combination of a domain, an object type, a life cycle state, and a principal (or system group).
- Authorized users can create and manage ad hoc rules that are for a specific object instance and a principal (or system group).

As an administrator, you can create and manage access control policy rules by opening the Policy Administrator as described in [Using the Policy Administrator](#). Select a domain and click **Update**. The **Administrative Domain** window opens.



On the **Administrative Domain** window, click the **Access Control** tab to bring it to the front of the window. To create an access control rule, click **Create**. To update a rule, select a row and click **Update**. Click the help icon to display detailed instructions.



Users with sufficient access control rights can create and manage access control ad hoc rules using the **Access** table. Users access this table in the following ways:

- When creating folders, users can change the access control permissions for participants from within the set of steps presented during the creation by selecting or clearing permission check boxes in the interface.
- After an object is created, users can select the **Manage Security** action to do the following activities:
  - View the access control permissions associated with the object that are applied for the user, team members, or others.
  - Modify the access control permissions that are granted to a participant by selecting or clearing permission check boxes in the interface. If a check box is disabled, the user cannot change the permission setting.
  - View the details about the rules used to calculate access control permissions on the object for a specific participant.

- Propagate access control changes for a folder within a project or program context (including folders shared to these contexts) to the parts, documents, and CAD documents in the folder (if Windchill ProjectLink is installed).
- Recursively apply the access control changes for a folder within a project or program context to all subfolders and propagate the changes to the parts, documents and CAD documents in each folder (if Windchill ProjectLink is installed).
- When sharing an object to a project, users can change access control permissions for participants from within the set of steps presented. Change access control permissions by selecting or clearing permission check boxes in the interface.

For example, the following **Access** table shows the org1 creator access:

Participant	Permissions
<input type="checkbox"/> org1 creator	<input checked="" type="checkbox"/> Full Control (All) <input type="checkbox"/> Read <input type="checkbox"/> Download <input type="checkbox"/> Modify <input type="checkbox"/> Modify Content <input type="checkbox"/> Change Permissions

(0 objects selected)

For details on using the **Access** table, click the help icon.

## Setting Permissions

Permissions represent the rights to perform operations on an object.

The following table lists the access permissions and describes the possible rights that can be granted or denied:

Permission	Description
Full Control (All)	<p>Full control.</p> <p>A user, group, or organization with the Full Control (All) permission has all rights currently defined and any defined in the future. Therefore, if new permission types are defined, you do not have to write rules that specifically grant them to users, groups, or organizations with full control.</p>
Read	<p>The right to know the existence of an object and to view the object and its attributes. Additionally, if the object has content, you can view an object's content information such as the file path to a local file or the location of external storage. This permission does not allow you to view the actual contents of the file.</p>

Permission	Description
Download	The right to download local files that are the primary content or are attachments of an object. This right is applicable to objects with content, such as documents or drawings.
Modify	<p>The right to change the attributes of an object, as well as other characteristics that are part of the object definition but are not controlled by the Modify Content or Modify Identity permissions.</p> <p>For versioned objects, a user must have the Modify permission on the latest iteration of each version of a target object to update the attributes common to all versions that are not part of the object's identity. Modify permission on a version of a target object is required to modify that version's attributes.</p>
Modify Content	The right to add, replace, and delete content as well as to modify content information. Content can be a local file, URL, or external storage for the primary content and attachments of an object.
Modify Identity	<p>The right to modify the subset of the attributes that determine the identity of an object.</p> <p>For a part, this subset includes the part number and the organization identifier (such as cage code) of the part, but not part name (which is often treated as a short description).</p> <p>For a folder, the attributes include the folder name.</p> <p>The subset of attributes affected by the Modify Identity permission for a given object type is determined through the modeling of classes. For information on customizing the code to modify the set of attributes used in determining the identity of an object, see the <i>Windchill Customizer's Guide</i>.</p>
Create By Move	The right to move an object into an administrative domain.
Create	The right to create an object.
Set State	<p>The right of a user to perform a set state operation where a state transition has been defined to allow the transition from the current life cycle state to the new state.</p> <p>To perform a set state operation, a user must have the Set State permission and there must be a valid state transition defined between the current state and the desired state. If there is no transition defined, the user must have the Administrative permission to perform the operation.</p> <p>For information on set state transitions and operations, see online help from the Life Cycle Administrator and the <a href="#">Life Cycles</a> chapter.</p>
Revise	The right to revise an object. Revising creates a new version of the object at the same level as the original in the version tree. For example, you can create Revision B from Revision A.

Permission	Description
New View Version	The right to create a version for a specific view. For example, creating a new view version of version B assigns the B.A identifier.
Change Domain	The right to move an object out of an administrative domain.  For information about administrative domains, see <a href="#">Managing Access to Data through Access Control Rules</a> in the Administration Overview chapter.
Change Context	The right to move an object out of a context.
Change Permissions	The right to change the ad hoc permissions that others have.  Users, groups, or organizations granted the Change Permissions permission are allowed to change the ad hoc permissions of others to the permissions they themselves have or to a subset of the permissions they have.
Delete	The right to delete an object.
Administrative	Determines the right to perform certain administrative tasks. (For example, this gives you the right to break a lock or change an object's owner.)

## Setting Permissions through the Policy Administrator

Through access control policy rules, you can establish whether a specific user, group, or organization is granted or denied permissions to the objects of a specified object type in a specified life cycle state.

Selecting certain permissions on the **Access Control Rule** window automatically selects other permissions when granting access to an object type. For example, if a group is given permission to create an object, the group typically should also be able to read and modify the object; however, these automatic selections can be deselected.

The set of properties that control which additional permissions are automatically selected starts with the following:

```
wt.access.permissionImplies.
```

The following table lists the permissions that are selected automatically for each permission granted.

Permission	Selects
Full Control	None
Read	None
Download	Read
Modify	Download and Read

Permission	Selects
Modify Content	Modify, Download, and Read
Modify Identity	None
Create By Move	Read
Create	Create By Move, Modify Content, Modify, Download, and Read
Set State	None
Revise	Create By Move, Modify Content, Modify, Download, and Read
New View Version	Create By Move, Modify Content, Modify, Download, and Read
Change Domain	None
Change Context	None
Change Permissions	None
Delete	Modify Content, Modify, Download, and Read
Administrative	None

For example, if you select **Grant** for **Modify**, the **Grant** for **Read** and **Download** buttons are automatically selected, as illustrated in the following part of the **Access Control Rule** window:

Permission	Grant	Deny	None
Full Control (All)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Read	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Download	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Modify	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Modify Content	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Modify Identity	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

If you do not want to permit **Read** access, simply click the **None** button for **Read** to clear it. Selecting **None** means that the rule neither grants nor denies the permission.

## Setting Permissions through the Access Table

Through the ad hoc access rules generated from the use of the **Access** table, users can grant permissions on specific objects.

The list of permissions displayed in the **Access** table is determined by the following preferences that are under the **Security** category of the Preference Manager:

- The **Access Permission Configuration for Products and Libraries** preference establishes the permissions that are displayed and which permissions can be edited when the **Access** table is launched from within a product or library context.
- The **Access Permission Configuration** preference establishes the permissions that are displayed and which permissions can be edited when the **Access** table is launched from all contexts except a product or library context.

Access the Preference Manager to view the default settings. For information about the Access table, see [Creating and Managing Access Control Rules](#). For information about the Preference Manager, see [Administering Preferences](#) in the Contexts chapter.

## About Access Control Lists

The Windchill access control list (ACL) mechanism, and the rules for evaluating ACLs, follow those in the `java.security.acl` package. This section provides a brief description of the way in which ACLs are derived from the access control policies for domains, and describes how they work to enforce access control.

When you create an access control rule, you specify the rule antecedent and the rule consequent:

- The rule antecedent comprises three parts:
  - The domain.
  - The object type, determines which rules within an access policy apply to a specific object.
  - The life cycle state, which identifies the life cycle phase that an object must be in for the permissions to apply. If the object type is not a life cycle managed type, then the state is not applicable and only rules with life cycle state set to ALL apply.
- The rule consequent comprises three parts:
  - A participant, which is a user, a user-defined group, a system group, or an organization. A user can be a member of more than one group.
  - Associated permissions.
  - Whether permissions are granted or denied.

For access control policy rules, the participant can be the pseudo-user OWNER or the pseudo-group ALL:

- A rule defined for OWNER specifies permissions that apply to the owner of an ownable object.
- A rule defined for ALL specifies permissions that apply to all participants.

You can create access control rules for some or all of the object types within a domain. Together, these rules constitute the access control policy for the domain.

An ACL is created on demand for each rule antecedent. An ACL is derived from the policy for a domain and its ancestor domains, and is composed of multiple ACL entries. Each ACL entry contains a set of permissions associated with a participant. Each ACL entry is either positive (+) or negative (–). If the entry is positive, the permissions are granted to the associated participant. If negative, the permissions are denied.

To improve performance, when ACLs are calculated, they are cached so they can be quickly retrieved the next time a user requests access to a particular object.

## Deriving ACLs from Access Control Policies

ACLs are the mechanism used to enforce access control. This section describes how ACLs are derived from the access control policy for a domain. The next section describes how ACLs work.

An ACL is generated for each object type, state, and domain. A given object is associated with the ACL whose domain, type, and state match that of the object. For example, all WTDocument objects of a given life cycle state, within a given domain, are associated with the same ACL. In addition, this ACL is different from the ACL associated with WTPart objects that belong to the same domain.

An ACL for an object is obtained by combining all rules that apply to the object's type, state, and domain. To make this definition precise, it is necessary to describe how rules are combined, and when a rule applies to a type.

A rule is applicable to a given type when the object type referred to in the access control rule is the type itself or one of its ancestor types. For example, a rule that applies to the WTDocument type also applies to incident reports if IncidentReport is a soft type of WTDocument.

Rules that apply to a type are combined by merging rules that have the same sign (+ or –) and participant. The merge is performed by calculating the union of all permissions within the consequents.

For example, consider the combination of the following rules:

	Domain	Type	State	Participant		Permission
Rule 1:	/ (Site)	WTOBJECT	InWork	Analysts	+	Read
Rule 2:	/Parts (Bike Production)	WTOBJECT	InWork	Engineers	+	Read
Rule 3:	/Parts (Bike Production)	IncidentReport	InWork	Analysts	+	Modify

The combination of these rules produces the following ACL entries for incident reports in the InWork state in the /Parts domain in the Bike Production context:

	Type	State	Participant		Permissions
/Parts (Bike Production)	IncidentReport	InWork	Analysts	+	Read, and Modify
/Parts (Bike Production)	IncidentReport	InWork	Engineers	+	Read

## How ACLs Work

Permissions for a participant defined in one access control rule may conflict with other permissions defined in other rules. For example, you could define an access control rule that gives everyone in the Team 1 group delete permission for all incident reports belonging to the /Acme domain when they are in the Under Review state. However, in another access control rule, you could explicitly deny user Audrey.Carmen that permission, despite her membership in Team 1. In such cases, the ACL mechanism calculates the net permissions for a participant.

As defined in the java.security.acl package, the net permissions for a participant are calculated based on the following rules:

- Each participant (user, user-defined group, system group, or organization) can have at most one positive ACL entry and one negative ACL entry; that is, multiple positive or negative ACL entries are not allowed for any participant. Each entry specifies the set of permissions that are to be granted (if positive) or denied (if negative).
- If there is no ACL entry for a particular user, group, or organization, that participant has the null permission set. In effect, having a null permission set denies the participant access to the object.
- If there is a positive entry that gives a participant a permission and a negative entry that denies the participant the same permission, the permission is not granted.



- Permissions that are explicitly granted (+) to the pseudo-user OWNER override any permissions denied (–) to the user that is the owner of the ownable object through a negative entry for the individual user or for a group or organization to which the user belongs.
- Permissions that are explicitly granted (+) to the user that is the owner of an ownable object or to a group or organization to which the user belongs override any permissions denied (–) to the pseudo-user OWNER.
- Permissions that are explicitly granted (+) or denied (–) an individual user always override that user's group or organization permissions. For example, user ReneN is a member of Group 1. According to an access control rule for the Acme domain, all members of Group 1 have modify permission to incident reports in the Under Review life cycle state. However, if another access rule explicitly denies ReneN permission to modify incident reports, then, ReneN is denied the modify permission, despite membership in Group 1.
- For a given user, the net group positive permission set is the union of all the positive permissions of each group and organization to which the user belongs. This includes permissions granted for the pseudo-group ALL. For example, if user ReneN belongs to Group 1, Group 2, and Group 3, ReneN's positive group permission set includes all of the permissions granted to those groups.
- For a given user, the net group negative permission set is the union of all the negative permissions of each group and organization to which the user belongs. This includes permissions denied for the pseudo-group ALL. Similarly, the permissions denied to Group 1, Group 2, and Group 3, are also denied to user ReneN, a member.

When the permissions are calculated for the ACL, the difference between the positive and negative group permission sets for user ReneN is used to determine access rights. For example, as a member of Group 1, ReneN is granted read permission to all incident reports in the Under Review life cycle state in the /Acme domain. However, ReneN is also a member of Group 2, which is denied read access to incident reports in that domain. When calculated, ReneN's permission to read incident reports is set to null for the /Acme domain.

The following table provides some further examples of permission calculation. Assume you, the administrator, are creating an access control policy for several domains. One of the users you have identified, Ann, belongs to two groups, G1 and G2. If you assign permissions as shown in the table, Ann's resulting permissions are identified in the last column.

	<b>G1 Permissions</b>	<b>G2 Permissions</b>	<b>Union of G1 and G2</b>	<b>Individual Permissions</b>	<b>Resulting Permissions</b>
+	Modify (M)	Create (C)	(C)+(M)	Delete (D)	(C)+(M)+(D)
-	Null set	Null set	Null set	Null set	
+	(M)	(C)	(C)	(D)	(C), (D)
-	-(D)	-(M)	-(D)	Null set	
+	(M)	(D)	(M)	(C)	(C)
-	-(D)	-(C)	-(C)	-(M)	
+	(M)	(C)	(M)+(C)	(D)	(C)+(D)
-	Null set	Null set	Null set	-(M)	

When you have defined the access control rules for domains, all of the instances of the object type of a particular state, and belonging to the same domain for which you have created rules, share an ACL. This association between the ACL and the object type, state, and domain is preserved thereafter. When a participant attempts to access an object (for example, to view it or modify it), the associated ACL is retrieved, and the policy is enforced. Once an ACL is calculated, it is cached so it can be retrieved quickly for the next access request.

For example, assume that within the /Acme domain, user Audrey.Carmen is a member of a group that has read, delete permission for all objects of the type WObject that are in the Closed state. She is also a member of a group that has modify permission for all incident reports within the /Acme/Support domain in the Closed state, where IncidentReport is a soft type of WObject. However, there is another access control rule within the /Acme domain for Audrey.Carmen as an individual user, that explicitly denies her delete permission for incident reports in the Closed state.

The following shows the ACL entry for Audrey.Carmen that is associated with incident reports in the Closed state within the /Acme/Support domain:

+Audrey.Carmen read, modify

When this ACL entry was derived from the access control policies for the /, /Acme, and /Acme/Support domains, Audrey.Carmen was given read and modify permissions. Because IncidentReport is a soft type of WObject, Audrey's read and delete permissions for objects of type WObject also apply to incident reports. However, because there is another access control rule that explicitly denies her delete permission for incident reports in the Closed state, she is not able to delete objects of that type/state combination belonging to the /Acme/Support domain.

## Rules Governing Domain-based ACLs and Ad Hoc ACLs

An access control rule for the domain applies to an object type. An ad hoc ACL applies to a specific instance of that object type. The ad hoc ACL, however, specifies only positive (+) permissions; it cannot be used to deny access to an object. If the ad hoc ACL grants a permission that is denied in the policy ACL, the ad hoc rule supersedes the policy rule, and the access right is granted.

## Distributed Administration of Policy Rules

Distributed administration is the administering of a Windchill solution by different groups of individuals. Each group has responsibility for a particular area of the solution, with enough privileges to fulfill their administrative responsibilities. Domains demark administrative areas in Windchill. Windchill supports distributed administration of access control, indexing, and notification policy rules. General information about setting up administrators can be found in [Establishing Administrators](#).

Access control, indexing, and notification rules are members of the domain to which the rule applies. For example, if you define an access control rule granting Read access to documents belonging to the Publications domain, then the rule itself belongs to the Publications domain. This allows policy rules to be administered by different groups of administrators.

To give a group of administrators the rights they need to manage policy rules for an area of the system, you need to define access control rules granting permissions to the group for the AccessPolicyRule, IndexPolicyRule, and NotificationRule object types, and the domain associated with their area of responsibility. A predefined access control rule for the / (Root) domain in the site context, grants all permissions to the Administrators group for all objects, so members of this group can manage policy rules for all domains.

For example, consider the following rules:

	Domain	Type	State	Participant	Permission
Rule 1:	/ (Site)	AccessPolicyRule	All	MarketingAdministrators	+Read
Rule 2:	/ (Site)	IndexPolicyRule	All	MarketingAdministrators	+Read
Rule 3:	/ (Site)	NotificationRule	All	MarketingAdministrators	+Read
Rule 4:	Marketing (Bike Production)	AccessPolicyRule	All	MarketingAdministrators	+Full Control (All)
Rule 5:	Marketing (Bike Production)	IndexPolicyRule	All	MarketingAdministrators	+Full Control (All)
Rule 6:	Marketing (Bike Production)	NotificationRule	All	MarketingAdministrators	+Full Control (All)

These rules grant all permissions to the MarketingAdministrators group for the policy rule object types in the Marketing domain of the Bike Production library context. They allow members of the MarketingAdministrators group to view, create, update, and delete rules in the Marketing domain or any of its descendent domains, but not to manage rules in any ancestor domains. The rules granting read permissions to the MarketingAdministrators group for the policy rule object types in the Root domain allows members of the MarketingAdministrators group to see the rules inherited from ancestor domains.

## About Default Access Control Policy Rules

When a Windchill solution is installed, the set of access control policy rules that are described in the [Installed Site Context Policies](#) section of the Contexts chapter is created for the initial domains in the site context. Similarly, additional access control rules are created when an organization context or an application context is created.

For the details on the organization rules, see [Context Access Control Policies](#).

For the details on the product and library rules, see the [Out-of-the-box Context Access Control Policies](#) section in the Products and Libraries chapter.



**Caution:** The access control rules set for the domains in the site context should not be modified without considering the full consequences of the modification. For example, changing the rule that grants Administrators Full Control (All) on the WObject object type in All states should not be modified. If this rule is removed by mistake, you may not be able to administer your Windchill solution.



**Caution:** Creating rules that deny permissions for the pseudo-group ALL is also discouraged. Denying access to ALL includes denying access to users in the Administrators group unless there is a rule granting access to an individual user that is in that group.

To repair the removal of the Administrators rule described above or to remove rules such as a rule that denies access to all participants, complete the following steps:

1. Using the xconfmanager from within the windchill shell, set the wt.access.enforce property in the wt.properties file to false:

```
xconfmanager -s wt.access.enforce=false  
-t <Windchill>/codebase/wt.properties -p
```

**Note:** Setting this property to false turns off access control. This means that none of the access control rules are enforced.

2. Restart Windchill so that the new property value is used.
3. Recreate the rule that was deleted using the Policy Administrator.
4. Set the wt.access.enforce property back to true and restart Windchill.

For additional information about using the xconfmanager utility, see [About the xconfmanager Utility](#) in the Administration Overview chapter.

## Managing Access to Enterprise Information

This section describes access control issues and strategies related to managing access to your enterprise information. The following characteristics of objects impact what access control rules need to be defined to manage access to your enterprise information and to define strategies for managing the information:

- Domain administered information -- includes any object that belongs to a domain. Change Domain and Create By Move permissions apply.
- Policy and ad hoc access controlled information -- includes any object to which domain-based access control rules or ad hoc access control rules can be applied.
- Content holder information -- includes any object to which files can be attached, and for which URLs or external storage information can be specified. Download and Modify Content permissions apply.
- Foldered information -- includes any object that is contained within a folder.
- Life-cycle managed information -- includes any object that is life-cycle managed. Each life-cycle managed object has associated life cycle state. Set State permission applies.

- CAD document information -- includes the objects that are used when creating and working with CAD documents. For additional information, see the workgroup manager guide that describes how to administer CAD documents in your Windchill solution.
- Contained information -- includes any object that belongs to a context. Change Context permission applies.
- Identity access controlled information -- includes any object with identity attributes whose access is controlled separately from other attributes. Modify Identity permission applies.
- Versionable objects -- includes any object for which multiple versions can be created. Revise permission applies.
- View manageable objects -- includes any versionable object that can be assigned to a view. New View Version permission applies.

**Note:** A specific object type can have one or more of the characteristics and thus needs to have access control rules set in multiple ways. The rules and strategies you set up must take all characteristics into account.

The following sections describe each characteristic and identify the access control requirements necessary for operating on an object that has the characteristic.

## Domain Administered Information

Access control decisions for an object that is a member of a domain are based on the following criteria:

### Object Type

Determines which rules within an access policy apply to an object.

### Domain

The domain determines which access control policies apply to an object.

An object's domain and type determine which policy ACLs are associated with the object. The policy ACL, in turn, specifies which participants have permissions for objects that share the same domain and type. The set of permissions is described in [Setting Permissions](#).

## Required Rules for Domain Administered Information

The following rule governs the movement of objects among domains:

Moving an object from one domain to another requires Change Domain permission for the object in the source domain and Create By Move permission in the destination domain.

For example, to change the domain of a folder in a library, the user must have the rights to change the domain of the folder in its current domain and have Create By Move permission for the folder in the target domain.

## Required Rule for Contained Information

The following rule governs the movement of objects among contexts:

Moving an object from one context to another requires Change Context permission for the object in the source context.

## Policy and Ad Hoc Access Controlled Information

Objects that are policy or ad hoc controlled are subject to access control. The permissions that can be set on these objects are described in [Setting Permissions](#). For example, to create an object of a specific type in a domain, a user must have Create permission for the object in that domain.

## Content Holder Information

A number of Windchill 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 can be attached and for which URLs and external storage information can be specified. For example, after a document is created and saved to the Windchill database, the user who created it can add a number of files to it, which are then uploaded to the database. When the document is later checked out, the user has the option to download one or more of the content files, which can be replaced with new or updated content when the document is checked in. Users can also request that read-only copies of one or more content objects be downloaded to the local file system. That is, users can access content files without checking the content holder out of the database.

The following are several access control implications for content holders:

- Workable objects are those objects that must be checked out and checked in.
  - If a content holder is workable, content can be added only to the working copy of the object (meaning that the object is checked out). The Modify permission is required to check out and check in an object. After an object is checked out, the permissions required are based on the affected attributes.
  - If a content holder object is not workable, the permissions required are based on the affected attributes.

- There is no way to create separate access control rules for content associated with a content holder. Rules applied to the object govern access to its content as well.
- The Download permission provides the right to access the primary content and attachments of a content holder where the source is a local file.
- The Modify Content permission provides the right to add, replace or delete primary content and attachments on content holders. The source of the primary content and attachments that can be modified can be local files, URL links, and external storage.
- The Modify permission provides the right to modify object attributes directly associated with the object.
- The Modify Identity permission provides the right to modify key attributes, which are considered part of an object's identity.
- The Replace Content action for a workable object does the check out and check in as part of the action and, therefore, it requires both Modify and Modify Content permissions.

## Foldered Information

The Windchill conceptual model for information storage is based on the organization provided by an operating system. The components of this model include the following:

- *Cabinets* -- a type of folder that is the top-level organizing mechanism in the Windchill solution. A cabinet is analogous to a disk drive in the Windows operating system. The Default cabinet is exposed as the top-level folder with the context name. Other cabinets (such as the System, Network, and Type Definitions) are not exposed.
- *Subfolders* -- a type of folder that holds objects and resides in cabinets or other subfolders.
- *Folder members* (also called *foldered objects*) -- objects that must be stored in a folder.

The following sections provide more information about cabinets, subfolders, and access control rules related to foldered information.

## Cabinets

In addition to being a folder object, a cabinet is a domain administered object. When a cabinet is created, it is associated with a domain. The associated domain determines its policy rules and administrative policies.

A cabinet can contain folder members, which include subfolders and links to other folder members. Cabinets cannot contain other cabinets.



The **Display Folder Domains** preference that is located under the **Security** category determines the visibility of the domain to which a cabinet or subfolder belongs. The domain can appear on properties pages and on dialogs for creating and updating folders. If the preference value is Yes, the cabinet or subfolder's domain is displayed. For subfolders, inheritance of the domain from a parent cabinet or subfolder is also displayed when editing the folder. If the value is No, the domain information is not displayed.

A cabinet may have a primary owner. By default, the owner of a cabinet is also the owner of all information stored in that cabinet. In general, cabinets provide an organizational root for information.

To facilitate organization and control of information, the system provides the following two types of cabinets:

- *personal*: A personal cabinet is associated with a single user, who is considered its owner. In other words, you are the owner of your personal cabinet and all of the information it contains. Access to personal cabinets is through a user's work list items; there is no direct access to personal cabinets.
- *shared*: A shared cabinet is not associated with a single user and generally does not have an owner. Like a common filing cabinet, a shared cabinet contains information intended to be shared among users and groups. The information stored in a shared cabinet also generally has no owner. The administrative rules determine who has access to the shared cabinet and its objects (for those objects that inherit their domain). The shared cabinets within the system can also be thought of as a vault for storing information. However, the access control rules applied to the domain associated with the cabinet determine the level of security the cabinet provides.

## Subfolders

Similar to that of an operating system, in which a root directory contains both subdirectories and files, a subfolder is used to hold Foldered objects. Subfolders reside in cabinets or other subfolders.

## Foldered Objects

A folder member (or foldered object) must always reside in a folder, whether a cabinet or a subfolder. A folder member can be located in only one folder at a time, and its identity must be unique within that folder. However, you can use a link to make a foldered object that resides in one folder also appear to be in another folder.

By default, the owner of a folder member is the owner of the folder in which it is located.

## Domain Inheritance for Foldered Objects in Windchill

The domain association for a foldered object (other than a cabinet) is inherited from its parent, unless a domain has been explicitly associated with the foldered object.

For example, assume that the following structure exists:

- A cabinet named Default belongs to the /Default domain.
- A subfolder named Technical Documents (located in the Default cabinet), belongs to the /TechDocs domain.
- A subfolder named Installation Guides (located in the Technical Documents subfolder), inherits its domain from its parent subfolder.

In this example, both the Technical Documents and Installation Guides subfolders are governed by the policies defined in the /TechDocs domain.

The following rules govern the movement of subfolders:

- If the subfolder inherits its domain, the subfolder and all its folder members that also inherit their domain are governed by the administrative policies in the domain associated with the subfolder's new parent, when it is moved. For example, if the Installation Guides subfolder was moved to another cabinet or subfolder, this subfolder and the folder members of the subfolder that inherit their domain from the subfolder would then be associated with the domain of the new parent.
- If the subfolder belongs explicitly to a domain, it continues to belong to that domain, when it is moved. For example, if the Technical Documents subfolder is moved to another cabinet or subfolder, its domain remains unchanged and both the Technical Documents and Installation Guides subfolders and the folder members of the subfolders that inherit their domain from the subfolders are still governed by the administrative policies of the TechDocs domain.
- When you change a subfolder to inherit its domain, the domain of the subfolder, and all its folder members that inherit their domain, are changed accordingly. For example, if you change the Technical Documents subfolder to inherit its domain, then both the Technical Documents and Installation Guides subfolders are associated with the Default domain.

If you want to ensure that a subfolder retains the same domain association as its current parent, even if it is moved, you must explicitly associate the subfolder with the domain of its current folder.

To change the domain of a subfolder, the user must have the Change Domain permission for the subfolder in the source domain and to the Create By Move permission for the subfolder in the target domain.

## Required Permissions for Foldered Object Activities

The following rules govern foldered objects:

- To change the contents of a folder (for example, to create or delete foldered objects or to move objects from one folder to another), the user must have Modify rights for the folders involved in the change. To navigate to an object that is to be deleted or moved, the user must also have the Read rights to the subfolders in the path to the object.
- To move objects between folders that are not in the same domain, the rules listed under [Required Rules for Domain Administered Information](#) also apply.
- To move objects between folders in different contexts, the rules listed under [Required Rule for Contained Information](#) also apply.

If the folders are in the same domain and context, no additional permissions are needed.

## Default Access Control Rules for Foldered Objects

By default, the Administrators group has Full Control (All) rights to a cabinet and all of its folder members. This is the case because of the predefined access control rule created for the Root domain granting Full Control (All) permissions to members of this group for objects of type WtObject and all of its soft types.

Whenever you create a new Windchill user object, a personal cabinet is created for the user. Since a Windchill user name does not need to be unique and all personal cabinet names must be unique, Windchill uses the `wt.folder.personalCabinetNamingAttribute` property in the `wt.properties` file to determine what the initial personal cabinet name should be for a given user. In most cases, the name of the personal cabinet is the user's name. For additional information about naming personal cabinets, see [Naming a User's Personal Cabinet](#).

By default, the personal cabinet is associated with a child domain of the /User domain. The name of the child domain is usually the name of the user's organization unless the user is not affiliated with an organization (see [Managing Users](#) for details). The user is the owner of the personal cabinet and any objects he or she creates within this personal cabinet or any of its subfolders. Therefore, by default the user has all rights to those objects. This is the case because of a default access control rule created for the User domain that is in the site context. The rule grants Full Control (All) permissions to the OWNER of objects of type WtObject and all of its soft types. However, the domain with which the user cabinet is associated can be changed, and appropriate rules granting owner rights would need to be defined for any other domains associated with personal cabinets.

## Life-Cycle Managed Information

A very important characteristic of an object is whether it is life cycle–managed. Objects that are life cycle managed are also domain administered. Therefore, the criteria for the objects includes the domain-administered criteria.

There are two types of life cycle templates that can be associated with objects: basic and advanced. The type of life cycle used influences the access to the objects. For information about life cycles, see the [Life Cycles](#) chapter.

Access control decisions for life cycle–managed information are based on the following criteria:

### Team

Users can participate in a role for an object. The team stores the current team membership by role. The team is resolved using the team template, life cycle template and the context team. For additional information on teams, see [Teams](#).

### Life Cycle

Determines an object's initial life cycle state by associating the object with a life cycle template. Life cycle state influences both policy and ad hoc ACLs. The ad hoc ACL is computed by binding life cycle and team roles to participants. The ad hoc ACL is stored within the object.

### Life Cycle State

Indicates the phase of the life cycle, which was used to compute the ad hoc ACL. It is used during execution to determine the policy ACL that is appropriate for an object.

An object's domain, type, and life cycle state determine which policy ACL is associated with the object. The policy ACL, in turn, specifies which participants have which permissions on objects that share the same domain, type, and life cycle state.

When an object is associated with a life cycle or workflow activity, access to that specific instance of the object type can be governed by an ad hoc ACL, in addition to the policy ACL associated with the object based on its domain, type, and state. The life cycle or workflow activity can include permissions for roles associated with each life cycle phase or workflow activity. For example, participants who fulfill a life cycle or workflow role by submitting, reviewing, or promoting the object to the next life cycle phase are given access rights. Ad hoc ACLs for a life cycle phase or workflow activity are in effect for the duration of that phase or activity.

## Required Permissions for Life-Cycle Managed Object Activities

In products and libraries, the ability to set the life cycle state of a life-cycle managed object is controlled by either of following conditions:

- The user has Administrative rights to the object.
- A Set State transition is defined for the current life cycle state of the object and the user has been granted the Set State access control permission. For information on the Set State transition, see [Promotion Process](#) in the Life Cycles chapter.

Read rights to the life cycle template and the team template are required in order to select them when creating a life-cycle managed object.

## Example of Using Life Cycle Roles

When an object is created, the user is asked to select a life cycle and a team for it. Therefore, life cycle roles can be resolved by mapping them to team roles, which are then mapped to actual users.

For example, assume the following:

- For the Under Review phase of the Development life cycle, the life cycle role Promoter is mapped to the team role Team Leader.
- In the Prototype team, Team Leader is mapped to Amanda Smith.

Then, user Pat Johnson chooses the Development life cycle and selects Prototype as the team when he creates a design document in his personal cabinet. Subsequently, Pat moves his design document to a shared cabinet.

Later, when the design document is promoted to the Under Review phase in its life cycle, Amanda Smith becomes the Promoter. Although the policy ACL does not grant Amanda modify rights to design documents, she does have that access permission for Pat's document as long as the document is in Under Review phase (that is, until she submits it for promotion to its next life cycle phase).

## Example Permissions Needed for Moving a Document

The following example illustrates the permissions that are needed for the move operation as a result of the characteristics of a document. Although the permissions can be granted by either policy or ad hoc access control rules, this example describes the use of policy rules.

Moving a document from one folder to another requires the permissions described in the list below. For example, consider moving an object of type WTDocument (which is a foldered object) from one folder (which is either the SubFolder or Cabinet object type) to another.

- If the process includes navigating to the document, then it requires Read permission for the context that the document resides in because the document is contained. Using a search to locate the document does not require Read permission for the context.
- Requires Read permission for the document in the domain it belongs to (that is, the domain of the source folder) in order to select it for moving, because the document is access controlled.
- Requires Read permission for the document in the domain it belongs to after the move (that is, the domain of the destination folder) in order to view it once it has been moved, because the document is access controlled.
- Requires Modify permission on the source and destination folders because the document is foldered and the folder content is being changed (removing the document from the source folder and adding it to the destination folder).
- If the source and destination folders are in different domains, then the domain of the document will change when it is moved, since it is domain administered and inherits its domain from the folder it resides in. Changing the domain requires the Change Domain permission for the document in the domain of the source folder and Create By Move permission for the document in the domain of the destination folder.
- If the source and destination folders are in different contexts, then the context of the document will change when it is moved (since it is contained). Changing the context requires the Change Context permission for the document in the context of the source folder.

## Example Permissions Needed for Creating a Part in a Shared Cabinet

The following example illustrates the permissions that are needed to create a part in a shared cabinet or folder. The permissions can be granted by either policy or ad hoc access control rules.

Creating a part in the user's personal cabinet requires the following permissions:

- Requires Read permission for the context that the part is being created in. Because the part is contained, Read permission is required to launch the **New Part** action in the context.
- Requires Create permission for the part in the domain it is being created in (that is, the domain of the folder). Create permission is required because the part is access controlled.
- Requires Modify permission on the folder. Modify permission on the folder is required because the part is foldered and the folder content is being changed (adding the part to the folder).

In addition to these permissions, the user may also need permissions to other objects related to the part creation. For example, to select a view, a life cycle, or a team for the part, the user must have Read permission for the view, life cycle template, or team to be selected.

## Access to Cabinets

Access control rules that apply to the cabinet (based on the domain it belongs to) do not extend to the objects located within that cabinet and its folders. For example, assume that user Bill Smith has Read permission for Cabinet type in domain X. However, having Read permission to cabinets does not give Bill Smith read access to documents in a cabinet. There must be an additional access control rule defined for documents that provides the Read permission.

Also, consider the following example:

- User Bill Smith *does not* have Read permission for the Cabinet type in domain X.
- He *does* have Read permission for the Requirements type, a soft type of WTDocument, in domain X.
- Consequently, Bill Smith can search for and find Requirements documents that reside in domain X. However, because he does not have read access to the cabinet itself, he cannot see the cabinet or any of its contents through the **Folders** window.

As illustrated by this example, it is important that you create logical rules that provide users with the access they need.

## Restrictive Rules for Cabinets

One way to manage access control for Windchill objects is to define rules granting limited access to the cabinet (applied to its ancestor type WObject), and then to add more rules that grant some participants additional permissions for specific foldered object types. In general, this strategy uses the potential of shared cabinets for use as an information storage vault.

For example, you may decide that all users within the system should be allowed to see the shared Development cabinet and its contents when they are navigating in Windchill Explorer. Also assume that only the Engineering and Design groups should be allowed to check out and modify Specification (which is a soft type of WTDocument) documents stored within that cabinet. The following rules (defined for the domain to which the shared Development cabinet belongs or for an ancestor of that domain), support this strategy:

Rule 1:

WObject	+	ALL	Read
---------	---	-----	------

Rule 2:

Specification	+	Engineering, Design	Modify
---------------	---	---------------------	--------

All folder members must reside within a cabinet or a subfolder. In addition to permissions such as those above, the required rules described in [Foldered Information](#) apply.

## Open Rules for Cabinets

Another strategy for applying access control to objects is to define relatively open rules for access to the cabinet (applied to its ancestor type WObject), and then to add more rules that deny access to certain object types.

For example, you may decide to grant all participants read, modify, create, and delete permissions to objects that belong to the domain associated with the shared Development cabinet. However, you may want to deny the Publications group the right to create, modify, or delete Specification documents, and deny the Marketing group the right to create, modify, or delete Requirements documents.

The following rules (defined for the domain to which the shared Development cabinet belongs or for an ancestor of that domain), would support this strategy:

Rule 1: WObject + ALL: read, modify, create, delete

Rule 2: Specification – Publications: create, modify, delete

Rule 3: Requirements – Marketing: create, modify, delete



## Access Control Strategies for Life-Cycle Managed Objects

Consider starting with more restrictive access control rules and then using activity- or life cycle–based rules to open up access. Also consider placing rules that grant wide access to information or access to information in its final state in a policy ACL, as access can easily be extended or restricted. A policy rule can change access to many objects, while changing access control permissions in ad hoc ACLs requires action on each individual object instance.

As described earlier, you can establish complementary access control rules for domains and the objects that are associated with them. Similarly, you can implement an access control strategy by balancing the use of policy and ad hoc ACLs.

Conversely, you can create a domain policy that provides for more open access to cabinets and their contents. In this case, you would define few (if any), access control rules within the life cycle.

## Combining Access Control Strategies for Cabinets and Life-Cycle Managed Objects

In some cases, you may decide to create restrictive domain policies, which provide only the minimum access to most users. Specifically, you can grant users read permission to one or more shared cabinets, so they can view the cabinet while withholding additional permissions for objects residing in the cabinet and its folders. Then, based on the life cycle and team associations for the objects within each cabinet, you can use ad hoc ACLs to grant certain participants the access permissions they need to fulfill their roles for a life cycle phase or workflow activity.

Life cycle roles can be mapped to team template roles when a life cycle is created. For example, the life cycle role Promoter can be mapped to the team role Team Leader. When a team is defined, roles are mapped directly to specific participants or to actor roles (of which there is only the Creator actor role currently defined). In addition, the context team roles and members are used. For additional information about teams, see the [Teams](#) chapter.

Additionally, life cycles can contain access control rules for specific phases and roles. For example, assume that the Development life cycle includes an Under Review phase. The access control rules for this phase specify that for the duration of the phase, the Promoter role has modify permission for the object.



# 13

## Type and Attribute Manager

This chapter discusses the basic concepts of types and the runtime typing capability. It describes the Type and Attribute Manager utility and how to use it to define new soft types, attributes, and constraints. The chapter also has information about managing types and attributes for parts, documents, CAD documents, dynamic documents, and change objects.

Topic	Page
Overview of Types and the Runtime Typing Capability .....	13-2
Overview of the Type and Attribute Manager Utility .....	13-7
Using the Type and Attribute Manager .....	13-19
Specifying Windchill Types .....	13-35
Managing Types and Attributes for Parts, Documents, and Change Objects ..	13-36
Managing Types and Attributes for CAD Documents .....	13-39
Managing Types and Attributes for Dynamic Documents .....	13-43
Best Practices .....	13-47
Glossary Terms for the Type and Attribute Manager Utility .....	13-51

## Overview of Types and the Runtime Typing Capability

The **Type and Attribute Manager** utility is used to manage the out-of-the-box type definitions, attributes, measurement systems, and quantities of measure. This section provides an overview the ways in which you can extend the out-of-the-box types and attributes. For information specific to the Type and Attribute Manager utility, see the section, [Overview of the Type and Attribute Manager Utility](#).

Each Windchill out-of-the-box modeled class appears as a node on the **Type Manager** tab. These modeled classes define the out-of-the-box *modeled types* that are available to you.

There are two ways to extend the set of out-of-the-box types:

- By modeling additional classes through Rational Rose, you can add subclasses to Windchill business objects.

After code is recompiled, any modeled classes that extend the basic Windchill business objects automatically appear as subtypes for the classes that were extended on the **Type Manager** tab. The Type and Attribute Manager utility supports inheritance; therefore the newly created subtype has both the modeled and non-modeled attributes of the parent type. For example, assume that your company creates a subclass in Rational Rose named wt.doc.SiteDocument which extends the Windchill base class wt.doc.WTDocument. Then wt.doc.SiteDocument appears as a subtype of wt.doc.WTDocument in the Type Manager and automatically has the attributes from wt.doc.WTDocument. From the Type Manager, you then can update the type to change the display name (for example, use Site Document), add an associated icon, add additional attributes, and further refine constraints, or you can create new subtypes of wt.doc.SiteDocument with its corresponding attributes and constraints.

For additional information on why you may want to use modeled classes for subtypes, see [When to Use Typing and When to Use Modeling](#) in this chapter.

- By using the runtime typing capability available through the **Type and Attribute Manager** utility, you can augment Windchill out-of-the-box business object types without changing the object model and writing code.

Runtime typing allows you to fine-tune the system and address changing needs without recompiling, rebooting, or stopping operations. Using the **Type Manager** tab to create a type that does not have an underlying modeled class creates a *soft type* as a subtype of the parent type. For example, you can create a soft type with the **Site Document** display name that is a subtype of wt.doc.WTDocument (instead of customizing code as described earlier). The **New Type** window allows you to enter the initial information such as name, description and display name. The new soft type subtype inherits both the modeled and non-modeled attributes of the parent type. Additionally, you can update the soft type to add an associated icon, add additional attributes, and further refine constraints, or you can create new subtypes under the soft type.

You can use the Type and Attribute Manager utility to do the following:

- Augment a Windchill part or document by adding additional [soft attributes](#), or adding [soft types](#) with different attribute sets (implementation).
- Quickly show your end users how Windchill could be used to solve their business problem (prototyping environment).
- Distribute Windchill to multiple divisions, when each division wants to slightly modify the site-specific modeled classes to enhance the part and document definitions for their own division (deployment.)

**Note:** You can provide display values for types, attributes, and valid value lists in multiple locales. For details, see the [Localizing New Type Definitions](#) section.

## Effects of Deploying a New Type

When you deploy a new type definition, the type is recognized and used by the following Windchill functionalities:

- Access control policies
- Indexing policies
- Notification policies
- Life cycle template definitions
- Base loader
- Reporting
- Windchill adapter

You cannot define external file vault policies or replication policies specifically for a new type. However, the type inherits the rules that apply to the modeled class on which the type definition is based.

## Using Typing in Conjunction with Classification

The Windchill classification capabilities complement the Windchill typing capabilities. Typing allows you to define the business process characteristics of an object that can be used for processing the object through its product life cycle. Classification allows you to define the attributes that describe the object's form, fit, and function; it is used to classify and organize a product master database to promote consolidation of suppliers and reuse of design components.

## Overview of Classification

Windchill's classification capabilities enable Windchill to become a searchable repository for part and supplier data. This allows your organization to consolidate suppliers, standardize parts, manage multiple part number schemes, and promote the reuse of design and component knowledge across the manufacturing organization.

Sourcing administrators and design engineers can search the Windchill repository by navigating a textual hierarchy or image matrix, or by initiating parametric searches against attributes that describe the form, fit, and function of the business object. Advanced searching capability is available to help find second sources and functional equivalents for a selected part.

If classification is implemented at your site, you can develop navigation structures to help end users in their searches. A navigation structure is a hierarchical set of navigation nodes, each with a textual representation and a graphical representation. Navigation structures present product data in a way that will help different users uniquely. For example, a design engineer may want to view a detailed mechanical hierarchy of parts with a large set of attributes, while a dealer is interested only in locating the springs used on a particular tractor model. While the user navigates a Windows-like folder structure, in the background, the system "navigates" an object type, such as parts or suppliers. Queries from this navigation structure are executed against objects of that specific type.

Classification structures are a special case of navigation hierarchy, where a child in the hierarchy is always a type of its parent. Windchill classification structures provide templates for classified Windchill objects, for example, parts and suppliers. Each node of a classification structure has an associated set of attributes that describe the part's form, fit, and function, as well as attribute value constraints and a representative graphical image. Classification node attributes and constraints are used as a template for assigning form, fit, and function attributes and constraints to newly classified parts and suppliers. The template is not strictly enforced, and the end user may optionally change the template definition depending on the form, fit, and function data available for any instance of the object being classified. A single instance of a part or supplier can be classified multiple times to reflect different views of the same business object.

## Comparison of Classification and Typing

Windchill's typing capability allows you to extend the Windchill data model without using the modeling and system generation capabilities of Windchill's Information Modeler. The typing capability allows you to add attributes and further refine constraints for existing Windchill business objects. Typing also allows you to create subtypes of existing Windchill business objects and add attributes and constraints to those subtypes.

The attributes assigned through either modeling or typing are directly related to the product and process information necessary to manage the business object over its complete product life cycle. Product and process information is used to support business processes such as: the development of a Bill of Materials (BOM), support of where-used queries, and the development of a complete and accurate change history. Many of the attributes assigned to the definition of an object are inherited from fundamental Windchill base classes to support the management of the product through its life cycle. Other attributes are inherited through your organization's addition of new attributes using either modeling or the Type Manager.

Classification allows the end user to omit defined attributes from a particular instance of a business object; however, modeling and typing strictly enforce their definitions of the attributes expected for each instance. Business objects can have multiple classifications but only one type.

## **When to Use Typing and When to Use Modeling**

The typing capability complements Windchill's model-driven customization approach of modeling with Rational Rose, system generation, and Java coding. The modeling and Java coding-based approach should be used to provide fundamental new types of information in the system. For example, a plant object would be a new type of information, where new functional capabilities are required and new behavior must be added, through new methods or the implementation of plug-and-play interfaces.

You should use modeling in the following situations:

- You need to add behavior to an extension of a Windchill business class with new methods or plug-and-play interfaces. The modeled class could subsequently be augmented with additional attributes, added through typing, to reflect changing business needs at some time in the future.
- You want to develop granular policy rules for administrative policies that recognize typing. For example, if you want to develop a replication rule that would replicate only a particular type of part, you would need to create the part type by extending the part through modeling. This is because replication policies are based on classes.
- You want to make your tailored business objects visible to the end user and the rest of the Windchill system; however, the customization effort is less than the effort required to make the object visible through typing.

## Localizing New Type Definitions

Types and their attributes can be localized in the same files and in the same manner as modeled classes and attributes; however, entries in these files are not generated automatically.

Soft types can be localized just like modeled types. For example, if one wanted to localize a soft-subtype of WTPart, the partModelRB.rbInfo would be appended with entries like:

```
# Entry Format (values equal to default value are not included)
# <key>.value=
# <key>.category=
# <key>.comment=
# <key>.argComment<n>=
# <key>.constant=
# <key>.customizable=
# <key>.deprecated=
# <key>.abbreviatedDisplay=
# <key>.fullDisplay=
# <key>.shortDescription=
# <key>.longDescription=
```

where *<key>* is the external form for the soft type "WCTYPE|wt.part.WTPart|com.myco.MySoftPart"

**Note:** Do not use multibyte characters in the soft type name. Multibyte characters are not supported.

Attributes can be handled in a similar way, but the corresponding file is com/ptc/core/meta/common/DefinitionResource.rbInfo:

```
# Entry Format (values equal to default value are not included)
# <key>.value=
# <key>.display=
# <key>.abbreviatedDisplay=
# <key>.fullDisplay=
# <key>.shortDescription=
# <key>.longDescription=
# <key>.dataType=
```

where *<key>* is the external form for the soft attribute "IBA|mySoftAttribute"



## Migrating Existing Type Instances to a New Type Definition

When a type definition is changed to add additional soft attributes, existing type instances are updated, using a strategy called *lazy migration*. This means the affected instances do not need to be changed immediately (that is, as soon as the new definition is deployed), but only when an individual instance is updated.

For example, if you add an attribute to a type, it will subsequently appear on forms to create new instances of the type and will be added to a new object instance if the user assigns it a value. It will not be automatically added to existing instances of the type; an end user must update the instance and give the attribute a value to add it to the instance.

## Overview of the Type and Attribute Manager Utility

The **Type and Attribute Manager** utility allows you to manage type definitions, attribute definitions, measurement systems, and quantities of measure. Users with the appropriate permissions can create, modify, or remove types or attributes. Additionally, users can configure display units for attributes, associate attribute definitions with type definitions, and constrain the values of those attributes.

The **Type Manager** and **Attribute Definition Manager** tabs are generally used together to create and edit types and attributes. The **Measurement System Manager** and **Quantity of Measure Manager** tabs provide you with the flexibility to define units and display units according to your business needs. For example, when assigning an attribute for length to a type, the standard (default) unit for length in the U.S. Customary System (USCS) is inches (in) which may not align with your business processes. Therefore, you can define your system by creating a new measurement system or overriding the defaults to better suit your business needs.

The overall procedure for creating and assigning attributes to types may include creating new or editing existing measurement systems and quantities of measure based on your business processes.

The following list is intended as a summary of how to use the Type and Attribute Manager. It provides a high level list of what to consider and the order in which each section should be defined for best results. The links within each step below will navigate you to the appropriate section for more information.

Typically, the measurement systems and quantities of measure provided out-of-the-box in the Type and Attribute Manager will be sufficient for your business needs. The following steps are provided for your consideration in the situation that you need to define new measurements.

1. Determine if you need to create a new measurement system by reading through the [Measurement System Manager Tab](#) section. The online help provides details for creating and modifying measurement systems.
2. Determine if you need to create a new quantity of measure by reading through the [Quantity of Measure Manager Tab](#) section. The online help provides details for creating and modifying quantities of measure.
3. Create new attributes or edit existing attributes using the [Attribute Definition Manager Tab](#).
4. Create new types and subtypes using the [Type Manager Tab](#).

**Note:** If applicable, override the Display Units for the attributes. The Type and Attribute Manager online help has more detailed information about the display units and supported units used when creating measurement systems and quantities of measure.

5. Add attributes to the types and subtypes using the **Template** tab on the **Type Manager** tab. For an example, see the [Adding an Attribute to a Type](#) section.
6. Set constraints for the attributes using the **Template** tab on the **Type Manager** tab. For an example, see the [Setting Constraints on Attributes](#) section.

## Accessing the Type and Attribute Manager Utility

From Windchill, access the Type and Attribute Manager utility from either the **Site** or the **Organization** tab. Click the **Utilities** sub-navigation link, then select the **Type and Attribute Manager** link. You can access the individual utilities by selecting the following tabs:

Type and Attribute Manager Utility Tab	Description
<a href="#">Type Manager Tab</a>	Allows you to create new subtypes of those parts, documents, change objects and other objects installed with your solution. From this tab, attributes are associated to types.
<a href="#">Attribute Definition Manager Tab</a>	Allows you to create new attributes that can be used by a variety of Windchill objects, and associated with different values.
<a href="#">Measurement System Manager Tab</a>	Allows you to define your own measurement systems and override the default display units for the four standard measurement systems. In addition, you can create new measurement systems.
<a href="#">Quantity of Measure Manager Tab</a>	Allows you to define your own quantities of measure and override existing display units. Windchill supports over 40 standard quantities of measure. New quantities of measure can also be created.

You can also launch the Type Manager from the **Types** sub-navigation link found on the **Site** and **Organization** tabs. From here, you can create, edit, and delete document types. However, you cannot create a document type from the **Types** page of the **Organization** tab when the organization context has the same name as the site.

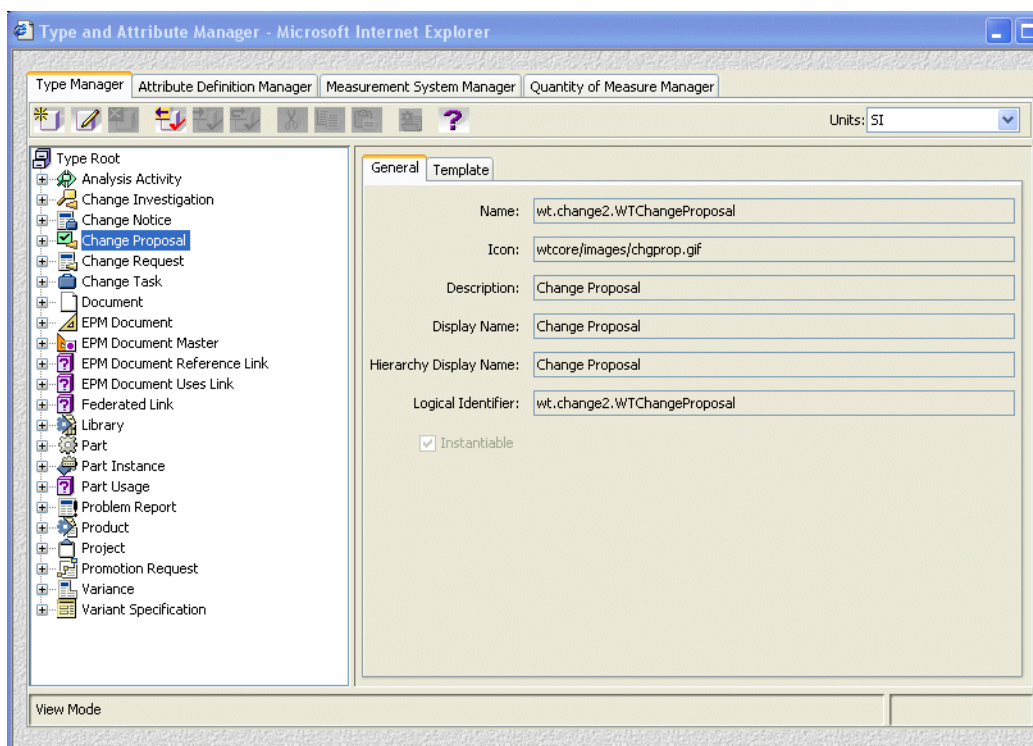
## Type Manager Tab

When the Type Manager is accessed from the **Site** tab, it provides the system administrator with unrestricted access to all types. The link from the **Organization** tab provides access to only those types that are in the active organization context at the time the Type Manager was launched.

The Type Manager user interface allows you to perform the following actions:

- Edit the existing Windchill out-of-the box business object types.
- Create new [soft types](#) of the existing Windchill out-of-the-box part, document, and change object types, and any modeled extensions to these types created at your site.
- Create new [soft attributes](#) for any existing type.
- Define constraints for any existing attribute.
- Edit types, attributes, and constraints.
- Delete types and attributes.
- Duplicate types (by copying and pasting) and move types (by cutting and pasting).
- Add a logical identifier for an existing type. For information on logical identifiers, see [Specifying Windchill Types](#).

The following graphic provides an example of the **Type Manager** tab:



When you launch the **Type Manager** tab in the context of an organization and view a type in the **General** tab, the internet style domain of the organization is prefixed in reverse order on the type name. For example, if the **Type Manager** tab is launched in the context of an organization representing PTC, then the name will be prefixed with "com.ptc". For more information about associating types in organization contexts, see the [Associating Soft Types with the Site and Organization Contexts](#) section.

The left pane displays the type hierarchy in an expandable and collapsible tree structure. The right pane is used for viewing, creating, and editing types, attributes, their values, and their constraints.

The **Units** drop-down list in the upper right-hand corner of the **Type Manager** tab allows you to toggle between measurement systems, providing immediate conversion from one measurement system to the next. If one measurement system's default is in inches and another system's default is meters, when you toggle from one to the next the system automatically converts the measurement.

**Note:** In order to create or modify types, a user must either be a site administrator or an organization administrator that is a member of Type Administrators group. For more information, see the [Administrator Roles](#) section.

The out-of-the-box types that appear in the Type Manager are referred to as [hard types](#). When you use the **Type Manager** tab to create a type that does not have an underlying modeled class, a subtype of the parent type is created.



**Caution:** See [Best Practices](#) for information about restrictions on creating soft types and soft attributes in Windchill.

The **Type Manager** tab allows you to create subtypes of those parts, documents, change objects, and other objects installed with your solution. When subtypes are created using the **Type Manager tab**, they are called soft types. Subtypes can be created for hard types, modeled types and soft types. In addition, you can add new attributes to and set constraints on the newly created soft types. Some soft types are defined by PTC during Windchill development and are included in your Windchill solution installation.

The Type Manager supports inheritance; therefore the newly created subtype has both the modeled attributes and non-modeled attributes of the parent type.

**Note:** When naming a type, it should be unique among siblings and can have a maximum of 200 alphanumeric characters and underscores. Hyphens are not allowed in type names and underscores are not allowed within the internet domain.

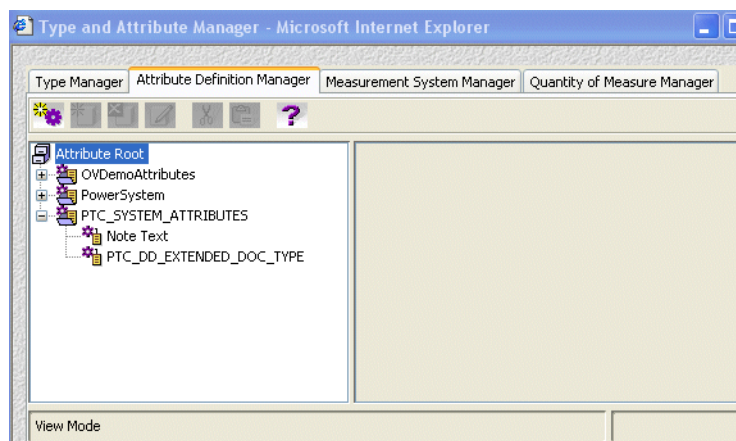
## Attribute Definition Manager Tab

Out-of-the-box site administrators can access the Attribute Definition Manager to create and edit attributes. Organization administrators need special permissions set to create and modify attributes; they must be added to the Attribute Administrators group. For more information, see the [Administrator Roles](#) section.

You can define attributes for the following object types:

- Documents and document subtypes
- Products
- Parts
- Parts usage links
- Serial numbered parts
- Problem reports
- Change activities
- Change requests
- Change notices
- CAD documents
- Dynamic documents

The following graphic provides an example of the **Attribute Definition Manager** tab:



In Windchill, you can define [soft attributes](#) in order to augment the attributes of out-of-the-box business objects. Soft attributes are non-modeled; they are created at runtime and do not require recompiling or interrupting operations.



**Caution:** See [Best Practices](#) for information about restrictions on creating soft types and soft attributes in Windchill.

The set of all soft attributes (that is, non-modeled) is called the [soft attribute set](#). An attribute in the soft attribute set can be used by different Windchill objects and classes and associated with different values. For example, the attribute material may have values such as steel, aluminum, and plastic when it is associated with a mechanical part, but when it is associated with an item of clothing, it may take on values of cotton, rayon, or polyester.

The soft attribute set is maintained as a list of attributes that are independent of any real objects or object classes. Therefore, attributes created from the **Attribute Definition Manager** tab do not have associated values.

An attribute represents a characteristic of an object. Examples of attributes for an object type of document, are document number, page count, length, and fax number. Each attribute has both a hierarchy display name and a display name. The hierarchy display name is used only when the attribute is viewed in the tree representation of the soft attribute set, and the display name is used when the attribute is displayed in the context of a Windchill object.

Attributes that are added to a type can be optional or required and constraints can be set on those attributes. Constraints specify characteristics, such as range restrictions or a valid value set. You can also adjust or provide constraints on inherited attributes.

After an attribute has been created, you can update it to specify a description and a display name. Each attribute also has a logical identifier used to identify the type in external configuration and properties files. A logical identifier must be defined for the soft attribute to make it searchable. If you are customizing, the logical identifier can be overridden in <Windchill>/codebase/LogicalAttributes.xml. For more information about customizing, see the *Windchill Customizer's Guide*. Additionally, each attribute has a data type, such as boolean, string, or real number.

**Note:** When naming a new attribute, the name should not contain the characters "\" or "/". It should be unique among siblings and can have a maximum of 200 alphanumeric characters and underscores. Hyphens and spaces are not allowed in attribute names. The suggested internet domain should not contain underscores or additional periods. Also, the name should not contain any lower case characters if it needs to be communicated with a Pro/ENGINEER designated parameter.

You can add an attribute to any of the supported types listed earlier in this section. Start the utility as described in the [Accessing the Type and Attribute Manager Utility](#) section and then select the **Type Manager tab** which displays the object types. For more information about adding attributes to types, see the section, [Using the Type and Attribute Manager](#) for an example of creating soft types, associating attributes to the types, and setting constraints. The online help provides detailed procedures for using the Type and Attribute utility.

## Data Types Supported in the Attribute Definition Manager Tab

The following types are supported:

- Boolean
- Date/Time stamp
- Real Number
- Real Number with Units
- Integer
- Reference
- String
- URL



## Measurement System Manager Tab

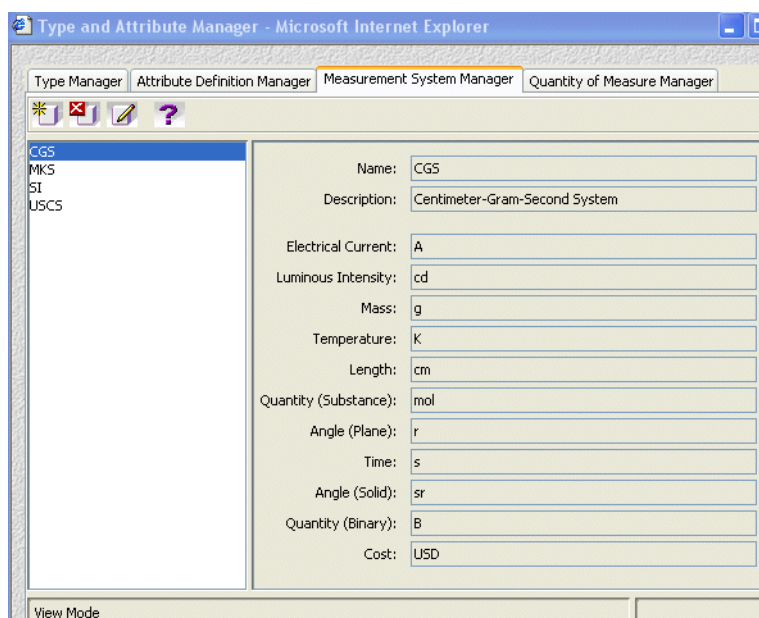
Out-of-the-box site administrators can access the **Measurement System Manager** tab to create and modify measurement systems. When a user is designated as an organization administrator, the user must also be added to the Attribute Administrators group to have permission for creating and modifying measurement systems. See the [Additional Administrative Groups](#) section in the Administration Overview chapter.

A measurement system is a set of units associated with the standard base measures, such as meters, kilograms, and seconds. In addition to user-defined measurement systems, Windchill supports the following standard measurement systems:

- International System of Units (SI)
- Centimeters-Grams-Seconds (CGS)
- U.S. Customary (USCS)
- Meters-Kilograms-Seconds (MKS)

In Windchill applications, you can define your systems to meet your needs -- this is useful if you are working with parts on a microscopic scale or a scale larger than normally used in the standard measurement systems listed above. In scenarios like this, a new measurement system can be created to define the display units of a very specific scale. The application's automatic conversion among measurement systems is useful when you are working with parts from various suppliers. For example, parts are often specified in U.S. Customary units (such as inches, pounds, and seconds) if they are made in the United States, but a metric system, such as CGS, is more widely used in other countries. While looking at a specific part, you can easily switch between the measurement systems for automatic conversion from one unit of measure to the next (for example, inches to nanometers).

The following graphic provides an example of the **Measurement System Manager** tab:



From the **Measurement System Manager** tab you can define your measurement systems to meet your specific business needs.

For detailed procedures on creating and modifying measurements systems, see the Type and Attribute Manager online help.

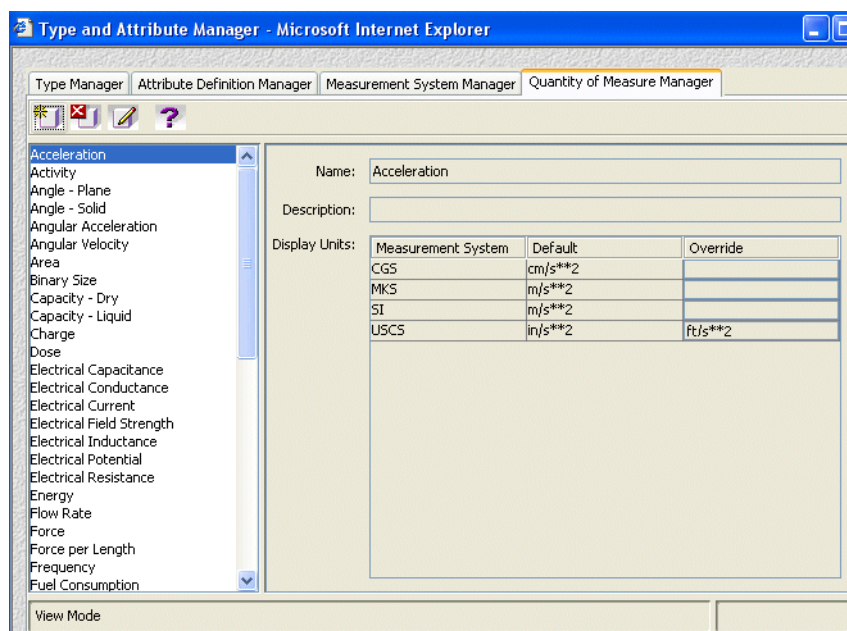
## Quantity of Measure Manager Tab

Out-of-the-box site administrators can access the **Quantity of Measure Manager** tab to create and modify quantities of measure. When a user is designated as an organization administrator, the user must also be added to the Attribute Administrators group to have permission for creating and modifying quantities of measure. See the [Additional Administrative Groups](#) section in the Administration Overview chapter.

Windchill supports over 40 standard quantities of measure in attribute definitions. Each quantity of measure has a syntactically valid set of units. Some quantities of measure, such as energy and torque, may be expressed in the same units but have different connotations to the user. The Type and Attribute online help provides a list of the supported quantities of measure.

A quantity of measure is defined according to its measurement system. For example, acceleration is a quantity of measure that is a function of the base measures length and time. So, in the International System of Units (SI) measurement system, acceleration is expressed as  $\text{m/s}^2$  (meters divided by seconds squared), while in the U.S. Customary (USCS) measurement system, it is expressed as  $\text{in/s}^2$  (inches divided by seconds squared).

The following graphic provides an example of the **Quantity of Measure Manager** tab:



From the **Quantity of Measure Manager** tab, you can define your own quantities of measure and override existing display units. For example, in the MKS measurement system, force is expressed as  $\text{kg}\cdot\text{m}/\text{s}^2$ , but you can override this setting and use Newtons (N) or Pound\_Force (lbf) instead. To override the base measures for a specific quantity of measure, use the [Best Practices](#) section to derive system-compatible units.

For detailed procedures on creating and modifying quantities of measure, see the Type and Attribute Manager online help.

## Supported Units

When creating measurement systems and quantities of measure, there are required supported unit strings that must be syntactically correct.

*Base units*, *pure units*, and *unit prefixes* are used in various combinations to build the supported unit strings. The Type and Attribute Manager online help contains a section on Windchill Units that provides examples of supported unit strings and how to build them. A list of the base units which are the basis for all convertible units can be found in this section.

## Administrator Roles

This section is provided as a quick reference for administrative permissions that apply to creating and managing type definitions, attributes, measurement systems, and quantities of measure. For more additional information about administrator roles and permissions, see the [Additional Administrative Groups](#) section in the Administration Overview chapter.

Site administrators can:

- create and modify types and attributes in both the site and organization contexts
- can create and modify units (quantities of measure and measurement systems) in both the site and organization contexts

Organization administrators can:

- view types and attributes that reside in the site context
- create and modify types in the organization context
- create and modify attributes and units in the organization context, when they are part of the Attribute Administrators group (see the Note for Attribute Administrators below)

Type administrators can:

- create and modify types

**Note:** When a user is designated as an organization administrator, they are automatically added to the Type Administrator group.

Attribute administrators can:

- create and modify attributes
- create and modify units (quantities of measure and measurement systems)

**Note:** When a user is designated as an organization administrator, the user must also be added to the Attribute Administrators group to have permission for creating and modifying attributes and units. Organization administrators are not automatically added to the Attribute Administrator group, see the [Additional Administrative Groups](#) section in the Administration Overview chapter.

## Using the Type and Attribute Manager

This section provides an example for using the Type and Attribute Manager.

The following procedures are intended to be used together, and consist of an overall example that is a common use of the **Type Manager** tab and the **Attribute Definition Manager** tab:

- [Creating a Soft Type](#)
- [Creating a Soft Attribute](#)
- [Adding an Attribute to a Type](#)
- [Setting Constraints on Attributes](#)
- [Editing a Type Definition](#)
- [Deleting a Type](#)

The examples that follow are based on a multinational company that has deployed Windchill to one of its divisions. Windchill contains a new *soft type* named "Part Type", which is extended from the *hard type*, "Part". This division wants to create an additional soft type, based on "Part Type", which will be named "Unit 1 Part".

The new soft type will contain the following new *soft attributes*:

- *cageCode*, an integer value identifying the vendor.
- *procurementLeadtime*, using a range constraint of days.
- *system*, a string type identifying whether the part is in the electrical, hydraulic, or pneumatic system.
- *spare*, a boolean value (this attribute will be used in the example of [Editing a Type Definition](#))

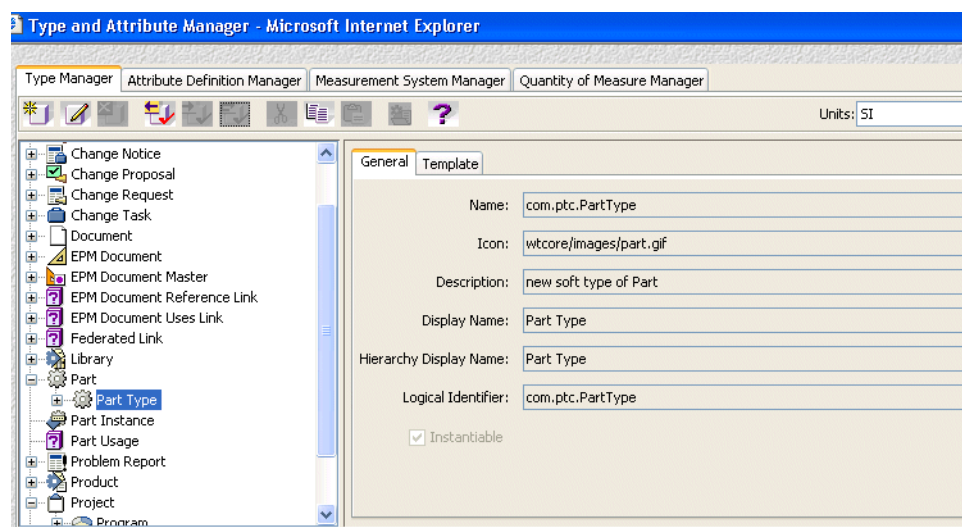
### Creating a Soft Type

Use the following procedure to create a soft type as part of the example described above.

1. Start the **Type and Attribute Manager** from the **Site Utilities** page, as described in the [Accessing the Type and Attribute Manager Utility](#) section.
2. Select the **Type Manager** tab to ensure it is the active tab.

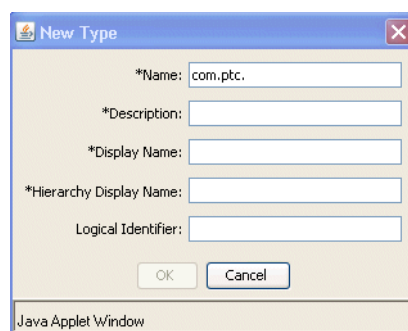
- Expand the **Part** type and then select the soft type named **Part Type**.

The **General** tab and the **Template** tab appear in the right pane of the Type Manager window. The **General** tab displays details about the selected type:



- Click the **New** icon.

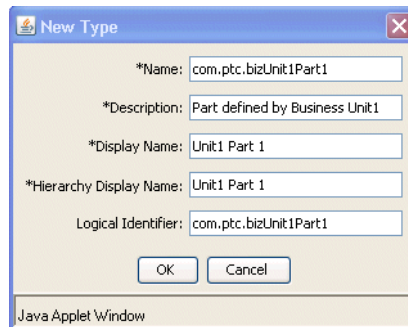
The **New Type** window appears.



In the window, the **Name** field contains a suggested prefix for the type (in this case, **com.ptc**). Including the suggested prefix ensures that the soft type being created is unique and is associated with the context from which you launched the Type Manager. For additional details on including a prefix in the type name, see [Associating Soft Types with the Site and Organization Contexts](#).

The name should be unique among siblings and can have a maximum of 200 alphanumeric characters and underscores. Hyphens are not allowed in type names and underscores are not allowed in the internet domain.

5. Fill in the each of the fields for the new soft type using "bizUnit1Part1" as the Name, as shown in the following figure:



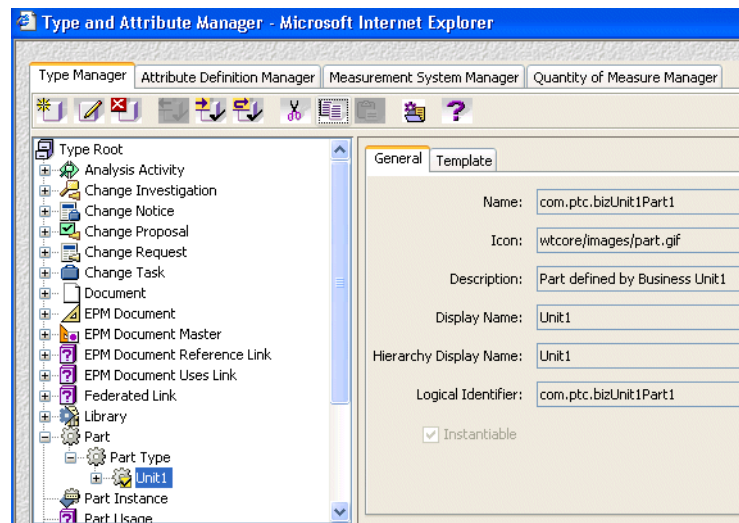
A screenshot of the 'New Type' dialog box. It contains the following fields: \*Name: com.ptc.bizUnit1Part1, \*Description: Part defined by Business Unit1, \*Display Name: Unit1 Part 1, \*Hierarchy Display Name: Unit1 Part 1, and Logical Identifier: com.ptc.bizUnit1Part1. There are OK and Cancel buttons at the bottom. The window title is 'New Type' and it is a Java Applet Window.

The **Logical Identifier** field used to identify the type in external configuration and properties files. It is often the same as the type name and should be unique across all types in the system.

A Logical Identifier must be defined for the soft type to make it searchable. It is limited to 200 characters made up of only letters, numbers, periods, and underscores. For more information, see the [Specifying Windchill Types](#) section.

6. Click **OK**.

The new Unit 1 soft type appears in the hierarchy as a subtype of Part Type, as shown in the following figure.



A screenshot of the 'Type and Attribute Manager' application running in Microsoft Internet Explorer. The 'Type Manager' tab is active. The left pane shows a tree view of the type hierarchy, with 'Unit1' selected under 'Part Type'. The right pane shows the 'General' tab for the selected type, with the following fields: Name: com.ptc.bizUnit1Part1, Icon: wtcore/images/part.gif, Description: Part defined by Business Unit1, Display Name: Unit1, Hierarchy Display Name: Unit1, Logical Identifier: com.ptc.bizUnit1Part1, and a checked 'Instantiable' checkbox.

**Note:** The new type is checked out as indicated by the yellow check mark and is unavailable for use in the system until it is checked in.



When you are ready to make the type available for use in the system, ensure the icon is selected and then click the Check In icon.

By default, the **Instantiable** check box is selected. When the type is instantiable, the type appears in the list of available types when you are creating an object such as a part or document. The check mark can be removed from the type by using the Edit icon.

## Creating a Soft Attribute

The following procedure explains how to create a soft attribute as part of the overall example described in the [Using the Type and Attribute Manager](#) section.

In this procedure you will add the new attributes (cageCode, procurementLeadtime, system, and spare) and associate them with the soft type named Unit 1 that was created in the [Creating a Soft Type](#) example.

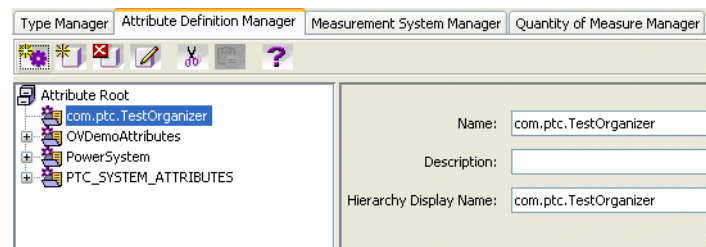
The attributes must reside in the [soft attribute set](#) to be accessible from the **Type Manager** tab. Soft attributes can be created using the [Attribute Definition Manager Tab](#) at any time, even before opening the Type Manager and creating a type. If you are already using the Type Manager, neither it nor the method server need to be restarted for the new attributes to be available for use.

1. Start the **Type and Attribute Manager** utility from the **Site Utilities** page, as described in the [Accessing the Type and Attribute Manager Utility](#) section.
2. Select the **Attribute Definition Manager** tab.

Prior to creating an attribute, you need to create an organizer. Attributes are organized using *organizers*, which serve no other purpose than to group the attributes, much like a folder is used in a Windows environment. Organizers can only exist beneath the root or other organizers; therefore, they cannot be children of attributes.

3. Click the Create Organizer icon and name it TestOrganizer.

The new organizer appears beneath the **Attribute Root** in the tree view.



4. Select **TestOrganizer** from the tree view and then click the New Attribute icon.
5. Enter the name **cageCode** in the **Name** field.
6. From the **Data Type** drop-down list, select **Integer Number**.

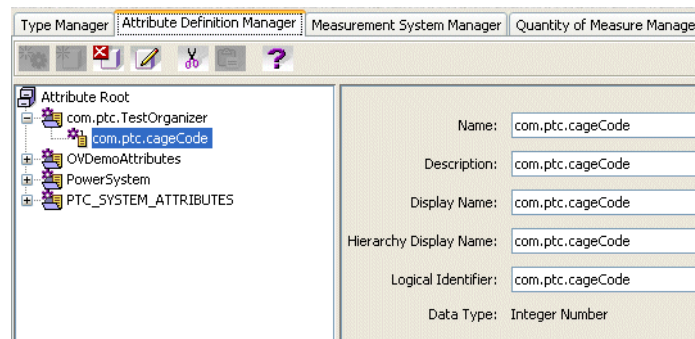


7. In the **Logical Identifier** field, enter a name to identify the type in external configuration and property files. It is often the same as the attribute name and should be unique across all types in the system.

The logical identifier is limited to 200 characters made up of only letters, numbers, periods, and underscores. To make the attribute searchable, use a logical identifier.

8. Click **OK** to create the new attribute.

The attribute **cageCode** now appears under the **TestOrganizer**, as shown in the following figure:



Use the steps in this section to add the remaining attributes. Designate the attributes with the following data types:

- *procurementLeadtime*, a range constraint data type
- *system*, a string type
- *spare*, a boolean value data type

## Adding an Attribute to a Type

Use the following procedure to associate an attribute with a type.

1. Select the **Type Manager** tab.
2. Click the **Template** tab.

The **Template** tab displays the types' attributes and their default values. Initially, no attributes appear. (Although they do not appear on this tab, all modeled attributes of the parent type are inherited by the new type.)

3. Select the Unit1 soft type that was just created and then click the Edit icon.

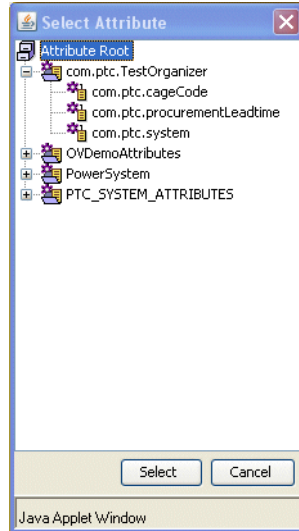
The type to which you want to add an attribute must be in edit mode prior to making any modifications.

4. Click on **Attribute Root** and then click **Add Attribute**.

The **Select Attribute** window opens.

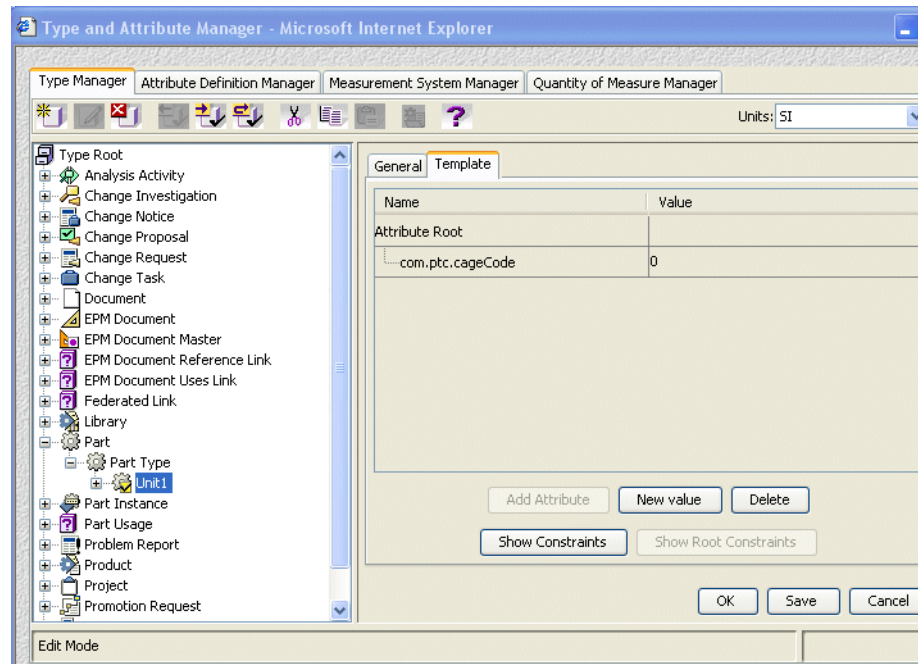
**Note:** All actions for the buttons at the bottom of the **Template** page are also available through a right-click pop-up menu.

5. Expand the attributes available under **TestOrganizer** and select the desired attribute, for this example select **cageCode**.



6. Click **Select**.

The window closes, and you are returned to the **Type Manager** window, which now lists the new attribute **cageCode** and a default value of 0.



7. If you want to change the default value, click in the **Value** field corresponding to the new attribute and enter a different value.

A default value is required for each attribute you add. The value is required to ensure there is at least one value for the attribute that is capable of satisfying the constraints you may define.

**Note:** You are not allowed to check in any changes to types that violate their own constraints.

8. Add the remaining attributes in the same manner.

## Setting Constraints on Attributes

Constraints can be set on attributes to determine whether they are editable and visible, and to limit their values and their cardinality.

All constraints are enforced during all operations, except those constraints which limit an attribute from being editable or visible. Specifically, the **Hidden** and **Immutable** constraints can be configured to apply only during specific operations. For example, if an Immutable constraint is applied during an Update operation, its result will be an attribute that can only be set during the initial object creation.

See the [Out-of-the-Box Constraints](#) section for a list of constraints and their descriptions.

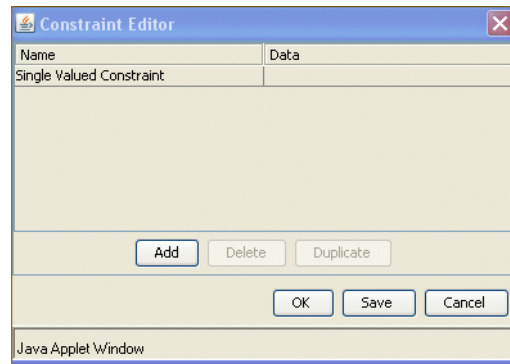
By default, when a soft attribute is added to a type via the **Type Manager** tab, or is loaded at the time of installation, each attribute will have a **Single Valued Constraint** applied. This constraint can be removed via the **Type Manager** tab.

Use the following procedure to set constraints on attributes:

1. Ensure the **Template** tab is the active tab.
2. Determine how you want to set the constraint:
  - Set a constraint for a specific attribute by clicking on the attribute and then click **Show Constraints**.
  - Set constraints that apply to all attributes of the type by clicking on the **Attribute Root** field and then click **Show Root Constraints**.

**Note:** When setting constraints on all attributes, the constraints will not be displayed on the individual attributes.

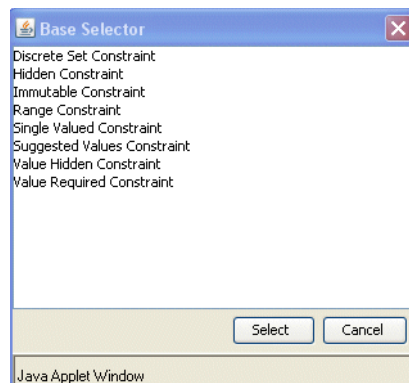
The **Constraint Editor** window appears, showing any existing constraints. Note the **Single Valued Constraint** appears on the attribute.



3. Click **Add**.

The **Add**, **Delete**, and **Duplicate** are buttons in the **Constraint Editor** window. You also can right-click on any constraint to display a pop-up menu with the same options.

The **Base Selector** dialog box appears, showing the available constraints:



4. Select the desired constraint and click **Select**.

For example, to set a range for the procurementLeadtime attribute, select the **Range Constraint** constraint and click **OK**.

The **Base Selector** window closes and you are returned to the **Constraint Editor**.

If you selected the Range Constraint, it now appears in the **Constraint Editor**.

5. Click inside the **Data** field and modify the two values (**From** and **To**) for the range of the attribute, as shown in the following figure:

Name	Data
Range Constraint	From: 3 To: 5
Single Valued Constraint	

Add Delete Duplicate

OK Save Cancel

Java Applet Window

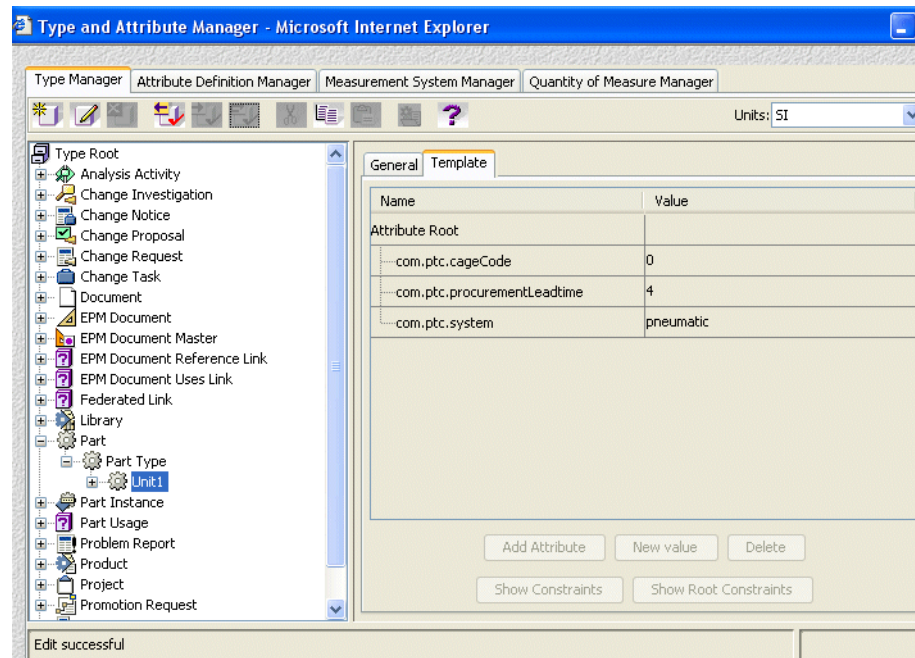
When you are finished, click **OK** to close the **Constraint Editor** window and return to the **Type Manager** window.

**Note:** You can remove the Single Valued Constraint by selecting it and then click **Delete**.

6. Specify values for the attribute named **system**, a String type.
- Select the system attribute on the **Template** tab, click **Show Constraints**, then click **Add** to return to the **Base Selector**.
  - In the **Base Selector** window, select **Discrete Set Constraint** and click **Select**.
  - In the **Constraint Editor** window, enter the possible values for the system attribute (electrical, hydraulic, and pneumatic) in the **Data** field. Separate each value with a vertical bar (|).
  - Delete the **Single Valued Constraint**.
  - Click **OK** to close the window and return to the **Type Manager** window. When the constraint is saved, the Constraint Editor checks for a vertical bar at the end of the list and adds one if it is not already there.

**Note:** By adding this type of constraint, you will get an error message if one of the values is not selected in the **Value** column.

The new attributes are created, and have associated constraints and default values as shown in the graphic below:



- f. Click **Save** to save the attributes and return to view-only mode.
- g. Check in the new type by selecting it and clicking the Check In icon. The type is now available for use in the system.

## Editing a Type Definition

A type can be edited from either the site or the organization context. When the **Type Manager** tab is launched in the context of an organization, you can only edit types associated with that organization. (The names begin with the internet style domain prefix of the organization.)

If you remove an attribute from a type that already has data associated with it, the system does not remove the attribute's values from any existing instances of that type. Additionally, if an attribute is added to a type that already has data associated with it in the database, the newly added attribute is not automatically associated with existing instances of that type.

Hyphens are not allowed in type names and underscores are not allowed within the internet domain.

The following procedure is an example of editing a type and is based on the preceding example. A boolean attribute named spare will be added to the type.

Before an attribute can be added to a type, it must exist in the [soft attribute set](#).

You can add an attribute using the same procedure given in the earlier example, or you can do it before or during your Type Manager session.



The following example assumes the spare attribute has already been created:

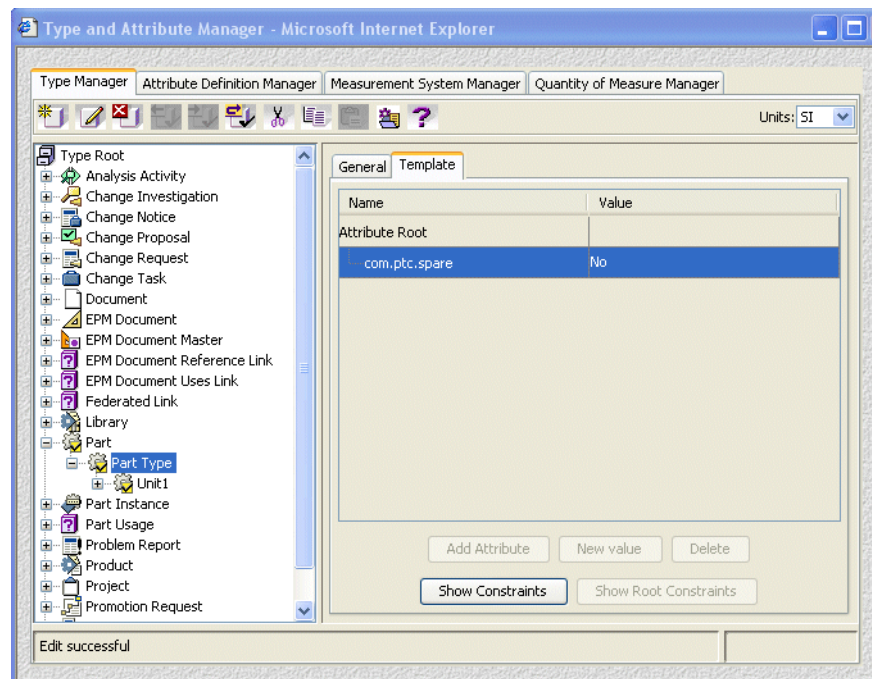
1. In the **Type Manager** window, select the type **Part** and click the **Edit** icon.

The type is checked out automatically and no longer available for use in the system.

2. Follow the procedure below to add an attribute and set a default value:
  - a. Click the **Template** tab.
  - b. Click **Attribute Root** and then click **Add Attribute**.
  - c. In the **Select Attribute** window, expand the **TestOrganizer** node and select the spare attribute. Click **Select**. The spare attribute has a boolean type.

The spare attribute is added with a value of **No**.

- d. If you want the default value to be yes, select **Yes** from the drop-down list in the **Value** field.
3. Click **OK** to save the new attribute.
  4. Click the **Check in** icon to check in the updated type and make it available for use in the system.



New instances of the **Part** type now have the spare attribute. Existing instances can wait until the first time they are accessed to be updated. At this time, the spare attribute, and its default value, are added to the instance. This is known as lazy migration and prevents excessive activity in the database when a new attribute is added or deleted.

## Deleting a Type

A type can be deleted from either the site or the organization context. To delete or modify types, a user must either be a site administrator or an organization administrator.

When you launch the **Type Manager** tab in the context of an organization, you can delete only types associated with that organization. (Their names will begin with the internet style domain prefix of the organization.) You can delete a [soft type](#) but you cannot delete a [modeled type](#).

Deleting a type also deletes all of its descendants. If you delete a type that has already been used to create objects in the database, the system does not remove the any existing instances of that type.

The online help provides a detailed procedure for deleting a type. The type must be checked out prior to deleting a type or a subtype and once deleted, it cannot be reversed.

**Note:** When a parent type is checked out and a subtype is deleted, it will result in the subtype being permanently deleted. The **Undo Check Out** option will not recover the deleted subtype.

## Out-of-the-Box Constraints

Constraints can be set on attributes to determine whether they are editable, visible, and to limit their values, and their cardinality.

The **Immutable** and **Hidden** constraints can only be applied on Create and Update operations. EPM Documents can also be applied to the EPM Upload operation. For example, if an attribute has one of these constraints applied to an Update operation, then the attribute will only be applicable when an instance of the type is being edited. If an attribute has one of these constraints applied to a Create operation, then the attribute will only be applicable when a new instance of the type is being created.

When EPM Upload is applied to an EPM Document, it is not possible to upload a CAD Document (EPM Document) through the Pro/ENGINEER application unless the following property is set:

```
wt.services/svc/default/wt.epm.attributes.EPMAttributeDelegate/  
PROE/wt.iba.value.IBAHolder/0=wt.epm.attributes.EPMAttributeDel  
egateWithWarnings/singleton
```

By setting the service.properties file using the xconfmanager tool, the parameter will be ignored and the upload will be successful. The event console will show the error message.



The following table provides information about the out-of-the-box constraints:.

Type of Constraint	Applies to Attributes of Data Type	Can Apply to Context	Description	Example of Constraint Value
Discrete Set Constraint	All	No	<p>The attribute value must be the same as one of the specified constraint values.</p> <p>A discrete set constraint is a list of legal values. Any value that a user supplies for the attribute must be in that list.</p>	<p>String data type: constraint value set: abc cde efg</p> <p>Legal strings can be abc, cde, or efg</p> <p>Integer data type: constraint value set: 1 2 3</p> <p>Legal integer value can be 1, 2, or 3</p> <p><b>Note:</b> ' ' is the delimiter for the string values. Currently, ' ' is the reserved character.</p>
Hidden Constraint	Yes	Yes	<p>The attribute and its value is hidden to the end user, but remains visible to the administrator on the <b>Type Manager</b> tab.</p>	<p>Constraint Value: <b>Create</b> or <b>Update</b>.</p> <p>The <b>Hidden</b> constraint can be configured to apply only during specific operations; create or update. For example, if create is applied to a hidden attribute, that attribute will be unchangeable when creating the part in PSE and the Windchill user interface, but it will be editable when the part is being updated.</p> <p>The default is <b>Update</b>. On migrated systems, these constraints will be treated as if they had <b>Update</b> applied.</p> <p>All attributes that are not Integer, String, Boolean, or Real Number will be treated as if they have an implicit <b>Hidden</b> constraint applied to the <b>EPM Upload</b> operation.</p>

Type of Constraint	Applies to Attributes of Data Type	Can Apply to Context	Description	Example of Constraint Value
Immutable Constraint	All	Yes	<p>The user cannot change, add, or delete the value of the attribute except during initial object creation.</p> <p><b>Notes:</b> To add this constraint, the user has to save the attribute value first.</p> <p>After this constraint is imposed to the container, the container is frozen. It cannot be modified any further.</p> <p>In Windchill PDMLink and Windchill ProjectLink, attributes with immutable constraints cannot be modified after the object is created; when you update an object, immutable attributes are present but grayed out.</p>	<p>Constraint value: <b>Create</b> or <b>Update</b>.</p> <p>The <b>Immutable</b> constraint can be configured to apply only during specific operations. For example, if the value <b>Update</b> is applied to an <b>Immutable</b> constraint, it's result will be an attribute that can only be set or edited during the initial object creation. The attribute will be immutable during an update to the object.</p>
LowerCase Constraint	Strings	No	This constraint requires the value to be lower case and will convert whatever case that is entered into lower case.	<p>No constraint value.</p> <p>If "MyValue" is entered, it is converted to "myvalue".</p>
Range Constraint	Can be applied to all data types, but will not be effective on the boolean data type.	No	<p>The actual value of the attribute must be greater than or equal to the maximum values specified (the range is inclusive)</p> <p><b>Note:</b> "From:" specifies the minimum value. "To:" specifies the maximum value.</p>	<p>Value: A to Z</p> <p>Legal strings can be Abc or BCDE.</p>

Type of Constraint	Applies to Attributes of Data Type	Can Apply to Context	Description	Example of Constraint Value
Single Valued Constraint	All	Yes	<p>No more than one value is allowed for a specific attribute. When applied to Container, no attribute can have more than one value.</p> <p><b>Note:</b> This constraint is not applicable in Windchill PDMLink and Windchill ProjectLink; multiple values are not allowed for attributes.</p>	No constraint value.
String Format Constraint	Can be applied to all data types, but only effective on the String data type.	No	<p>Provide a set of basic masking to regulate the format of a string. The constraint value is a set of strings defining positional formats for the string content. C, L, and D in constraint value are reserved characters and should not be used as delimiters; all the other characters are considered delimiters.</p> <p>C means one letter or one digit. L means one letter. D means one digit.</p> <p>The definitions of letter and digit can be found in Java.lang.Character Class.</p>	<p>1. SSN Formatting value: DDD-DD-DDDD. Legal strings can be 123-45-6789, 452-98-4444.</p> <p>2. Telephone number Formatting value: (DDD)DDD-DDDD   DDD-DDD-DDDD   D-DDD-DDD-DDDD. Legal string can be (555)454-6789, 555-198-3247, 1-800-436-7869, 1-800-CAN-HELP.</p> <p>3. Air flight seat number Formatting value: DL D-L. Legal strings can be 1A, 3-E.</p>
String Length Constraint	Can be applied to all data types, but only effective on the String data type.	No	<p>The length of the string value must be greater than or equal to the minimum, and less than or equal to the maximum values specified (the range is inclusive).</p> <p><b>Note:</b> "From:" specifies the minimum length. "To:" specifies the maximum length.</p>	<p>From: 3 To: 200. 3 &lt;= length of legal string &lt;= 200.</p>

Type of Constraint	Applies to Attributes of Data Type	Can Apply to Context	Description	Example of Constraint Value
Suggested Values Constraint	All	No	Provide a set of suggested values to the Constrainingable.  <b>Note:</b> In Windchill PDMLink and Windchill ProjectLink, these values are presented in a tool tip on HTML object creation and update forms.	String data type: constraint value set: abc cde efg.  Legal strings can be abc, cde, or efg.  Integer data type: constraint value set: 1 2 3.  Legal integer value can be 1, 2, or 3.  <b>Note:</b> ' ' is the delimiter for the string values. Currently, ' ' is the reserved character.
UpperCase Constraint	Can be applied to all data types, but only effective on the String data type.	No	The String values are converted to uppercase when saved.	No constraint value.
Value Hidden Constraint	All	Yes	The value of the attribute is hidden to the end user, but remains visible to the administrator on the <b>Type Manager</b> tab.  In the user interface and in PSE the value will be displayed as <b>Hidden</b> .	No constraint value.
Value Required Constraint	All	Yes	The attribute must have at least one value.	No constraint value.

Type of Constraint	Applies to Attributes of Data Type	Can Apply to Context	Description	Example of Constraint Value
Wildcard Constraint	Can be applied to all data types, but only effect on the String data type.	No	<p>The <b>String</b> attribute value must match the Wildcard pattern of constraint value specified.</p> <p>Contains: Contains the value specified.</p> <p>Begins With: Begins with the value specified.</p> <p>Ends With: Ends with the value specified.</p> <p>Exact: Exactly the same as the value specified.</p>	<p>'Contains' abc.</p> <p>Legal strings can be Ababc, abcZ, AabcZ.</p> <p>'Ends With' er.</p> <p>Legal strings can be Aer, Developer.</p>

## Specifying Windchill Types

When you are working with the types that are available from the Type and Attribute Manager, you need to be aware of how to identify the types. For example, you can specify types in object initialization rules, properties, preferences, publishing to Arbortext, and custom code.

When specifying an object type, use the logical identifier of the type.

The logical identifier of a type can be set when out-of-the box types are loaded or can be set when you create (or update) a type through the Type Manager.

The logical identifier must be unique across all types in the system and is limited to 50 characters, with the restriction that only letters, numbers, periods, and underscores are supported.

Often, the logical identifier is the same as the type name. If you are unsure of whether a type has a logical identifier, you can view the information from the **General** tab of the Type and Attribute Manager.

**Note:** Logical identifiers can be used in custom Info\*Engine tasks and other custom code where there is a need to refer to a type.

**Tip:** If the type does not have a logical identifier, you can add one by editing the type using the Type and Attribute Manager.



**Caution:** After a logical identifier has been set and used in your solution, do not change the identifier. Changing the identifier after it is being used can cause problems.

## Managing Types and Attributes for Parts, Documents, and Change Objects

The create action for parts, documents, and change objects includes a **Type** drop-down list from which the user can select the type of object to be created. You can define defaults for type lists, and exclude types from appearing in a type list. This section also provides some information that can help you understand how soft attributes are displayed.

The following table describes the preferences affecting the type drop-down list. These preferences can be found under the Create and Edit category in the Preference Manager. The possible values are listed within each preference:

Preference	Description
Document Create Default Type	Specifies the default type for creating documents.
Part Create Default Type	Specifies the default type for creating parts.
Change Task Create Default Type	Specifies the default type for creating change tasks.
Problem Report Create Default Type	Specifies the default type for creating problem reports.
Change Notice Create Default Type	Specifies the default type for creating change notices.
Change Request Create Default Type	Specifies the default type for creating change requests.
Variance Create Default Type	Specifies the default type for creating variances.

## Excluding Types from a Types List

By default, the **Type** drop-down list displayed in the create windows includes all creatable and instantiable types, starting with the root type (WTPart or WTDocument), and includes all subtypes, both soft and modeled.

To exclude types from the list, use the following properties in the wt-properties file. These properties contain a delimited list of types to be excluded from the **Type** drop-down list. If the property already exists, you can add types to the list. If the property does not exist in your wt.property file, you can add it.

Object	Property
Part	wt.typepickerexclude/wt.part.WTPart
Document	wt.typepickerexclude/wt.doc.WTDocument
Change Issue	wt.typepickerexclude/wt.change2.WTChangeIssue
Change Request	wt.typepickerexclude/wt.change2.WTChangeRequest2
Change Notice	wt.typepickerexclude/wt.change2.WTChangeOrder2
Change Activity	wt.typepickerexclude/wt.change2.WTChangeActivity2

Use the logical identifier of the type when setting the properties. For example, if your site has a custom modeled subtype of WTPart called ext.part.MyPart (with a logical identifier of the same name) and a soft type of MyPart called com.mycompany.foo.MySoftPart (with a logical identifier of the same name), and you want to exclude these types from the **Type** drop-down list, then you can set the following property:

```
wt.typepickerexclude/wt.part.WTPart=ext.part.MyPart,com.mycompany.foo.MySoftPart
```

By default, a comma is used to separate types in the list. However, that default can be overridden by modifying the wt.property:

```
wt.typepickerexcludelimiter=,
```

**Note:** Modify the wt.properties file using the xconfmanager utility. For details on using the xconfmanager utility, see [About the xconfmanager Utility](#). For information on specifying the logical identifier of the type, see [Specifying Windchill Types](#).

## Excluding Types from Simple and Advanced Search Pages

The list of types appearing on the Simple Search Page and the Advanced Search Page can be configured so that specific types are filtered out of the Type Picker.

The following wt.property files apply only to the search pages:

- com.ptc.windchill.enterprise.search.client.typeChooser.allSearchExclude

This property is a comma separated list of types that will get filtered while building the list of types that are displayed in the Type Picker on both the Simple Search and Advanced Search Pages.

The default value of this property is  
"wt.part.WTProduct,wt.part.WTSerialNumberedPart".

A site or organization administrator can add more types to the list for exclusion from both search pages. This list of types applies to all users; there is no user or site-specific control.

- com.ptc.windchill.enterprise.search.client.typeChooser.simpleSearchExclude

This property is a comma separated list of types that will get filtered out while building the list of types that are displayed in Type Picker on the Simple Search Page only.

**Note:** This filter does not apply to the Advanced Search Page.

The default value of this property is "" (blank).

A site or organization administrator can add more types in this list separating them with a "," (comma).

## Understanding Soft Attribute Behavior on Object Create and Update Windows, and on Information Pages

The following list describes the soft attribute behavior on object create and update windows:

- Soft attributes are not marked as required on create and update windows unless they are given a **Value Required** constraint in the Type Manager. If a nonrequired attribute is not given a value on a create or update window, it is displayed without a value on the object information page.
- The default values given for attributes in the Type Manager are not prepopulated in the attribute input fields on the create and update windows. If the attribute has not yet been set for the object, and the attribute has a default value, a **Use Default** button is displayed to the right of the input field. Clicking this button sets the attribute to the default value. Input fields for custom modeled attributes also have a **Use Default** button if a default value was specified for the attribute.



If your site does not want to display a **Use Default** button for a given attribute, the button display can be turned off by customizing using the AllClients.xml configuration file. For customization details, see the *Windchill Customizer's Guide*.

- Multi-line input fields are displayed for string type soft attributes if the maximum permitted length of the string exceeds the value of the **String Length Threshold For Multiline Input** preference. The maximum length of string soft attributes is 500 characters by default, but can be set to a smaller value using the **String Length** constraint in the Type Manager.

A multi-line input field is defined as a field in which multiple lines of text can be entered. It is not necessarily more than one line in height.

You can override this behavior by specifying that a single-line or multi-line input field is to be used for a given attribute in the AllClients.xml configuration file.

- On create and update windows required modeled attributes are listed before nonrequired modeled attributes; each group of attributes is sorted alphabetically by display name.
- On information pages, the attributes are sorted alphabetically by display name with no differentiation of required and nonrequired attributes.

## Managing Types and Attributes for CAD Documents

Data files created by CAD applications and saved to Windchill using a workgroup manager or Pro/ENGINEER Wildfire are managed in Windchill as CAD documents.

There is one main out-of-the-box soft type that is used for CAD documents. This soft type is defined as a subtype under the wt.epm.EPMDocument modeled type. The soft type is named *<prefix>*. DefaultEPMDocument, where *<prefix>* is generated by reversing the internet domain specified for the default organization during the installation. For example, if the internet domain specified is acme.com, then the *<prefix>* used in the soft type name is com.acme. This same prefix is used for all out-of-the-box soft types that are automatically loaded.

The soft type used for CAD documents cannot be deleted nor can you create subtypes under this subtype.

You can perform the following actions with regard to the CAD document soft type:

- Add new attributes for the soft type.
- Define constraints for any existing attribute.
- Update the type, attributes, and constraints.

- Delete attributes.
- Map attributes to specific parameters in other applications.

To assign attributes to CAD documents, add the attributes to the default CAD document soft type. All attributes are defined globally using the **Attribute Definition Manager** tab.

The attributes you add can be optional or required, and constraints can be supplied for any of the new attributes. Constraints specify characteristics, such as range restrictions or a valid value set. You can also adjust or provide constraints on inherited attributes. If there is an Immutable Constraint applied to an attribute, the attribute cannot be removed from the type definition. For additional information about setting constraints, see [Setting Constraints on Attributes](#).

**Tip:** To avoid difficulties based on differences in how Windchill and various CAD editing applications assign significant figures to restricted attribute values, PTC recommends that you use the **Significant Figures Set Constraint**, rather than **Discrete Set Constraint**, for constraining real number soft attribute values. This allows the system to accept values within a reasonable range of precision on either side of the desired value.

## Renaming the CAD Document Soft Type

Pro/ENGINEER Wildfire assumes that the out-of-the-box CAD document soft type has not been renamed. PTC recommends that you do not rename the out-of-the-box CAD document soft type rather than changing the **Display Name**, which is visible in the client.

You can rename the soft type; however, the soft type needs to be recognized by the system in order to assign the proper default soft type. Therefore, if you must change the name, you also need to record the change by setting the following site level preference. Otherwise, the system will fail to recognize the soft type as the default type for CAD documents.

The preference **CAD Document default soft type** can be found in the site context under the category EPM Service Preferences > Soft Type Preferences. Set this preference to the logical identifier of the soft type using the Preference Manager.

## Additional Soft Types Related to CAD Documents

Windchill provides the Workgroup Manager CAD Document soft type as the subtype for adding attributes that can have different values for each iteration of the item. Although this is probably the most commonly used type for CAD documents, you should be aware of the other soft types that can be used. In all of the following soft types, *<prefix>* is generated when the type is loaded by reversing the internet domain specified for the default organization during the installation:

- Workgroup Manager CAD Document Master soft type (*<prefix>.DefaultEPMDocumentMaster*) is a subtype of the EPM Document Master type (wt.epm.EPMDocumentMaster).

Attributes that are added to this soft type have only one value for all iterations. Therefore, changing the value of an attribute on a Workgroup Manager CAD Document Master soft type changes that value for all iterations.

Attributes added on this type are the Windchill equivalent of Pro/INTRALINK non-versioned attributes.

- Workgroup Manager CAD Document Uses Link (*<prefix>.DefaultEPMMemberLink*) soft type is a subtype of the EPM Document Uses Link (wt.epm.structure.EPMMemberLink) type.

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

- Workgroup Manager CAD Document Reference Link (*<prefix>.DefaultEPMReferenceLink*) soft type is a subtype of the EPM Document Reference Link (wt.epm.structure.EPMReferenceLink) type.

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

**Tip:** Do not add attributes with the same name to different soft types. By using different names you can avoid confusion for users when they are adding and displaying the attribute values.

## Mapping Attributes to CAD Parameters

Windchill attributes can be mapped to specific CAD parameters in other applications.

Currently only Windchill attributes on the EPM Document type, EPM Document Reference Link type, and EPM Document Uses Link type can be explicitly mapped to parameters in Pro/ENGINEER Wildfire and other CAD applications. Mapping the attributes on the Workgroup Manager CAD Document, a subtype of the EPM Document types listed above, publishes Windchill information to the Pro/ENGINEER system and other CAD applications.

Attributes are mapped to [CAD parameters](#) using either [implicit mapping](#) or [explicit mapping](#). Implicit mapping refers to the system matching the attribute and the CAD parameter automatically. No user intervention is necessary.

Implicit mapping is recommended as the preferred mapping method because released data could be changed without a new iteration being reviewed. If a change is made to an explicitly mapped attribute and then a file is downloaded, you might see a different result than before the change to the mapping was done.

### Implicit Mapping

Implicit attribute-to-parameter mapping occurs when there is a soft attribute in Windchill with a name (all uppercase) identical to the name of a designated parameter in a Pro/ENGINEER model file, or other CAD application. When the CAD parameter is uploaded into Windchill as content of a CAD document, the values of the CAD parameter are transferred to the Windchill attribute.

When Windchill attributes are implicitly mapped to CAD parameters, they are [file-based attributes](#), that is, they have full access in both Windchill and the CAD application. For more information, see the [Setting Attribute Access in CAD Applications and Windchill](#) section.

### Explicit Mapping

The site and organization administrator can specify per soft type, whether or not a Windchill attribute is explicitly mapped to a parameter in another application.

When any of the types listed above, or their subtypes are selected from the **Type Manager** tab, the **Attribute Mapping** tab is displayed. This tab allows attributes to be explicitly mapped to specific CAD parameters. This mechanism replaces the old process of editing the iba.properties file to create [explicit mapping](#).

See the Type and Attribute Manager online help for detailed procedures for creating explicit attribute-to-parameter mappings.

## Setting Attribute Access in CAD Applications and Windchill

Windchill attributes can be set with different levels of access in both CAD applications and Windchill. An attribute can either be editable in both applications, or it can be designated to be read-only in one application and editable in the other.

The following list describes the available combinations. For detailed procedures on setting attribute access, see the Type and Attribute Manager online help.

- Windchill attributes with full access in CAD applications and Windchill

The values of this type of attribute can be edited or deleted in both the Windchill user interface and the CAD application. This is also known as a [file-based attribute](#).

- Windchill attributes that are read-only in CAD applications, but have full access in Windchill

The values of this type of attribute can be edited or deleted in the Windchill user interface and is read-only in the CAD application. This is also known as a file-based attribute.

- Windchill attributes that have full access in CAD applications but are read-only in Windchill

The values of this type of attribute can be edited or deleted in the CAD application and is read-only in the Windchill user interface. This is also known as a file-based attribute.

- Windchill attributes that have full access in Windchill but are hidden from CAD applications

The values of this type of attribute can be edited or deleted in the Windchill user interface and is not visible in the CAD application. This is also known as a [nonfile-based attribute](#).

## Managing Types and Attributes for Dynamic Documents

Data files created by the Arbortext Editor and saved to Windchill using the Windchill connection to the editor are managed in Windchill as dynamic documents.

There are two out-of-the-box soft types defined for use with the Arbortext Editor. Both reside under the wt.epm.EPMDocument modeled type and provide the initial set of soft types defined for dynamic documents.

The soft type named `<prefix>.DynamicDocument` (where `<prefix>` is generated by reversing the internet domain specified for the default organization during the installation) is a subtype of EPM Document and is used to identify XML files authored in the Arbortext Editor and to identify graphic files referenced in the XML files.

The soft type named *<prefix>.BurstConfiguration* is a subtype of the Dynamic Document soft type and is used to identify burst configuration files stored in Windchill. Burst configuration files provide the details of how to split up a document into individual dynamic documents based on user-defined rules. For example, a document could contain multiple chapters and the burst configuration file might have a rule that says to store each chapter as a separate dynamic document in Windchill.

## Renaming Dynamic Document and Burst Configuration Soft Types

The current Windchill PDMLink implementation with Arbortext assumes that the loaded soft types are named as described in the previous section. PTC recommends that you do not rename the out-of-the-box soft types other than changing the **Display Name**, which is visible in the client.



**Caution:** If you rename the soft types, the soft types need to be recognized by the system in order to assign the proper default soft types. Therefore, if you change the names, the system will fail to recognize the soft types as default types unless you also record the changes by setting the following site level preferences:

- The **Dynamic Document default soft type** preference controls the default type established for dynamic documents and can be found in the Windchill Workgroup Manager > Client category.
- The **Dynamic Document Burst Configuration soft type** preference controls the default type established for bursting configuration files and can be found in the Windchill Workgroup Manager > Client category.

Set the value of each of these preferences to the logical identifier of the soft type using the Preference Manager. For details on the required type format, see the [Specifying Windchill Types](#) section.

## Creating Additional Soft Types for Use with Arbortext Editor and the Arbortext Publishing Engine

PTC recommends that you define a different soft type for each XML element you want to associate with bursting rules (for example, chapters and sections). In the bursting configuration file that is managed from the Arbortext Editor, those setting up Arbortext must specify the soft types that are applied at the time that documents are burst into Windchill. For information on bursting, see the Arbortext Editor online help.

The soft types defined are also used in the PublishRules file for configuring representations and are available to users when they browse Windchill for dynamic documents.

After the following activities are complete, users can select a specific soft type to narrow searches for objects and the soft types are available for use when bursting dynamic documents and when creating representations (such as PDFs) from dynamic documents:

- Windchill connection has been installed and configured for the Arbortext Editor. For details, see the readme file delivered with the Windchill connection and the Arbortext Editor online help.
- Arbortext Publishing Engine is installed and configured. For details, see the *Windchill System Administrator's Guide*.
- Related soft types have been created.

The following is an example that shows the mapping of Arbortext XML elements and Windchill Dynamic Document soft types.

Assume the following Arbortext XML elements are the levels to be burst:

```
book
chapter
sect1
```

To support these bursting levels in Windchill PDMLink, create three soft types. Each soft type is used for one of the Arbortext XML elements. For example, soft types with the following display names could be created:

```
Book
Chapter
Section
```

For details on creating soft types, see the online help available from the Type Manager. For examples of creating and editing types, see the [Using the Type and Attribute Manager](#) section.

**Note:** The type names specified in the Type Manager include a prefix that identifies the owning organization for the type. A suggested prefix is always presented when you are creating a new subtype and is generated based on the internet domain associated with the owning organization principal of the context from which the type is being created. For additional information, see the [Associating Soft Types with the Site and Organization Contexts](#) section.

After the soft types are created, communicate with those setting up the Arbortext bursting configuration file and the PublishRules file to tell them which soft type they should associate with each Arbortext XML element. The following section describes the format of the type name that is required in Arbortext.

## Specifying the Logical Identifier for Types Used with Arbortext

Setting up document bursting in Windchill for use with the Arbortext files requires the use of logical identifiers for all types associated with dynamic documents. This is because the types are referenced in files within the Arbortext Editor and must be unique across the entire system.

To determine the logical identifier for a type, view the type from the **Type Manager** tab. For details on using the Type Manager, see [Using the Type and Attribute Manager](#).

On the **General** tab, there is a **Logical Identifier** field that contains the value that should be used for the type in Arbortext files. If the field is empty, you can update the type to add its logical identifier. For details on the logical identifier format, see [Specifying Windchill Types](#).

## Defining Windchill Soft Attributes for Attributes Used with Arbortext

As is the case with other types, you can create soft attributes in Windchill PDMLink that can be stored and used with objects that are associated with the Dynamic Document soft type or subtypes of Dynamic Document. Arbortext bursting rules can be used to populate the soft attributes from XML attributes when storing objects in Windchill. For example, if the author enters a vendor's name in an XML document, bursting rules can be set up so that the vendor's name is populated in a Windchill soft attribute, such as supplier. The benefit to setting up soft attributes for this purpose is that the author only enters the information in one location, but the information is accessible through the XML document and through the Windchill soft attribute.

Additionally, you may want to create a specific set of attributes that are associated with the representables created when publishing dynamic documents through the publishing agent. These attributes would be associated with the WTDocument type or a subtype of WTDocument. For information about using the publishing agent for publishing dynamic documents, see the *Windchill System Administrator's Guide*.

For more information, see [Creating a Soft Attribute](#).



## Best Practices

The following sections provide best practices for soft types and attributes in Windchill and for searchable types.

### Working with Multiple Organization Contexts

This section provides conceptual information about Windchill's ability to define and manage soft types and soft attributes in a site with multiple organization contexts.

Soft types and soft attributes can be created in either the site or an organization context. When soft types and soft attributes are created within an organization context, they are kept separate from all other organization contexts and remain private.

There are two scenarios where soft types and soft attributes can be seen and used by all organization contexts:

- When soft types and soft attributes are created by the site administrator in the site context, they are seen and available for use by all organization contexts.
- At the time of installation, the site context is associated with an owning organization principal. When that same owning organization principal is associated with an organization context, all organizations can see the soft types and attributes created within that organization context.

The information below is intended to provide an overview of the relationship between the site and organization contexts and how it applies to the creation of types and attributes.

Each Windchill system will have, at a minimum, one site context and one organization context, each with an associated owning organization principal. During the installation process, an organization principal is created and associated with the site context. When a site has multiple organization contexts, each will have an owning organization principal.

When an organization context is created, if the same organization principal that owns the site context is chosen as the owner for the new organization context, all types and attributes created within that organization context will be available in the site context, and therefore all organizations. To keep the soft types and soft attributes private to an organization, its owning organization principal must be unique from the organization principal associated with the site context.

Each organization principal has an internet domain that is used to identify the organization contexts for soft types and soft attributes. When soft types and soft attributes are created, Windchill provides a suggested prefix for the name of the new type or attribute. The suggested prefix is based on the context from which the **Type Manager** tab, or the **Attribute Definition Manager** tab was launched. The suggested prefix is the reverse of the internet domain for the owning organization principal of the organization context. For example, an organization with the internet domain, acme.com would suggest the prefix, com.acme.

**Note:** An error will occur if a type or an attribute is created within an organization context and it has an internet domain that is not associated with a known organization context.

## Associating Soft Types with the Site and Organization Contexts

When creating soft types from the **Type Manager** tab of the Type and Attribute Manager utility, the interface always provides a suggested prefix for the new type. The suggested prefix is based on the context from which the Type and Attribute Manager was launched. The suggested prefix is the reverse internet style domain derived from the internet domain defined on the organization principal owning the context.

**Note:** The internet domain defined on the organization principal is important when creating new soft types. During the installation process, a default organization principal is created that contains an internet domain associated with the organization principal. The internet style domain prefix for soft types (created by reversing the internet domain) is used to identify which organization owns the type. The default organization principal is associated with the site context; any types owned by this organization principal are available from all organization contexts.

Additionally, any soft types created with internet style domain prefixes that do not map to known organization principals are considered to be site types and are available from all organization contexts. In an environment where there is only one organization used, you can create the organization context using the default organization principal (as is the case when creating the organization context using the PDMLinkOrgContainer.xml described in the [Working With the Initial Organization Context](#) section of the Getting Started chapter). In this case, the soft types created from the site context and from the organization context have the same internet domain.

If you want to create soft types that are only available from a specific organization context, the organization principal used when creating the organization context must be different from the organization principal associated with the site context. In an exchange environment where there are multiple organizations, you can restrict access to the soft types created in an organization context to those members in the context. This is done by setting the internet domain defined in the organization principal associated with the organization context to a value different from the site internet domain. For information on changing the internet domain, see [Changing an Established Internet Domain](#).

## Restricting the Use of Soft Types and Soft Attributes

If you are not customizing the Windchill interface, restrict the use of soft types and soft attributes as described in the following paragraphs. If you do not restrict their use, client customizations are necessary to expose the new soft types and soft attributes.

From the **Site** and **Organization** tabs, administrators who can access the **Type Manager** tab (see [Accessing the Type and Attribute Manager Utility](#)) can define soft types for the following object types without requiring any client customizations:

Type Display Name	Type Name
Analysis Activity	wt.change2.WTAnalysisActivity
Change Investigation	wt.change2.WTChangeInvestigation
Change Notice	wt.change2.WTChangeOrder2
Change Proposal	wt.change2.WTChangeProposal
Change Request	wt.change2.WTChangeRequest2
Change Task	wt.change2.WTChangeActivity2
Document	wt.doc.WTDocument
EPM Document	wt.epm.EPMDocument
EPM Document Master	wt.epm.EPMDocumentMaster
EPM Document Reference Link	wt.epm.structure.EPMReferenceLink
EPM Document Uses Link	wt.epm.structure.EPMMemberLink
Federated Link	wt.federation.FederatedLink
Folder	Folder
Library	wt.inf.library.WTLibrary

Type Display Name	Type Name
Part	wt.part.WTPart
Part Instance	wt.part.WTProductInstance2
Part Usage	wt.part.WTPartUsageLink
Problem Report	wt.change2.WTChangeIssue
Product	wt.pdmlink.PDMLinkProduct
Project	wt.projmgmt.admin.Project2
Program	Program
Promotion Request	wt.maturity.PromotionNotice
Variance	wt.change2.WTVariance
Variant Specification	com.ptc.wpcf.doc.VariantSpec

For information on defining soft types for use with dynamic documents in Windchill PDMLink, see [Managing Types and Attributes for Dynamic Documents](#).

Soft attributes are defined in the site context through the **Attribute Definition Manager** tab. An organization administrator can define attributes if they have been added to the Attribute Administrators group, see [Administrator Roles](#).

From the **Site** and **Organization** tabs, administrators who have access to the **Type Manager** tab and the **Attribute Definition Manager** tab can define soft attributes and add them to types. For information on defining soft attributes for use with dynamic documents in Windchill PDMLink, see [Managing Types and Attributes for Dynamic Documents](#).

In a Windchill ProjectLink exchange environment, individual organization administrators can define WTDocument subtypes for the project contexts the organization is hosting.

In a Windchill PDMLink environment (standalone or combined with Windchill ProjectLink), type management activities can occur in the site and organization context.

Implications for Windchill users are as follows:

- Business objects created in the site context must be instances of modeled classes or instances of soft types defined in the site context.
- Business objects created in an organization context must be instances of modeled classes or instances of soft types defined in the site or given organization context.
- Business objects created in an application context must be instances of modeled classes or instances of soft types defined in the site or parent organization context.

## For Searchable Types

By default, the types that are searchable are maintained in the **All Applicable Object Types Search** preference. This preference can be updated at the site and organization levels using the Preference Administrator. For information on specifying types in preferences, see [Specifying Windchill Types](#).

If you add a type through the **Type Manager** tab, you can also update the preference to make the type searchable. For additional information, see the *Windchill Customizer's Guide*. You can also exclude types from the search pages; see [Excluding Types from Simple and Advanced Search Pages](#).

## Glossary Terms for the Type and Attribute Manager Utility

This section provides definitions for terms used throughout this chapter.

### **CAD parameter**

A CAD parameter is unique to a CAD application and can be defined by a CAD application user or by a CAD administrator at a site. A CAD parameter is similar to an attribute.

**Note:** In some CAD applications, parameters are referred to as properties.

### **explicit mapping**

Explicit mapping refers to a manual process the user performs using the **Attribute Mapping** tab on the **Type Manager** tab to map Windchill attributes to CAD parameters. When attributes on the EPM Document, EPM Reference Link, and EPM Uses Link types are mapped to CAD parameters using the **Attribute Mapping** tab, they are considered explicitly mapped.

### **file-based attribute**

By default, all Windchill attributes are file-based attributes unless configured specifically as a nonfile-based attribute. When a Windchill attribute is file-based, it is mapped to a CAD parameter and is either read-only or it can be edited or deleted.

**hard attribute**

A characteristic that identifies and describes a business object that has an associated field in the Java class of the business object's type, and a column in the type's database table. Hard attributes are defined by PTC during Windchill development and are typically, but not always, represented in a Rational Rose model. Additionally, hard attributes can be created by customizing Windchill.

**hard type**

A business object that has its own associated Java class and database table. Hard types are defined by PTC during Windchill development and are typically, but not always, represented in a Rational Rose model. Additionally, hard types can be created by customizing Windchill.

**implicit mapping**

Implicit mapping refers to the system matching the attribute and parameter automatically. No user intervention is necessary.

**modeled attribute**

Modeled attributes have an associated field in the Java class of the business object's type, a column in the type's database table, and are always represented in a Rational Rose model. Modeled attributes are a subset of hard attributes and are defined by PTC during Windchill development. Additionally, modeled attributes can be created by customizing Windchill.

**modeled type**

A business object that has its own associated Java class, a database table, and is always represented in a Rational Rose model. Modeled types are a subset of hard types. Every modeled type is a hard type, but not every hard type is a modeled type. Modeled types are defined by PTC during Windchill development. Additionally, modeled types can be created by customizing Windchill.

**nonfile-based attribute**

A nonfile-based attribute has full access in Windchill but is not visible in the CAD application. The value can be added, modified, or deleted through the Windchill user interface. The value will not be passed to the CAD parameter, therefore it will not be visible in the CAD application.

**soft attribute**

A characteristic that identifies and describes a business object that does not have an associated field in the Java class of the business object's type, nor does it have a column in the type's database table. Soft attributes are defined using the **Attribute Definition Manager** tab, and can be associated with the following:

- object types using the **Type Manager** tab
- object classification using the Classification Administrator installed with Windchill PartsLink
- a specific instance of an object, also known as IBAs, (instance-based attributes)

**Note:** Additional attributes can be loaded in the Windchill code.

**soft attribute set**

The set of all soft attributes that are defined using the **Attribute Definition Manager** tab and can be used by multiple Windchill objects and classes. The soft attribute set does not contain hard attributes or modeled attributes, and can be maintained independent of object classes and object types.

**Note:** *Soft Attribute Set* replaces *Global Attribute Set*.

**soft type**

A business object that uses its parent type's Java class and database table. Some soft types are defined by PTC during Windchill development and are included as part of your Windchill solution installation. However, most soft types are created and defined by the site or organization administrator using the **Type Manager** tab.





# 14

## Reporting

This chapter describes the various uses of reporting within Windchill, including the Windchill Business Reporting functionality.

<b>Topic</b>	<b>Page</b>
Overview .....	14-2
Windchill Business Reporting.....	14-6
Authoring Reports .....	14-9
Editing Reports.....	14-10
Searching for Reports.....	14-11
Monitoring Reports .....	14-11

## Overview

Within your Windchill solution, you may encounter reports in a number of locations including the following:

- The **Reports** page on most tabs.
- The **Related Reports** list when viewing product structures.
- The **Report Manager** utility.
- The **Change Monitor** page on **Product** and **Library** tabs. (Windchill PDMLink only)
- The **Auditing Administration** utilities.

The sections below further describe these locations, and the reporting functionality available from each.

## Reports Page

The **Reports** page located on most tabs is the primary location for reporting functionality within your Windchill solution. The **Reports** table contains the out-of-the-box predefined reports for each tab that were installed with your Windchill solution, and also any new reports created at your site.

If your site has enabled Windchill Business Reporting, then additional features are available for the predefined reports viewed from the **Reports** table, and may be available for new reports created by your site. For additional information, see [Windchill Business Reporting](#) below.

From the **Reports** table, you can perform a number of actions, depending on your permissions and the tab you are viewing, such as editing report information, deleting reports, and updating the contents of the **Reports** table to include new reports when they are available. Any user who can see the report in the **Reports** table can view the report.

For information on performing these actions, see the help available from the **Reports** table.

## Predefined Reports

The following predefined reports are installed out-the-box with the specified Windchill solutions, and displayed on the **Reports** table of appropriate tabs or other locations, as noted below. These predefined reports make use of the additional features available if Windchill Business Reporting is enabled at your site, but are still available with fewer features if Windchill Business Reporting is not enabled.

**Note:** All out-of-box predefined reports are displayed on the **Site** tab **Reports** table as well as the other locations listed below. These display locations can be edited by the site administrator.

For Windchill PDMLink:

Report Name	Tab or Display Location	Description
Problem Report Status Report	Product, Library, Change, Organization, and Site	Lists the state and status of all problem reports.
Change Request Status Report	Product, Library, Change, Organization, and Site	Lists the state and status of all change requests.
Change Notice Status Report	Product, Library, Change, Organization, and Site	Lists the state and status of all change notices
Variance Status Report	Product, Library, Change, Organization, and Site	Lists the state and status of all variances.
Multi-Level BOM Report	Product Structure Browser and Product Structure Explorer Related Reports list, Organization, and Site	Shows all attribute information for a complete product structure of multiple levels.

For Windchill ProjectLink

Report Name	Tab or Display Location	Description
Project Item Status Report	Home, Project, Organization, and Site	Lists the project items associated with all projects that meet the input criteria.
Program Item Status Report	Home, Program, Organization, and Site	Lists the program items associated with all programs that meet the input criteria.
Project Status Report	Home, Project, Organization, and Site	Lists status information for a list of projects that belong to an organization.
Program Status Report	Home, Program, Organization, and Site	Lists status information for a list of programs that belong to an organization.

**Note:** Predefined reports are not provided for Arbortext Content Manager or ProINTRALINK 9.0.

## Legacy Reports

On the **Home**, **Program**, **Project**, and **Organization** tab **Reports** pages, there are also tables containing legacy reports that have been available in past releases. These reports remain available for your use, but are not defined out-of-the-box to utilize the additional capabilities provided by the Windchill Business Reporting functionality. For more information, see the help available from these tables.

## Related Reports for Products

Windchill provides a number of related reports that can be used when viewing a product structure with the Product Structure Explorer and the Product Structure Browser. These reports include:

- Single Level BOM
- Indented BOM
- Multilevel Where Used
- Multilevel BOM Compare

**Note:** If Windchill Business Reporting is available at your site, then the Multi-Level BOM related report is launched in the Windchill Business Reporting viewer. Otherwise, it is displayed in the HTML viewer similar to the other related reports.

If you want part masters in the same level with the same quantity to be considered identical in the Multilevel BOM Compare report, you need to set the following property to true:

```
wt.part.identicalProductMastersAreSame
```

For more information on the related reports, see the *Windchill PDMLink User's Guide*.

## Report Manager

Windchill provides the ability to create custom reports using the Report Manager, and its report generation utility, Query Builder. You access this utility through the **Report Manager** link on the **Utilities** page of any application context tab. The Query Builder utility is designed to be used by those who have a working knowledge of the Windchill data object model. For more information on using Query Builder, see the help available from the Query Builder utility, and the *Windchill Customizer's Guide*.

## Change Monitor Reports

**Note:** The Change Monitor reports are available only from Windchill PDMLink.

Change Monitor reports are available from the **Change Monitor** page that is on the **Product**, **Library**, and **Change** tabs of Windchill PDMLink. The **All Special Reports** list on the **Change Monitor** page includes the out-of-the-box reports loaded when you installed Windchill PDMLink and all custom reports that are available. (These reports are not defined out-of-the-box to make use of the Windchill Business Reporting functionality.)

You use the Report Manager to create custom Change Monitor reports. The availability of a custom report is determined by where the report is created:

- Custom reports created from the **Site** tab are available to all products and libraries.
- Custom reports created from the **Organization** tab are available to all products and libraries whose owning organization is the organization that was active when the custom report was created.
- Custom reports created from the **Product** tab are only available to the product that was active when the custom report was created.
- Custom reports created from the **Library** tab are only available to the library that was active when the custom report was created.

To create a custom Change Monitor report:

1. Navigate to the tab under which you want to create the report.
2. Click **Utilities** and then open Report Manager by clicking its link on the page.
3. Click **New** to create a new report.
4. Set the location under which the report is stored as follows:
  - a. From the **File** menu, click **Set Location**.  
The **Set Location** window opens.
  - b. From the **Look in Cabinet** field, select System (if it is not already in the field)
  - c. Open the System folder by double clicking it in the field displayed under the **Look in Cabinet** field.

- d. Continue opening folders under the System folder until the path displayed in the **Location** field is:

/System/Reports/ChangeMonitor/Custom

- e. Click **OK**.

5. Create and save your custom report. For help with Report Manager activities, click the help icon in Report Manager.

To use the custom report, navigate to the applicable **Product** or **Library** tab and click the **Change Monitor** link. On the **Change Monitor** page that opens, the new report appears in the **All Special Reports** list. Click the custom report link from the list to execute the report.

## Auditing Administration Reports

The **Auditing Administration** utilities, available from the **Organization** and **Site** tab **Utilities** pages, offer logs and reports that provide information on system auditing activities, such as license usage, records of which user made what changes to a particular object, and other events. For more information, see the [Auditing](#) chapter.

## Windchill Business Reporting

Windchill Business Reporting uses a powerful third-party reporting application that works with your Windchill data to provide rich and flexible report options. If your site has enabled the Windchill Business Reporting functionality, additional reporting capabilities are available to you, including:

- scheduling reports to run over-night or during weekend hours when system demands are lower.
- scheduling reports to run at regular intervals, such as weekly, monthly, or quarterly.
- choosing from many report output formats, including HTML, PDF, Microsoft Excel, CSV, and XML.
- selecting from multiple report delivery options, such as:
  - printing the report.
  - e-mailing the report to specified users.
  - saving the report as a file on your system.
  - saving the report as a report view, allowing you to later run the report using the same input parameters, without having to specify them again.

If you are unsure whether your site has enabled Windchill Business Reporting, check with your system administrator. (ProINTRALINK 9.0 does not support Windchill Business Reporting.)

In addition to Windchill Business Reporting, your site may have installed additional optional modules:

- Windchill Business Report Author -- for more information, see [Authoring Reports](#).
- Windchill Business Report Monitor -- for more information, see [Monitoring Reports](#).

## Viewing Reports with Windchill Business Reporting

When a report defined for use with Windchill Business Reporting is viewed, it is automatically displayed in the Windchill Business Reporting viewer.

**Note:** The additional Windchill Business Reporting tools listed below are available only when you are logged into Windchill Business Reporting, and are not available from the viewer. To access these tools you must log into Windchill Business Reporting (see the following section).

## Accessing Windchill Business Reporting

To access Windchill Business Reporting, navigate to the following URL:

`<machine>:<port>/cognos8`

Where `<machine>` is the machine on which Windchill Business Reporting gateway was installed, and `<port>` is the Windchill Business Reporting gateway machine's web server port (if you are using the default port value of 80, then you do not need to specify the port in the URL). If your site has been configured to use SSL, you need to preface the URL with HTTPS. If you are uncertain of what values to use for this URL, see your system administrator.

Out-of-the-box, all users in the Administrative LDAP can log in to Windchill Business Reporting. Your site may have configured Windchill Business Reporting to authenticate against an Enterprise LDAP, as well. For more information, see [Working with LDAP Directory Services](#) in the Principals (Users, Groups, and Organizations) chapter.



**Caution:** Out-of-the-box, Windchill Business Reporting does not provide the same sort of access controls found within your Windchill solution. Any user who can log in to Windchill Business Reporting has full access. This means that any user who logs into Windchill Business Reporting may be able to view information within Windchill Business Reporting that they do not have permissions to view within your Windchill solution. Check with your system administrator to see whether additional access control for Windchill Business Reporting has been established. For information on setting up access control within Windchill Business Reporting, see the *Windchill Customizer's Guide*.

Once you are logged in, you can utilize the tools available within Windchill Business Reporting. These tools include:

- **Report Studio** - Available with the optional Windchill Business Report Author module. This tool allows you to create business reports using drag-and-drop authoring tools.

**Note:** Report Studio is only available if you are using the Internet Explorer browser.

- **Event Studio** - Available with the optional Windchill Business Report Monitor module. This tool allows you to monitor business events and define actions to be taken when specified thresholds are reached.

For further information about these tools, see the documentation available from the Windchill Business Reporting documentation page, as described in [Accessing Documentation within Windchill Business Reporting](#) below.

## Configuring Windchill Business Reporting

To further configure Windchill Business Reporting for your site, you can access the configuration utility at the following location:

- for Windows: `<WBR_Home>\bin\cogconfigw.exe`
- for UNIX: `<WBR_Home>/bin/cogconfig.sh`

Where `<WBR_Home>` is the machine on which Windchill Business Reporting was installed. There may also be a shortcut to the configuration utility on the machine where Windchill Business Reporting was installed, depending on the selections made during installation.

For further information, refer to the documentation available from the configuration utility.

## Accessing Documentation within Windchill Business Reporting

In addition to help available throughout Windchill Business Reporting, guides documenting the additional reporting capabilities and configuration options are available from the documentation page within Windchill Business Reporting.

This page can be accessed in one of two ways from the machine on which Windchill Business Reporting (host component or gateway server) was installed:

- If shortcuts were created during your installation, select the **Cognos 8 Documentation** shortcut.
- Navigate to the following location and open the file:

`<WBR_Home>\webcontent\documentation\c8_mtoc.html`

where `<WBR_Home>` is the installed location of Windchill Business Reporting.

If you do not have access to the machine or machines where Windchill Business Reporting was installed, contact your system administrator.




# Authoring Reports

Reports can be authored and displayed in the **Reports** table using the following tools:

- [Query Builder](#)
- [Info\\*Engine Tasks](#)
- [Windchill Business Report Author](#)

For all reports, regardless of how they are authored, the **Ready for Use** check box must be selected before the report is visible to all users in the **Reports** table. Until then, the report is visible only to users with modify permission to the report (by default, site administrators).

To select the **Ready for Use** check box, edit the report by selecting the edit action from the **Reports** table or from the information page for the report.

Clicking the update table icon  on the **Reports** table causes any reports that are defined, but not currently displayed in the **Reports** table to be added and displayed once the table refreshes. This action updates the **Reports** table in all contexts for which you have create permissions; if you are a site administrator, every **Reports** table is updated.

## Query Builder

Queries written and saved in the Query Builder utility are saved as report templates, which can then be run from the Report Manager. You can also select the **Expose as Report** flag in the **Report Properties** window, to expose the report template as a report object in Windchill. The next time the **Reports** table is updated, this new report object is included in the appropriate location.

Query Builder can also be used to create data sources for reports authored for use by Windchill Business Reporting.

For additional information, including information on custom input pages for reports, see the help available from the Query Builder, and the *Windchill Customizer's Guide*.

## Info\*Engine Tasks

Info\*Engine tasks can be used to create data sources for reports authored for use by Windchill Business Reporting. For information on using Info\*Engine tasks in this manner, see the *Windchill Customizer's Guide*.

## Windchill Business Report Author

If your site has the optional Windchill Business Report Author module, then you can use the Report Studio tool within Windchill Business Reporting to author reports which take advantage of its more robust reporting capabilities, including:

- Drag-and-drop report authoring.
- Incorporate text, charts, and images within one report.
- Create reports using a variety of charts and graphs, including crosstabs, bar and 3D bar, pie, line, gauge, funnel, scatter, dot density, waterfall, and others.

Data sources for these reports are created from your Windchill data using Query Builder or Info\*Engine tasks as noted in the previous sections.

**Note:** The Report Studio tool in Windchill Business Reporting works only with Internet Explorer.

When authoring reports within Windchill Business Reporting, it is necessary that you save the reports within a folder structure that mirrors the context hierarchy of your Windchill solution. For detailed information on authoring reports for use with Windchill Business Reporting, see the *Windchill Customizer's Guide*.

## Editing Reports

Users with modify permissions to reports in a particular context can edit Windchill reports which reside in that context. Out-of-the-box, only site administrators can edit Windchill reports, as the predefined reports all reside in the site context.

Edit reports by selecting the edit action from the **Reports** table, or from the information page for the report. Editing allows you to change certain report information, including the display locations, name, description, ready for use status, and other information.



**Note:** If the report is defined to display in Windchill Business Reporting, then the report name in Windchill must match the report name defined in the Windchill Business Reporting system. Otherwise the report will not run in Windchill Business Reporting, but will run in the default HTML Windchill report viewer.

For more information on editing reports, see the help available from the **Edit Report** window.

To allow additional users to edit reports in other contexts, grant them modify permissions to the report object type in the desired context using the Policy Administrator. For more information, see [Administering Domains and Policies](#) in the Contexts chapter.

For information on how to edit reports within Windchill Business Reporting, see the documentation available from the Windchill Business Reporting documentation page. For more information, see [Accessing Documentation within Windchill Business Reporting](#), above.

## Searching for Reports

Reports can be searched for using the Windchill search functionality. Both report objects and report templates can be returned in the search results. As both the report object and the report template may have the same name, these may appear to be duplicate search results, but they are not. On the **Advanced Search** page search results, the icons for report templates  and report objects  are displayed for clarification.

**Note:** When searching for reports by name, only the English name is searched upon. Any localized name available is used for display only, and is not a searchable attribute.

For information on searching for reports within Windchill Business Reporting, see the documentation available from the Windchill Business Reporting documentation page. For more information, see [Accessing Documentation within Windchill Business Reporting](#), above.

## Monitoring Reports

If your site has the optional Windchill Business Report Monitor module, you can use the Event Studio tool to:

- Monitor report parameters.
- Define actions to be executed when certain report parameter limits are reached.
- Monitor report parameters against specified thresholds, and automatically notify users when a threshold has been crossed.

For more information, see the documentation available from the Windchill Business Reporting documentation page. For more information, see [Accessing Documentation within Windchill Business Reporting](#), above.



# 15

## Life Cycles

This chapter provides information about life cycles and how to define them using the Life Cycle Administrator.

Topic	Page
Overview .....	15-2
The Life Cycle Model .....	15-2
Basic and Advanced Life Cycles.....	15-4
Managing Life Cycle Processes .....	15-5
Out-of-the-box Life Cycle Templates .....	15-13
About Life Cycle Iteration .....	15-17
Creating a Life Cycle Template .....	15-42
Overriding and Reassigning Life Cycle and Team Templates.....	15-42
Defining Additional Life Cycle States .....	15-43
Access Control for Life Cycle Administration.....	15-44
Best Practices .....	15-44

## Overview

Business information and business objects generally become more mature and reliable over time. *Life cycles* define the way in which these business objects mature, providing a model for a product's commercialization process.

A Windchill user in the role of life cycle administrator can create a variety of life cycles. These life cycles define the states and transitions associated with various business objects and the available transitions between those states.

## Accessing the Life Cycle Administrator

You can access the Life Cycle Administrator utility by clicking the **Life Cycle Administrator** link on the **Utilities** pages that are under the **Site**, **Organization**, **Product**, and **Library** tabs.

The **Life Cycle Administrator** link from the **Site** tab provides you, the site administrator, with unrestricted access to all life cycles. From the **Site** tab, you see only life cycle templates that have been created in the site context and those provided out-of-the-box.

From the **Organization** tab, you see life cycle templates created in the site context and those that are in the active organization context when the Life Cycle Administrator was launched.

The link from the **Product** and **Library** tabs provide access to the life cycle templates created in the site and the organization context and those that are in the active context (Product or Library) when the Life Cycle Administrator is launched.

## The Life Cycle Model

A life cycle is an automated, graphical model, employing phases and gates, used to manage business objects as they progress from conceptualization through obsolescence.

While an object is in a specific life cycle phase, certain business rules apply, such as access control rules defined for that phase.

When created, an object modeled to be life cycle-managed enters a life cycle phase, where it is assigned an initial state, and is then associated with the initial phase of its life cycle.

## Windchill Solutions

Object initialization rules are used to designate to which life cycle an object type is assigned when a user creates a new object of that type. Workflows associated with an object, life cycle-manage the maturing processes from state to state through workflow activity templates. When new versions of an object are created through the Revise action, the team and the life cycle is assigned to the object following the same rules as when the object was created, and the life cycle state is set to the first state of the life cycle.

Depending on several business environment considerations (impact of change, development team, maturity of object, type of object, and so forth), it may be important to tailor the life cycle to include certain business operations and to define how these operations are executed. Transition rules tailor how Change Management, Revise, Promotion, and Set State operations should behave and which workflow processes you should use to manage these processes.

## Life Cycle States

The sophistication of business processes to manage an object depends upon the maturity of the object. State is an enterprise object, and its meaning is applied regardless of the life cycle by which a given object was processed. For example, if an access control rule applies to a Requirements object in the Under Review state, the rule is applicable to all Requirements objects in that state, even if they arrived at the state through different life cycles. However, each phase of a life cycle must be associated with a life cycle state chosen from among all states defined in the system.

**Note:** If the domain policy denies access to a user and if an ad hoc policy grants the user access, then the ad-hoc policy is used, that is, the user is granted access. Ad hoc policies overrule and can provide access to a user where the domain policy denied it.

As a life cycle administrator, you can create a variety of life cycles. These life cycles, which are stored in the System folder, define the phases and gates associated with various business objects. For the life cycle of each object, you can define the transitions through which the object must move, and the behavior associated with the object while it is in a specific state.

Windchill provides a Default life cycle, with many predefined states, such as In Work, Under Review, and Released.

Before you begin creating life cycles, you should understand life cycle iteration and life cycle roles, as described in this guide.

## Basic and Advanced Life Cycles

Windchill supports two types of life cycles: basic and advanced. There are major differences between the basic and advanced life cycle that are visible to the user. The main differences are in the setup of the team definitions for an object, in the workflow processes for managed objects, and in the system performance when creating and revising objects. Using basic life cycles provides improved performance for core creation and revise operations.

You can designate whether the life cycle includes certain aspects of a life cycle definition. This is especially useful for team definitions and the use of workflow processes in Windchill.

When you create a basic life cycle, only the life cycle template, state, and transitions are defined. When you create an advanced life cycle, extra capabilities can be defined including; phase information, team, workflow processes, and promotion criteria.

**Note:** PTC recommends that you use the basic life cycle for improved performance and scalability.

Any particular context could have a mix of basic and advanced life cycles associated with objects. Ideally, an object is associated with an advanced life cycle, just in time when needed, such as when you route an object in Windchill ProjectLink.

Any time you need to manage access via life cycle team roles and to drive an object through its life cycle states by a workflow process, use an advanced life cycle.

### Basic Life Cycles

When you create a basic life cycle, you define the following:

- The life cycle properties, including name, location, an optional description, the object classes to which the life cycle applies, and whether the life cycle is enabled.
- Phases and gates that define the life cycle.
- Transitions to be defined from the current (selected) state to any defined state within the life cycle template.

The basic life cycle is designed to use the context team defined in any application context. A basic life cycle does not include workflow process definitions and relies on the workflow process definitions with change management, routing, or promotion processes. Use this type of life cycle for most Windchill objects because the basic life cycle provides improved performance during the creation and revision of objects.




When you create an object, such as a part or document, and select a team template and a basic life cycle; the selection of the team has no effect. Basic life cycles do not utilize teams or workflows.

## Advanced Life Cycles

When you create an advanced life cycle, you define the following:

- The life cycle properties, including name, location, an optional description, the object classes to which the life cycle applies, and whether the life cycle is enabled.
- Phases and gates that define the life cycle.
- Roles, such as Reviewer or Workflow Assignee, for each life cycle phase. These roles can be mapped directly to users or user-defined groups, but are most often mapped to team roles or actors.
- Access permissions for the role players associated with each life cycle phase.
- Workflow processes to be associated with each phase and gate.
- Promotion criteria to be satisfied at each gate.
- Transitions to be defined from the current (selected) state to any defined state within the life cycle template.

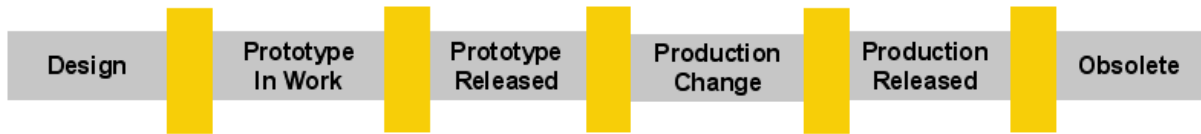
**Note:** An advanced life cycle is denoted by  on the Life Cycle Administrator window.

The advanced life cycle type provides the ability for a separate team to be assigned for instances of this object type and to include workflow process definitions within the life cycle states to manage the maturing of the object type. Use this type for managing objects that have company-specific business processes, special access control requirements, or additional flexibility in managing these types of objects.

## Managing Life Cycle Processes

In Windchill, the life cycle is the core capability to manage an object's maturity (referred to as a life cycle state) as well as access control policies to manage or view the object for a set of roles in the system. Each object type in a Windchill system (such as a part, specification, drawing, and so forth) can have unique sets of life cycle states and access control policies. Object initiation rules are used to designate which life cycle an object is assigned when a user creates a new object of that type. Workflows (associated with an object) life cycle-manage the maturing processes from state to state through workflow activity templates.

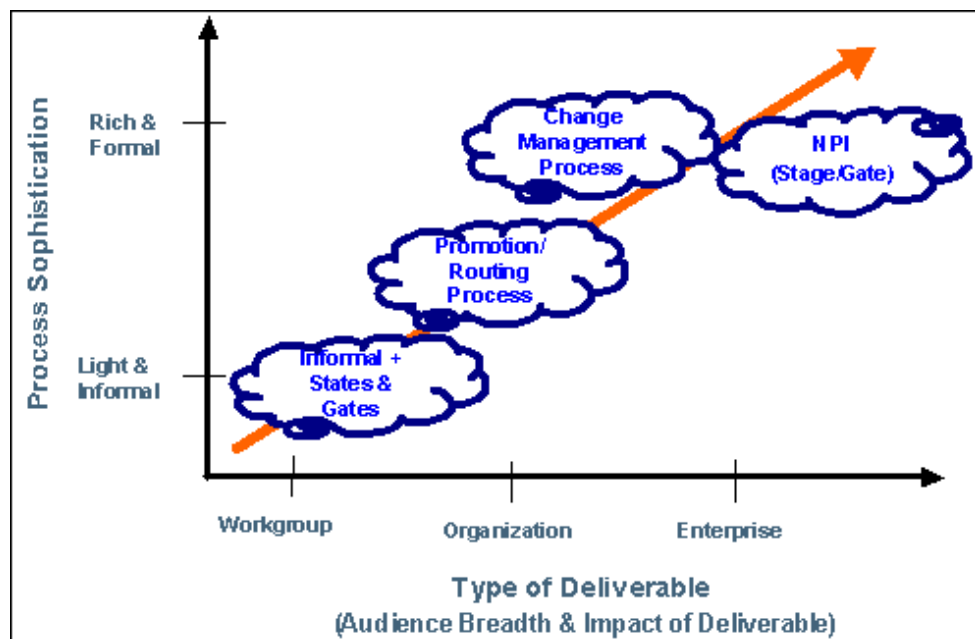
By default, when new versions of an object are created through the Revise action, the life cycle state is set to the first state of the life cycle. A life cycle administrator can specify the target state of the Revise action.



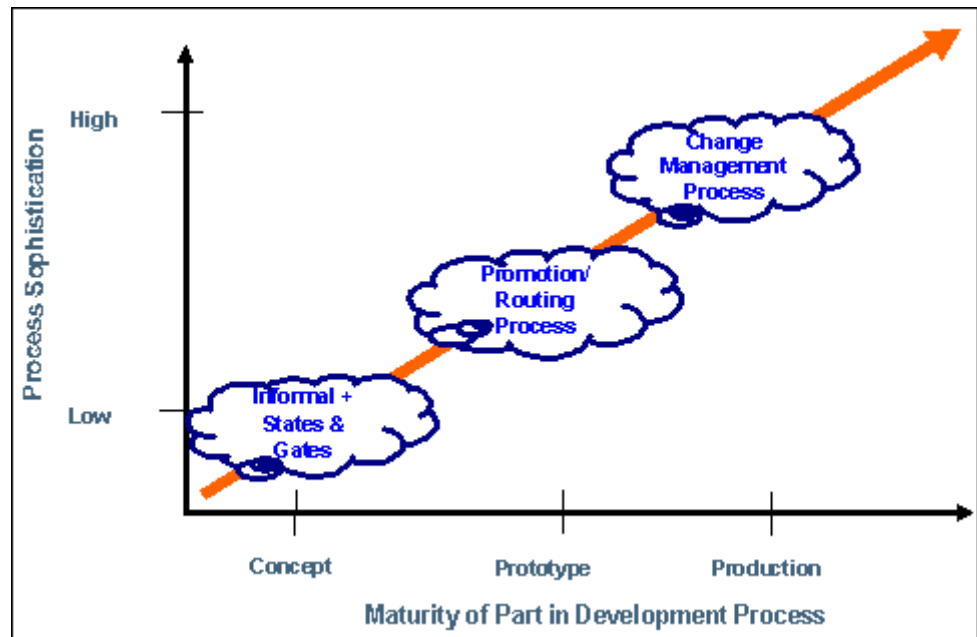
Depending on several business environment considerations (impact of change, development team, maturity of object, type of object, and so forth), it may be important to tailor the life cycle to include certain business operations and to define how these operations are executed.

For example, a development team in the automotive division of a company can have different review processes for releasing a design specification than the industrial products division. The automotive division may require a prototype to be created for a new product before the first production versions of the product are released. Formal change processes are required to update these prototype designs. After a part has been initially released to production, new versions of the design are designated as Production Change, and they must undergo a formal change process.

The following figure is an example of the variety of business processes that can be used with a Windchill system to mature different types of product development deliverables. The development team processes for the review and release of a design specification can differ from those required for the release of a new configuration of the product.



The next figure illustrates a case when a combination of the business processes is required for some product development deliverable objects, such as parts, as they move from a concept, through prototyping, and into production.



Windchill PDMLink supports the following business processes that you can use to manage the life cycle of a part, document, or CAD document. You can use transition rules to control when in the life cycle this operation is available. For example:

- Set State - Enables you to informally set the life cycle state of an object.
- Promote - Enables you to set the state of one or more objects to a new life cycle state as part of a review process.
- Change - Enables you to execute a change order for a product development object.
- Revise - Enables you to create a new version of the object.

For more information on transition rules, see [Transition Rules](#).

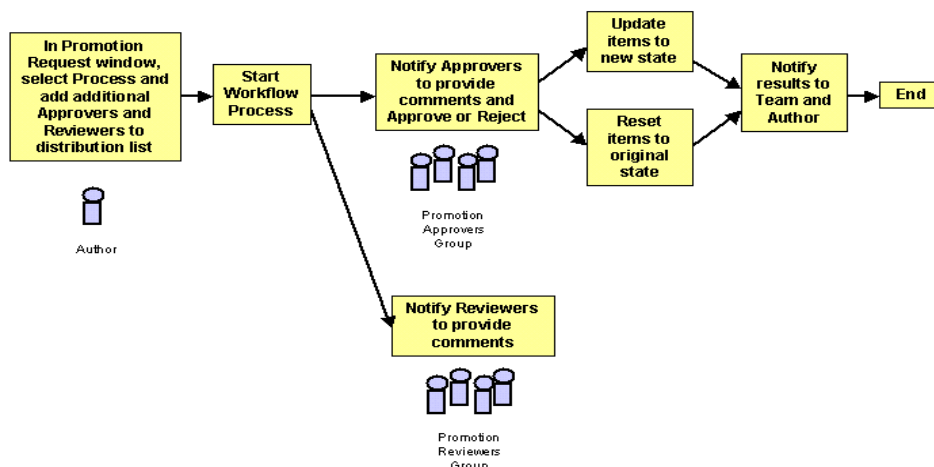
## Promotion Process

For the promotion process, you can use two standard review and approve workflow processes. With the enhancements to life cycle management, these two processes can be used to manage the development of CAD documents, parts, and documents.

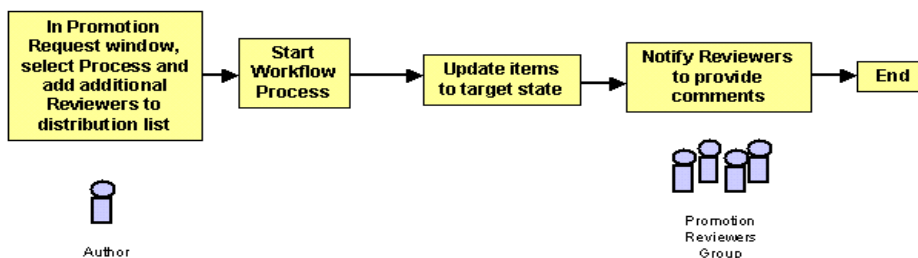
The promotion request window guides a user requesting a promotion of one or more CAD documents, parts, and documents through the promotion process. The user can add dependents of parts and CAD documents to the request through this process. All objects requested for promotion must be set to the same target life cycle state. If the objects requested for promotion do not share any life cycle states, no promotion can be completed.

Two standard workflow processes for promotion are as follows:

- **Promotion Request Approval Process** - A process to notify a set of Approvers to approve or reject the Promotion Request and invite a list of Reviewers to provide comments to the Promotion Request. If all Approvers approve the request, the objects in the promotion request are updated to the target life cycle state. Otherwise, these objects are returned to their original life cycle state.



- **Promotion Request Review Process** - A process to automatically set the state of promotion request objects to the target life cycle state and invite a list of Reviewers to provide comments on the promotion request.



A list of Approvers who must approve the promotion request is defined by the Promotion Approvers system group for a product or library. A list of Reviewers, who are invited to provide comments to the objects under review, but do not approve the request, is defined by the Promotion Reviewers system group for that product or library. The user submitting the promotion request can also add (in the last step of the **Promotion Request** window) Approvers or Reviewers for each specific review through a list of users and user-defined groups for the product and library.

When a promotion request is created for one or more CAD documents, parts, and documents, the list of specific iterations of these objects is maintained in the baseline for this promotion request. Additionally, for a set of CAD documents on a promotion request, a set of viewables is generated for this baseline set, and it can be viewed directly from the promotion request task or from the promotion request form.

**Note:** If you want to disable viewables from being generated in the promotion process, set the following property to false; `publish.promote.enabled=false`.

## Out-of-the-Box Workflow Processes for Promote Transition

Windchill PDMLink uses two out-of-the-box workflows with the Promote transition; **Promotion Request Approval Process** and **Promotion Request Review Process**. See the [Promotion Process](#) section for a brief description of these workflows.

Out-of-the-box, a preference is loaded into the site context that specifies the two default transition processes; initially, this is set to two processes (listed above). These two workflow processes are presented for every promotion request; however, preferences may be configured to provide zero or more workflows to reflect business practices.

The **Default Promotion Processes** preference sets the default workflow processes that are used by the promotion process. The value is set using a list of the workflow process template names delimited by the character "~" (tilde).

You can use the Preference Manager on the Site tab to manage preferences. The **Preference Manager** is found on the **Utilities** page. For more information, see the [Administering Preferences](#) section in the [Contexts](#) chapter.

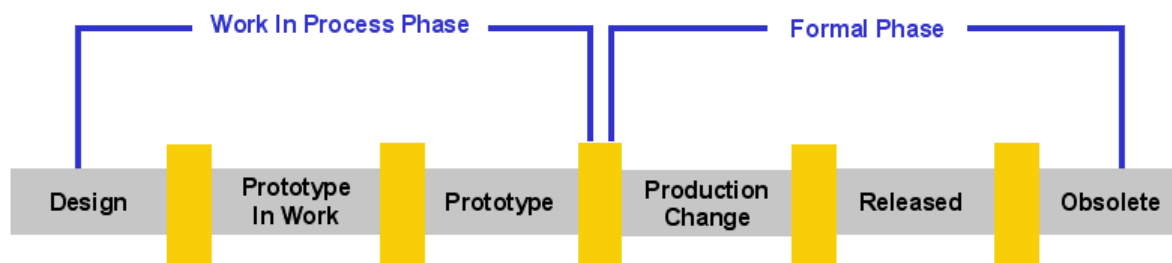
## Transition Rules

Life cycle transition rules provide capabilities for defining some behaviors of an object. These rules apply to the following actions in Windchill PDMLink:

The **Transitions** tab allows you to define transition paths from the selected state to any other state within the life cycle template.

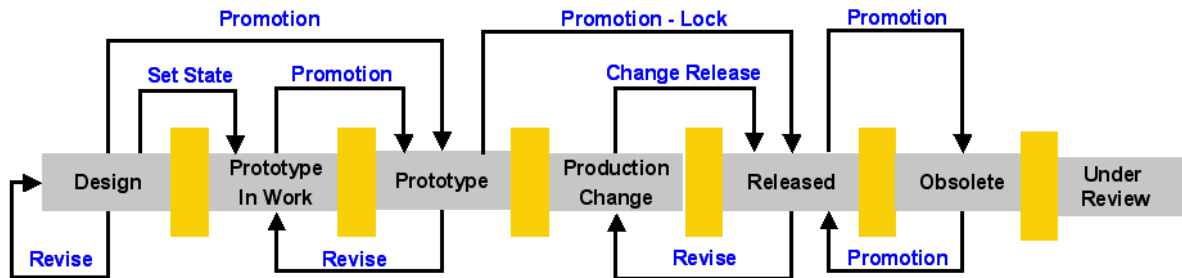
- **Change** - Enables you to execute a change order for a product development object. In particular, the Change transition identifies the state to which an object is released from a given state.
- **Lock** - Enables you to identify a temporary state in which an object resides only while the promotion process is being executed: at the end of the process, the object either promoted to a specified state (see description of Promote transition) or returned to its original state. Typically, the state identified by a Lock transition specifies restricted access for modification. Lock transitions are always optional.
- **Promote** - Enables you to set the state of one or more objects to a new life cycle state as part of a review process. Transition rules can be used to control when in the life cycle this operation is available.
- **Revise** - Enables you to specify the initial life cycle state of a new version of the object. Transition rules can be set up to control the life cycle state of the new version based on the state of the version being revised. Additionally, rules can be set up for a life cycle state to use a new version sequence when the Revise action occurs.
- **Set State** - Enables you to informally set the life cycle state of an object. Set state is an extension of the administrative set state function to users without administrative privilege. The user must have create permission for the promotion request in the context in question.

Using a set of transition rules, you can define appropriate sets of business processes for the stage of development for an object. For example, the processes for developing a part can be informal in the early stages of development, but will require more formal processes if the part is introduced into production.



Using transition rules, you can set up a variety of business processes to be available for each type of object in a Windchill PDMLink system.

The following diagram is an example of how these rules can be applied to a life cycle.



Some business practices require more formal processes, such as review and approval cycles. The Promotion and Change transition rules have associated workflows for that purpose. By default, the promotion transition rule has two workflow processes defined. Depending on your business practices, zero or more workflow processes can be configured for use with a specific promotion process.

For some promotion review processes, it is important to freeze or lock the promotion objects, so that no changes are made during the review process. For the example life cycle above, the lock transition can be added to set the objects to the Under Review state during a promotion process.

For change management processes, the new life cycle state for approved change notice is now defined by the Change transition and is no longer fixed to one specific life cycle state.

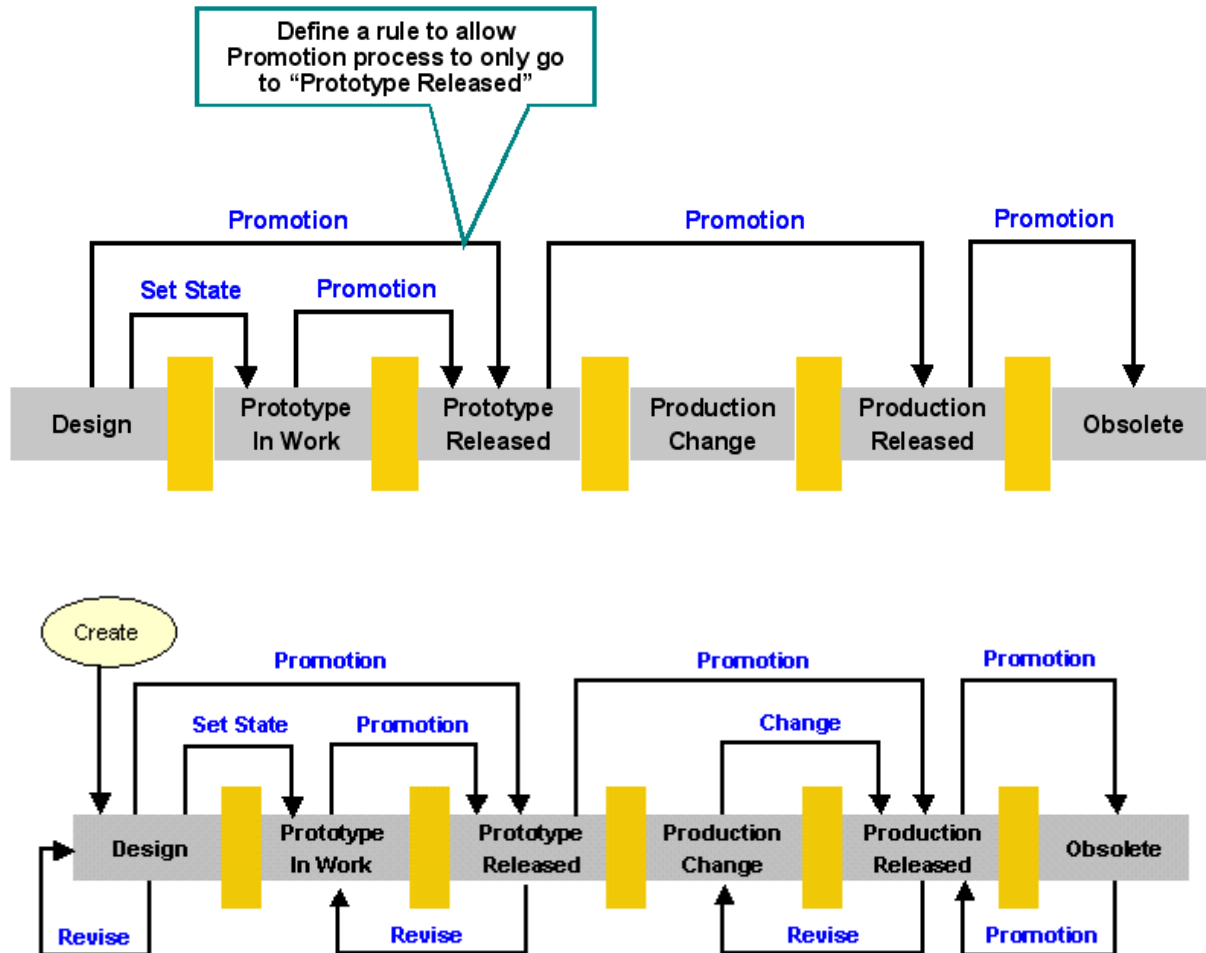
These transition rules are defined for each life cycle.

Transition rules may also identify whether the objects are locked during Promotion reviews. The object resides temporarily in the locked state while the promotion process executes. If a lock transition is defined for a particular state and an object using the life cycle is designated for promotion on a promotion request, the workflow of the promotion request sets the state of the object into the state identified with the lock transition.

Suppose your life cycle model looked like this:



You determine that you want to establish transition rules:



## Associating Life Cycles with Object Types

You can associate object types with life cycles.

**Note:** If you create a type and associate it with a default routing life cycle, there may be problems when users create documents using templates created with the type. The default routing life cycles start automatically, so when the user tries to create from a template using one of these life cycles and selects the **Check Out and Download** action, the checkout and set state (life cycle) gets caught in a race condition.



This can cause a couple of outcomes:

- The set state fails; the user has a checked-out document that is In Work, and the life cycle does not happen.
- The set state succeeds and the user has a checked-out, newly-created document that has been immediately routed for review.

PTC recommends you create your own routing life cycle that would contain a specific Submit task later in the process, not an automatic Submit upon creation.

## Out-of-the-box Life Cycle Templates

The following out-of-the-box life cycle templates are identified as basic or advanced.

Basic Life Cycle Templates	Advanced Life Cycle Templates
Action Item	Approval Routing
Approval	Change Activity Life Cycle
Basic	Change Notice Life Cycle
Library Development	Change Proposal Life Cycle
Notify	Change Request Life Cycle
One Phase Development	Default
Release	Notify Routing
Review	Part Request Life Cycle
Two Phase Development	Problem Report Life Cycle
	Promotion Request Life Cycle
	Release Routing
	Review Routing
	Two Level Approval Routing
	Variance Life Cycle

## Windchill PDMLink

The following Windchill PDMLink out-of-the-box life cycle templates are basic life cycles:

- Basic
- Library Development



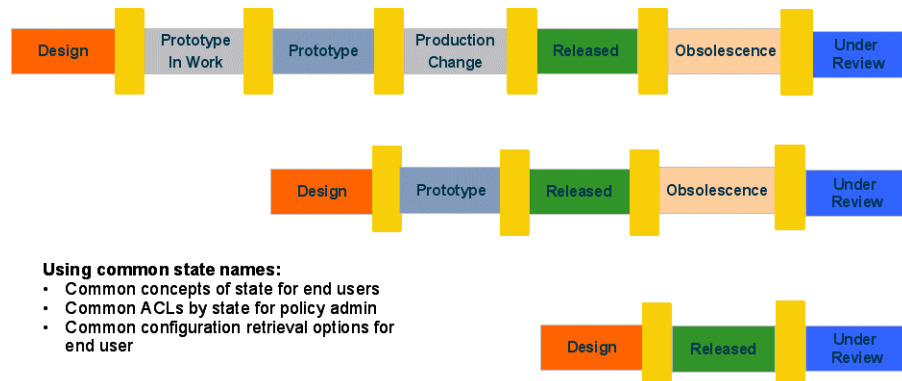
- One Phase Development



- Two Phase Development



The following figure shows the states that the three basic life cycle templates have in common.



The following Windchill PDMLink life cycle templates are advanced:

- Change Activity Life Cycle
- Change Notice Life Cycle
- Change Proposal Life Cycle
- Change Request Life Cycle
- Default
- Part Request
- Problem Report
- Promotion Request
- Variance Life Cycle

## Using the Product Design Template

The out-of-the-box Product Design template that is available in Windchill PDMLink establishes life cycle templates that are complete with transition rules and workflow processes for business objects; it uses the One Phase Development and Two Phase Development templates. Remember, the Product Design template should be used as an example. Your site should determine how to use transition rules and workflow processes, and set up default life cycles for the site, not just at a particular product level. For additional details on the Product Design template, see [Product Design Template](#) in the Products and Libraries chapter.

### Access Control for Parts Established Through the Product Design Template

The Product Design template also establishes access control policy rules for business objects. The following table shows the access control lists (ACLs) for parts as established in the Product Design template.

<b>System Group</b>	<b>Design</b>	<b>Prototype In Work</b>	<b>Prototype</b>	<b>Production Change</b>	<b>Released</b>	<b>Obsolete</b>	<b>Under Review</b>
Confirmed Members	No Access	No Access	Read	No Access	Read	Read	Read
Marketing	No Access	Read	Read	Read	Read	Read	Read
Procurement Engineer	No Access	Read	Read	Read	Read	Read	Read
Quality Engineer	No Access	Read	Read	Read	Read	Read	Read

System Group	Design	Prototype In Work	Prototype	Production Change	Released	Obsolete	Under Review
Designer	Read Create Modify Delete Revise NVV <sup>1</sup>	Read Modify Delete Revise NVV	Read Modify Delete Revise NVV	Read Modify Delete Revise NVV	Read Revise NVV	Read	Read
Manufacturing Engineer	Read	Read	Read	Read	Read	Read	Read
Design Team Leader	Read Create Modify Delete Revise NVV	Read Modify Delete Revise NVV	Read Modify Delete Revise NVV	Read Modify Delete Revise NVV	Read Revise NVV	Read	Read
Product Manager	Full	Full	Full	Full	Full	Full	Full

1. NVV is New View Version.

For additional details about the access control settings in the Product Design template, see [Product Design Template](#) in the Products and Libraries chapter.

## Windchill ProjectLink

The following out-of-the-box Windchill ProjectLink life cycle templates are basic:

- Action Item
- Approval
- Basic
- Notify
- Release
- Review

The following Windchill ProjectLink life cycle templates are advanced:

- Approval Routing
- Default
- Notify Routing
- Release Routing
- Review Routing
- Two Level Approval Routing

## About Life Cycle Iteration

Working with life cycles is an iterative process. Like version-controlled objects, *iterated objects* are checked in to and out of shared locations; however, unlike version-controlled objects, they cannot be revised. Instead, any change to an object creates a new and separate iteration when it is checked in. Earlier iterations, which may still be in use, are unchanged and unaffected by the new iteration. Only the latest iteration is available for new uses.

To make changes to a life cycle template, you must check out a copy. (Click **Edit** on the **Life Cycle Administrator** page to check out a copy of the selected life cycle.) While it is checked out, no one else can check out a copy, but the original can be viewed or selected to manage an object. When you have completed changes to the checked-out copy, you must save it and check it in, so it is available to others. It then becomes the latest iteration. Objects that are being managed by an earlier iteration continue to be managed by that iteration. They are not affected by the newer iteration.

## Viewing Iteration History

You can view the iteration history of a life cycle from the **Life Cycle Administrator** utility.

Select a life cycle template on the **Life Cycle Administrator** navigation panel and then click **Iteration History**. A list of all the life cycle iterations appear with the date and time of last modification, and the name of the modifier. Select any iteration, and click **View** to view the life cycle.

## Creating a Life Cycle Template

The **Life Cycle Administrator** utility displays a list of existing life cycle templates and their locations. Using the buttons on this page you can create, edit, view, and delete life cycles. You can import and export life cycles among other functions.

You can sort the templates list in ascending and descending order by clicking on the column header.

From the **Life Cycle Administrator** utility, you can create a new life cycle or edit an existing one:

- Click **New** to open the **New Life Cycle** window.
- Click **Edit** to open the **Edit Life Cycle** window.

Editing uses essentially the same procedures as creating a new template, but you modify information, rather than creating a new template. When you edit a life cycle, it is automatically checked out.

You must have the necessary access permissions to create or edit a life cycle. If you do not have the required permissions, the **New** and **Edit** buttons are enabled, but you get an error message when you try the operation. For the necessary access control permissions, see the [Access Control for Life Cycle Administration](#) section.

Use the toolbar buttons on the **Life Cycle Administrator** page to create a graphical representation of the life cycle you are defining. For a description of the buttons and their functions, see the online help.

When you create a life cycle, you define the following:

- The properties of the life cycle, including name, location, the object types to which the life cycle applies, and whether the life cycle is enabled. (See [Life Cycle Properties](#).)
- Phases and gates defining the life cycle. (See [Defining Life Cycle Phases and Gates](#).)
- Transition rules to help determine the path from the selected state to any other state within the life cycle template. (See [Defining Transitions](#).)

In addition, the following are defined for advanced life cycles only:

- Roles, such as Submitter or Promoter, for each life cycle phase. These roles are mapped to users, user-defined groups, organizations, actors, or other roles. (See [Selecting Life Cycle Roles](#).)
- Access permissions for the roles associated with each life cycle phase. (See [Defining Life Cycle Access Control Rules](#).)
- Workflow processes to be associated with each phase and gate. (See [Associating a Workflow Process with Phases and Gates](#).)
- Promotion criteria to help determine whether or not an object is ready to move to the next phase in its life cycle. (See [Defining Promotion Criteria](#).)

## Life Cycle Properties

You can view or edit the properties of a life cycle template by selecting a life cycle template in the **Life Cycle Administrator** window and then clicking either **View** or **Edit**.

The upper half of the screen displays a graphical view of the life cycle's phases and gates, as shown in the graphic below. The yellow bars represent a gate and the labeled grey sections (In Work, Completed, and Cancelled) represent a specific phase in the life cycle. The example below shows the phases, In Work, Completed, and Cancelled.



The lower half of the screen contains the **Properties - Life Cycle** panel which displays the properties of the life cycle. To view the properties of a specific phase, click the icon (the grey section) of the phase to select it. To return to the **Properties - Life Cycle** panel, click the background of the life cycle.

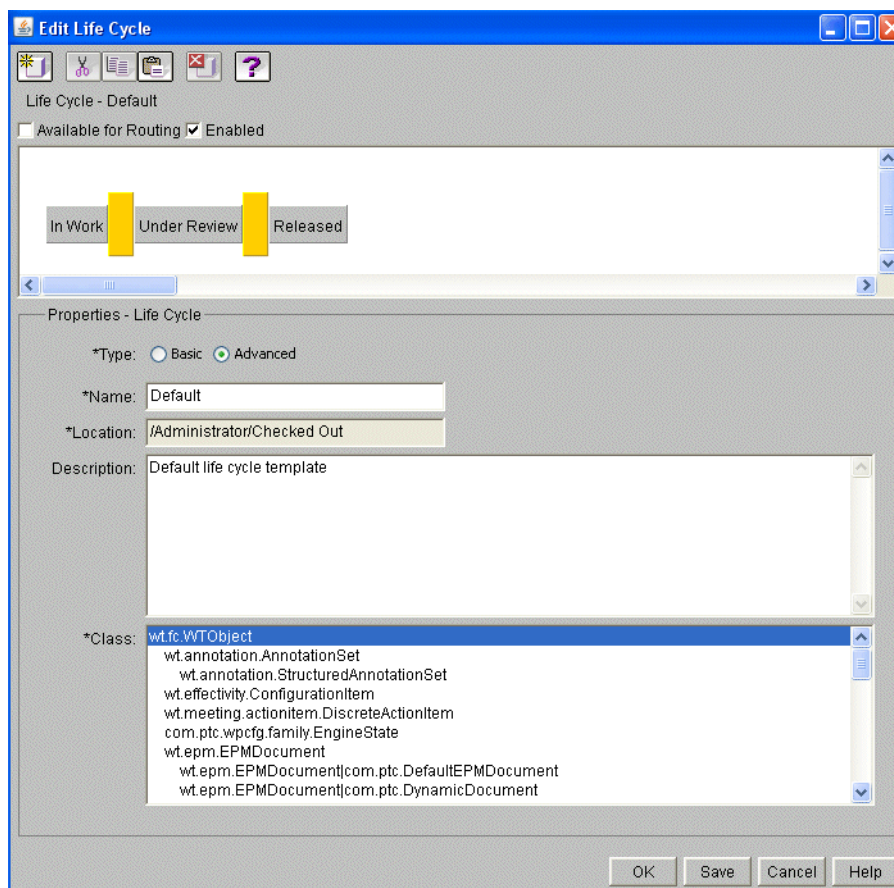
In the **New** (or **Edit**) **Life Cycle** window, the **Properties** panel (the lower part of the window) displays the properties of the life cycle itself. When you select a phase icon, this panel reflects the properties of that phase.

A life cycle has the following properties:

Life Cycle Property	Description
Type	Specifies whether the life cycle is basic or advanced. The basic life cycle contains only the life cycle template, state, and maturity history. The advanced life cycle includes extra capabilities of phase information, team, related objects, and so forth.
Name	Specifies the name of the life cycle. Life cycle names must be unique. This is a required property.  If you enter a name already in use, an error message appears. When you edit a life cycle, you can change the name of the life cycle.
Location	Specifies the cabinet and folder in which this life cycle is stored. The System cabinet is the default location.  <b>Note:</b> Windchill automatically checks in the template to the System folder for the context in which it was created.
Description	Specifies optional text describing this life cycle.
Class	Specifies the object type to which this life cycle applies. This is a required property.

Life Cycle Property	Description
Enabled	Indicates whether the life cycle-managed object is enabled or disabled. Select the check box to enable a life cycle-managed object when the life cycle is created. Typically, you clear the <b>Enabled</b> check box only when you plan to delete the life cycle in the future, when it is no longer being used by a life cycle-managed object.
Available for Routing	Used only by Windchill ProjectLink, this property specifies the life cycle templates can be used for routings.  <b>Note:</b> Only life cycle templates that have one workflow associated to the first phase of the life cycle support routing.

The following figure displays the **Edit Life Cycle** window, with the Default life cycle and its properties displayed:






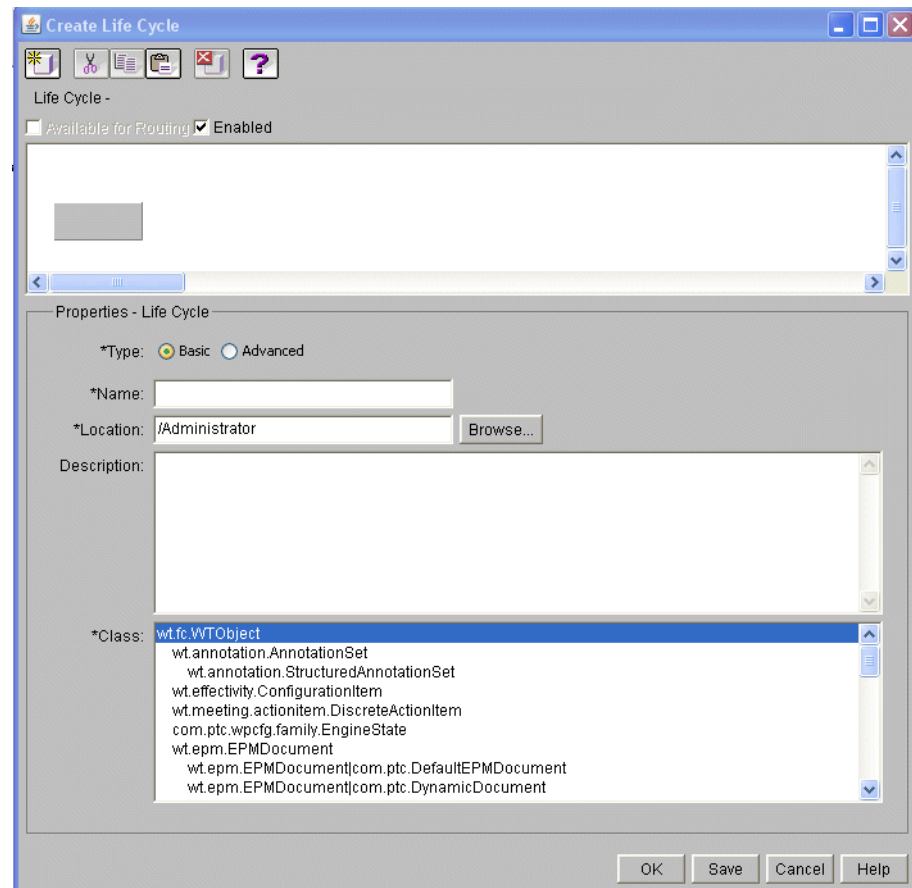
The **Class** display in the panel above provides a tree view of all object types subject to life cycle management. You must choose the type to which this life cycle applies. Because Windchill types are hierarchical, the life cycle is applicable to the selected type and all of its subtypes. A type can inherit more than one life cycle; you can directly associate a life cycle with a given subtype. For example, you could associate a life cycle with the type WObject, and all its subtypes would also be associated with that life cycle. You could also associate those subtypes (for example a change request, WTChangeRequest2) with other life cycles.

When users create objects that are subject to a life cycle, a life cycle must be selected as part of the creation process. The list of selectable life cycles is dependent on what has been defined in the object initialization rules for the selected type. If the user does not have create permission based on the initial state of a life cycle, that life cycle will be filtered out of the list.

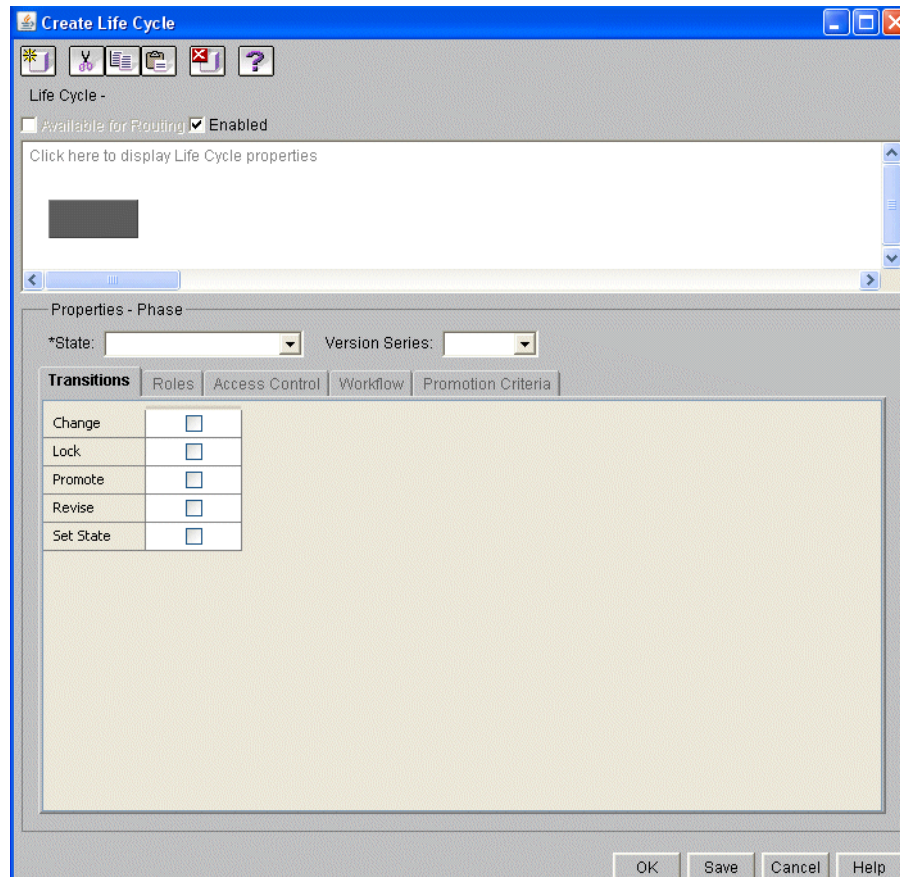
## Defining Life Cycle Phases and Gates

The following graphic displays the **Create Life Cycle** window.

When you first create a new life cycle, select the New Phase icon  to open a single undefined phase. A grey rectangle appears in the diagram area.

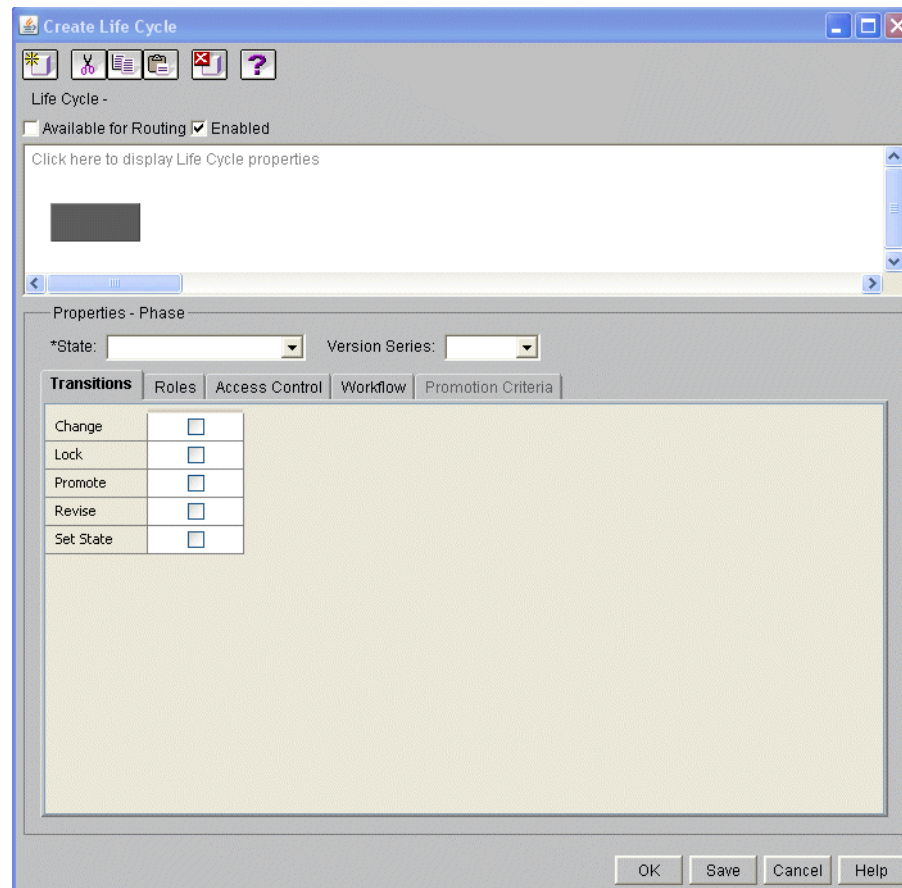


The **Properties–Phase** panel is displayed when you click on a phase (the grey section) of the life cycle diagram. The following figure displays the **Properties–Phase** panel of a basic life cycle.



View the properties of the life cycle itself by clicking anywhere on the life cycle diagram background to open the **Properties–Life Cycle** panel. The **Properties–Life Cycle** panel also opens when you delete a phase.

The figure that follows displays the **Properties–Phase** panel of an advanced life cycle.



Note the difference between the two tabs. With a basic life cycle, the only tab available is **Transitions**. With an advanced life cycle, **Transitions**, **Roles**, **Access Control**, **Workflow**, and **Promotion Criteria** are available for all but the last states.

Use the tabs to define the characteristics of the life cycle phases and gates.

The following table provides a brief description of the phase properties.

Phase Property	Description
State	<p>When you add a phase icon to the life cycle diagram, you must choose the state it represents from the drop-down list, which is populated with all available states.</p> <p>Windchill provides predefined states (for example, In Work and Under Review). You can define additional states by adding them to the StateRB.rbInfo resource file. For additional information, see, <a href="#">Defining Additional Life Cycle States</a>.</p> <p>When you select a state, its name appears on the phase icon. The other phase properties you add define the behavior associated with an object while it is in this state.</p>
Version Series	<p>For each state, you can define a version series. The version series determines the versioning scheme for objects in the state. The choices in the Version Series drop-down list include mil std and numeric. The chosen series is an attempt to select the correct versioning scheme; various runtime circumstances determine whether the specified series selection is actually employed.</p> <p>For more information, see <a href="#">State-based Revision Sequences by Life Cycle State</a>.</p>
Transitions	<p>For each life cycle phase, you can define the transition path from the selected state to any other state within the life cycle template. If you do nothing from the Transitions tab, the template uses the default state transitions. You can define additional transitions by adding them to the TransitionRB.rbInfo resource file as described in the <i>Windchill Customizer's Guide</i>. This resource file is located in the &lt;Windchill&gt;/src/wt/lifecycle directory.</p>

Phase Property	Description
Roles	<p>For each life cycle phase, you can select roles (for example, Reviewer or Workflow Assignee).</p> <p>For each life cycle phase, you can select roles (for example, Reviewer or Workflow Assignee). Specific users should be assigned to those roles in the Team so that proper role resolution takes place at runtime.</p> <p>These roles are mapped to role players. A role player can be specified as a user, a user-defined group, an organization, an actor, or another role.</p>
Access Control	You can also define access control rules that will be in effect for this phase. These rules, which specify permissions for each role, will add and override (in case of conflict) to those that are already in effect for the object, based on the domain's access policy.
Workflow	You can choose a workflow process to be associated with this life cycle phase and with the gate representing promotion to the next phase.
Promotion Criteria	You can define the criteria for promotion of an object from this phase to the next phase in the life cycle.

An object must be approved and explicitly promoted in order to move forward in its life cycle. To illustrate this, promotion gate icons divide the phases in the life cycle diagram.

## State-based Revision Sequences by Life Cycle State

The versioning schemes for Windchill PDMLink are identified based on the object type through object initialization rules.

For a specific object type, such as a part or document subtype, each object type can use a specific version scheme referenced by a scheme name.

These version schemes are then identified in an XML file. For example the out-of-the-box XML file loaded with Windchill PDMLink is:

```
StateBasedVersioning.xml
```

This file contains the sequences and seeds (for example, NUMERIC and MILSTD).

**Note:** To change version schemes, do not manually modify the XML file; instead, use the steps described in the [Setting Up a File-based or State-based Versioning Scheme](#) section of the [Contexts](#) chapter to load an updated XML file.

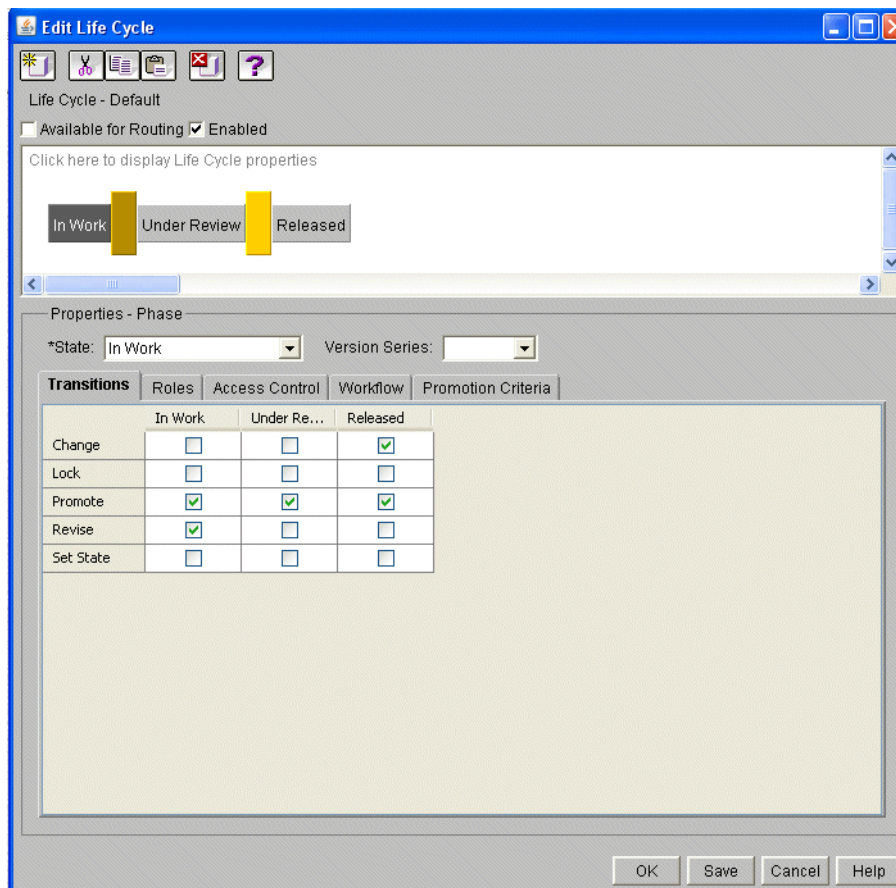


For information on setting up the state-based versioning scheme that corresponds with the **Version Series** drop-down list that is available for each state, see [Administering the Versioning of Parts, Documents, and CAD Documents](#) in the Contexts chapter.

## Defining Transitions

Click the **Transitions** tab of the **Properties–Phase** panel to select the transitions to be associated with this life cycle phase. When you create, edit, or view a life cycle, the Transitions tab displays.

The figure that follows shows the Default life cycle on the **Edit Life Cycle** window, with the **Transitions** tab active.



In the figure above, the **In Work** state is selected in the life cycle panel. This means that information displayed on the tabs in the graphics shown below, including the **Transitions** tab, correspond to the In Work phase of the life cycle being edited. A transition is a named path from a source phase to a destination phase in a given life cycle. In this figure, **In Work** is the source phase. The table of check boxes indicates to which destination states there are particular transitions.

For example, the check in the first column of the third row means that there is a Promote transition from **In Work** to **In Work** (this is an example of a reflexive transition, where the source and destination phases are the same).

Similarly, the check in the third column of the first row means that there is a Change transition from **In Work** to **Released**.

So, the defined transitions shown in this figure are:

In Work – Change to Released

In Work – Promote to In Work

In Work – Promote to Under Review

In Work – Promote to Released

In Work – Revise to In Work

These are not all the transitions for the entire life cycle, but only for those with **In Work** as a source phases. Other transitions are viewed by activating the **Transitions** tab when other source phases are selected.

## Transition Defaults

Life cycle transition defaults are applied whenever a life cycle is saved (either interactively or via a bulk load) or is upgraded and does not otherwise contain explicit transition information.

Default transitions are initialized according to a specification recorded in the *wt.properties* file. The specification is encoded using a small language, whose grammar is given (beginning with a valid *wt.properties* entry):

```
wt.lifecycle.transitions.default.<transition_name>=<expression_list>

transition_name = { key element of wt.lifecycle.Transition }
expression_list = expression ["," expression_list]
expression = term ["|" expression]
term = "ALL" | "FIRST" | "LAST" | state_name
state_name = "\"" { key element of wt.lifecycle.State } "\""
```

Each property entry specifies a rule for creating default instances of a particular transition type in each life cycle. Each rule is applied to each state in each life cycle. Each expression in each *expression\_list* is evaluated to create transitions to the specified states. However, each *term* in each *expression* is evaluated in order of appearance only until one is encountered that validly specifies one or more states; the remaining terms in the expression are ignored.

The out-of-the-box default transition expressions are:

```
wt.lifecycle.transitions.default.PROMOTE=ALL
```

```
wt.lifecycle.transitions.default.CHANGE="RELEASED"
```

```
wt.lifecycle.transitions.default.REVISE=FIRST
```

These rules can be interpreted as follows:

- **PROMOTE:** create a PROMOTE transition from each state to every state in each life cycle.
- **CHANGE:** create a CHANGE transition from each state to the RELEASED state in each life cycle. If the life cycle does not have a RELEASED state, no transition is defined.
- **REVISE:** create a REVISE transition from each state to the first state in each life cycle.

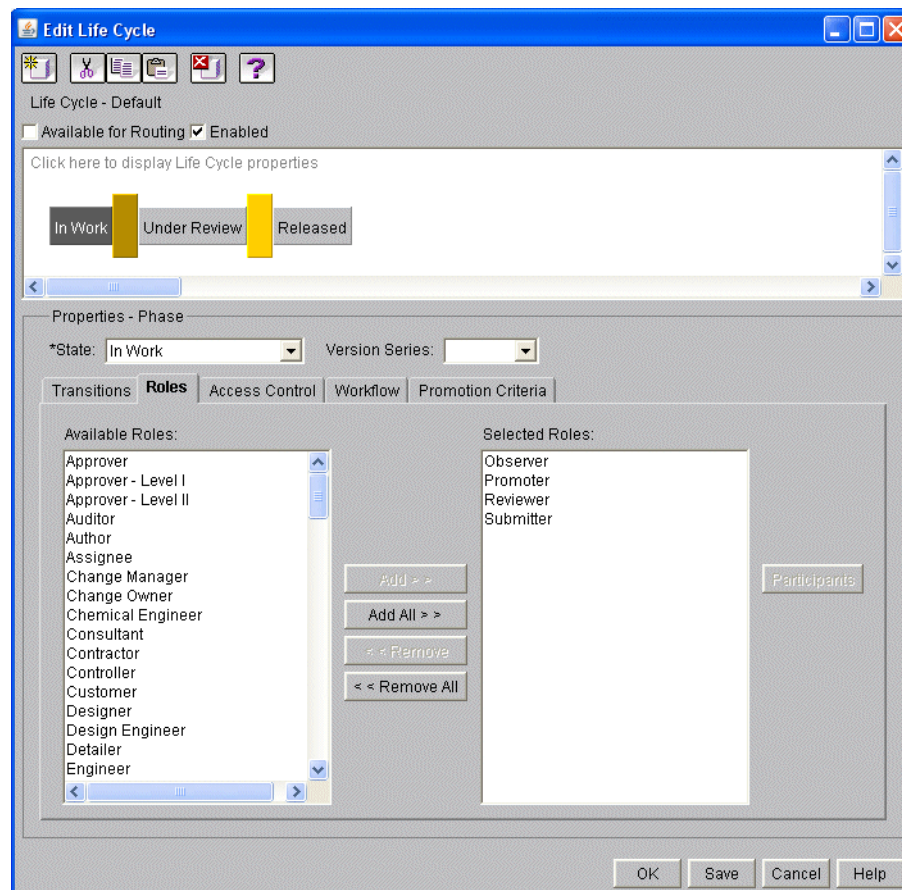
## Selecting Life Cycle Roles

**Note:** This section applies only to advanced life cycle templates.

Click the **Roles** tab of the **Properties–Phase** panel to select the roles to be associated with this life cycle phase. When an object is promoted to this phase in its life cycle, these roles are resolved to principals (users, user-defined groups, or organizations) who perform one of the roles in the **Available Roles** list.



The following figure is an example of the **Roles** tab panel.



To add a role to the phase, select the role and click **Add** to move it to the **Selected Roles** list. You can also click **Add All** to move all the displayed roles to the **Selected Roles** list. Click **Remove** or **Remove All** to delete roles from the list.

**Note:** The **Available Roles** list is populated with predefined roles. You can define additional roles by adding them to the RoleRB.rinfo resource file. Defined roles are added to this list when you recompile RoleRB.info and deploy the class file to your production environment. For additional information, see the enumerated types information in the *Windchill Customizer's Guide*.

## Role Mappings

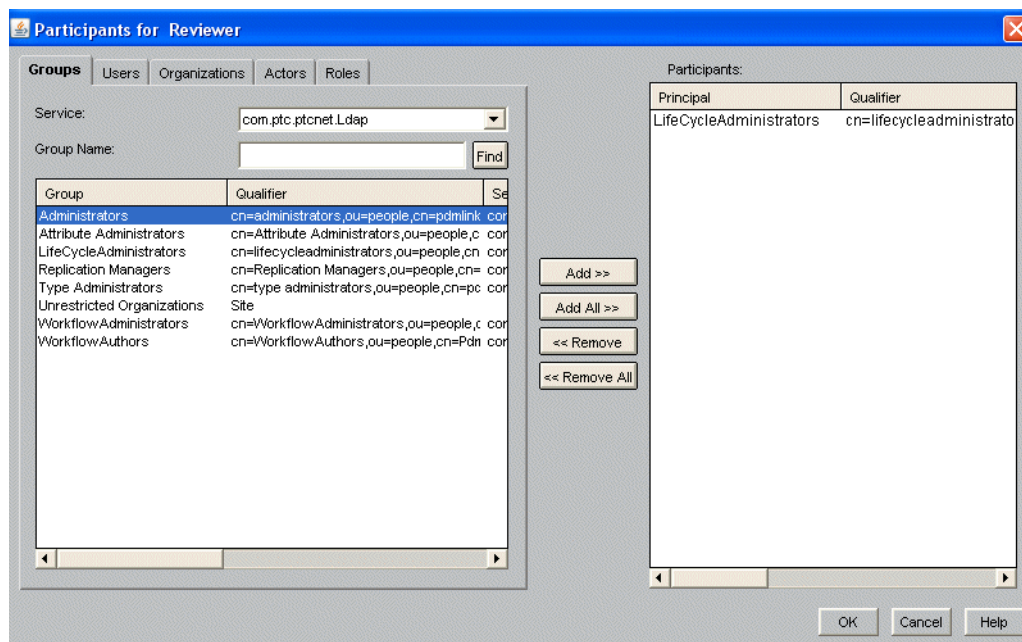
A role mapping is resolved in one of several ways:

- You can directly map life cycle roles to users, user-defined groups, organizations, or other roles. However, since organizations generally want to define only a small number of life cycles, it is not often practical to map life cycle roles directly to principals. Using life cycles and teams together allows role participants to be identified at runtime, rather than making this mapping an explicit part of the life cycle definition.

- You can map life cycle roles to team roles. At runtime, the role is resolved according to the team role mapping. (For example, the life cycle role Promoter could be mapped to the team role Team Leader, and the life cycle role Promoter would be resolved at runtime according to the Team Leader role, as mapped in the team.)
- You can map a life cycle role to an actor. That is, you can map a role to someone who performs a specific action within the context of the business object. At runtime, this role is then resolved to the principal who created the object with which the life cycle is associated. For example, you could assign the Creator actor to the Submitter role for a given life cycle phase. For that phase, the user who created the object would be assigned the Submitter role at runtime. If the Submitter role is defined in the team, it resolves to the team's Submitter role.

### Selecting Participants for Roles

To add participants to a specific life cycle role, select a role in the **Selected Roles** window. Then click **Participants** on the **Roles** tab, to choose participants for the selected role. The **Participants for** window opens (as shown in the following figure), allowing you to choose users, user-defined groups, organizations, an actor, or other roles to be mapped to this role.



- Click the **Users** tab to choose from a list of users defined in the context-specific list of users filtered by service (source) and access control. From this tab:
  - Select **All** or select a specific directory service, from the **Service** drop-down list. The default service displayed is the service associated with the bundled Aphelion Directory. If users are contained in an enterprise directory, be sure to select the service for that directory.

- To limit the users displayed in the table, enter the name of a user in the **Full Name** or **User Name** field, and click **Find**. You can use the \* or % wild card characters in the name, where \* represents one or more characters and % represents exactly one character.

The specified services are searched and the matching results display in the table. When your system has a lot of users, this is the recommended method of finding a specific user.

If you leave the **Users** field blank and click **Find**, all users from the selected service display in the table. If your system has a lot of users, displaying all users may take a long time.

- Similarly, click the **Groups** tab to choose from a list of user-defined groups or click the **Organizations** tab to choose from a list of organizations.
- Click the **Actors** tab and choose an actor to base your selection on a particular user action. Creator is the only actor defined. The Creator is resolved at run time to the user who created the selected object.
- Click the **Roles** tab and choose a role to resolve the life cycle role.

To add a principal to a role, select it and click **Add** to move it to the **Participants** list. You can also click **Add All** to move all the displayed users, user-defined groups, or organizations to the **Participants** list. Click **Remove** or **Remove All** to delete participants from the list.

Click **Help** to view detailed instructions for selecting participants.

## Defining Life Cycle Access Control Rules

**Note:** This section applies only to advanced life cycle templates.

Access to a specific object (for example, a document or a part) is controlled by the access policy for the domain in which the object is located. In many contexts, policy access control lists (ACLs) are sufficient for controlling access to objects; however, when an object is part of a life cycle, there are often many different principals who must participate in moving an object through its phases.

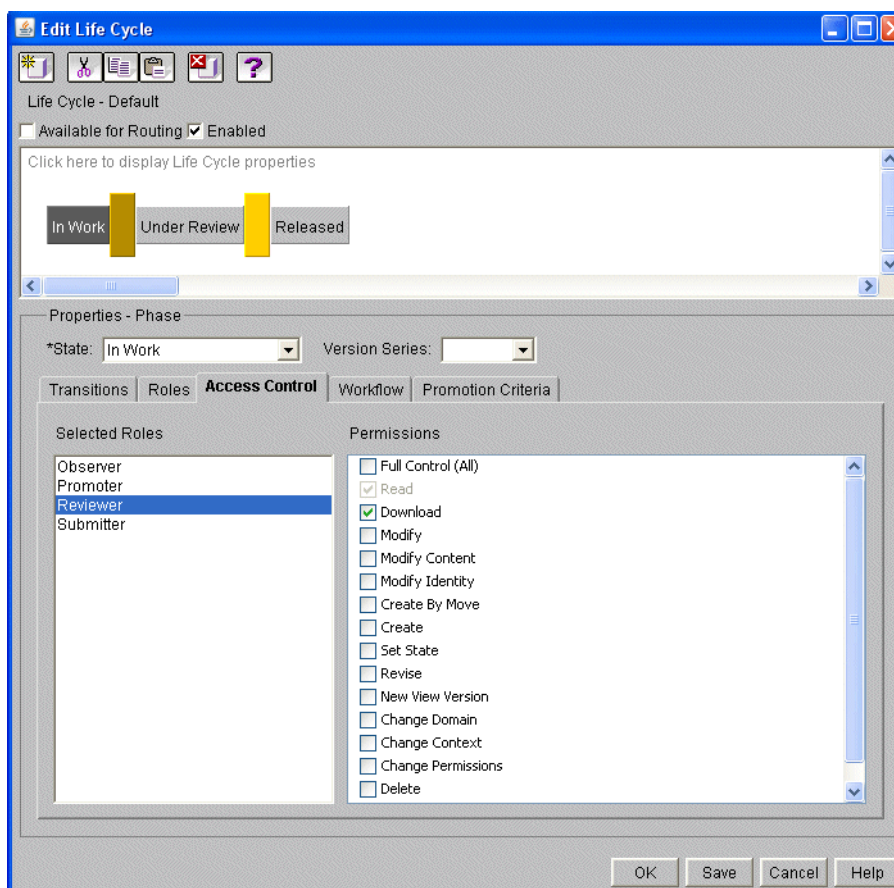
Depending on your security needs, you may not want to create domain policy rules that provide all of the necessary permissions. If this is the case, access to an object can be controlled by an *ad hoc ACL* that is part of its life cycle. In general, policy ACLs apply to an object type within a domain, and ad hoc ACLs apply to an instance of the type while that object remains in a specific life cycle phase.

Rules in an ad hoc ACL are added to the rules in the policy ACL for a given object. The ad hoc rules exist for the duration of a specific life cycle phase. These rules grant roles additional access to an object during the life cycle phase. This access is then revoked when the object moves to a new phase and the participant no longer needs the ad hoc permissions.

Ad hoc ACLs can only grant permissions. They cannot be used to deny access to an object.

Create ad hoc rules from either the **New** or **Edit Life Cycle** window. Click the **Access Control** tab, select a role, and choose the appropriate access permissions.

The following figure is an example of the **Access Control** tab panel.



All roles are automatically given Read permission, so the associated principals can access their tasks and view the object. By default, submitters are automatically given Modify permission so they can submit the object for promotion as part of editing it; however, you can change this at your site. For each role, you can also select one or more of the permissions described in the [Access Control](#) chapter earlier in this guide. You can also learn more about domain access control and the relationship between policy and ad hoc ACLs.

The following table describes the permissions available on the **Access Control** tab:

Permission	Rights Given
Full Control (All)	Full control. A user, group, or organization granted the Full Control (All) permission is granted all permissions currently defined and any that might be defined in the future. Therefore, if new permission types are defined, you do not have to write rules that specifically grant them to users, groups, or organizations with full control access.
Read	Allows the selected role the right to know the existence of an object and to view the object and its attributes. Additionally, if the object has content, you can view an object's content information such as the file path to a local file or the location of external storage. This permission does not allow you to view the actual contents of the file.
Download	Allows the selected role the right to download local files that are the primary content or are attachments of an object. This right is applicable to objects with content, such as documents or drawings.
Modify	<p>Allows the selected role the right to change the attributes of an object, as well as other characteristics that are part of the object definition but are not controlled by the Modify Content or Modify Identity permissions.</p> <p>For versioned objects, a user must have the Modify permission on the latest iteration of each version of a target object to update the attributes common to all versions that are not part of the object's identity. Modify permission on a version of a target object is required to modify that version's attributes.</p>
Modify Content	<p>Allows the selected role the right to modify any local file, URL, or external storage for the primary content and attachments of an object with content. This includes modifying content information and adding, replacing, or deleting content.</p> <p><b>Note:</b> The <b>Modify</b> permission is required to check out an object. A checkout is required before user can modify the content.</p>

Permission	Rights Given
Modify Identity	<p>Allows the selected role the right to modify a subset of the attributes that determine the identity of an object.</p> <p>For a part, this subset includes the part number and the organization identifier (such as cage code) of the part, but not part name (which is often treated as a short description).</p> <p>For a folder, the attributes include the folder name.</p> <p>The subset of attributes affected by the Modify Identity permission for a given object type is determined through the modeling of classes. For information on customizing the code to modify the set of attributes used in determining the identity of an object, see the <i>Windchill Customizer's Guide</i>.</p>
Create By Move	Allows the selected role the right to move an object into an administrative domain.
Create	Allows the selected role the right to create a new object.
Set State	<p>Allows the selected role the right of a user to perform a set state operation where a state transition has been defined to allow the transition from the current life cycle state to the new state.</p> <p>To perform a set state operation, a user must have the Set State permission and there must be a valid state transition defined between the current state and the desired state. If there is no transition defined, the user must have the Administrative permission to perform the operation.</p>
Revise	<p>Allows the selected role the right to revise an object. Revising creates a new version of the object at the same level as the original in the version tree. For example, you can create revision B from revision A.</p> <p><b>Note:</b> Windchill ProjectLink does not support the Revise permission.</p>
New View Version	<p>Allows the selected role the right to create a version for a specific view.</p> <p><b>Note:</b> Windchill ProjectLink does not support the New View Version permission.</p>

Permission	Rights Given
Change Domain	<p>Allows the selected role permission to move objects out of an administrative domain. This permission does not provide the right to do anything other than allowing a user to change the administrative domain of an object. For example, to move an object from one folder to a new folder that has a different domain association, the user must have the following permissions:</p> <ul style="list-style-type: none"> <li>• Modify permission on both the source and destination folders as well as,</li> <li>• Create By Move permission on the object in the destination domain and,</li> <li>• Change Domain permission on the object in the source domain.</li> </ul>
Change Context	<p>Allows the selected role to move a business object out of a context.</p> <p><b>Note:</b> This permission does not provide the right to do anything other than allow a user to change the context of an object. To move a business object to a domain in a different context (which is currently required because domains do not span contexts), the user must also have the Change Domain permission.</p>
Change Permissions	<p>Allows the selected role the right to change the ad hoc permissions that others have. Users, groups, or organizations granted the Change Permissions permission are allowed to change the ad hoc permissions of others to the permissions they themselves have or to a subset of the permissions they have.</p>
Delete	<p>Allows the selected role to delete an object.</p>
Administrative	<p>Allows the selected role the right to perform certain administrative tasks. (For example, an administrator would have the right to break a lock or change an object's owner.)</p>

## Associating a Workflow Process with Phases and Gates

**Note:** This section applies only to advanced life cycle templates.

By default, all advanced life cycles have predefined workflow processes associated with the phases and gates.

When a business object reaches a specific state, the workflow process associated with that state will be automatically initiated and the business object will be forwarded through the workflow process.

The workflow process that moves objects through the associated phases and gates is separate from the life cycle of the object. You can manage one or more objects with a workflow process.

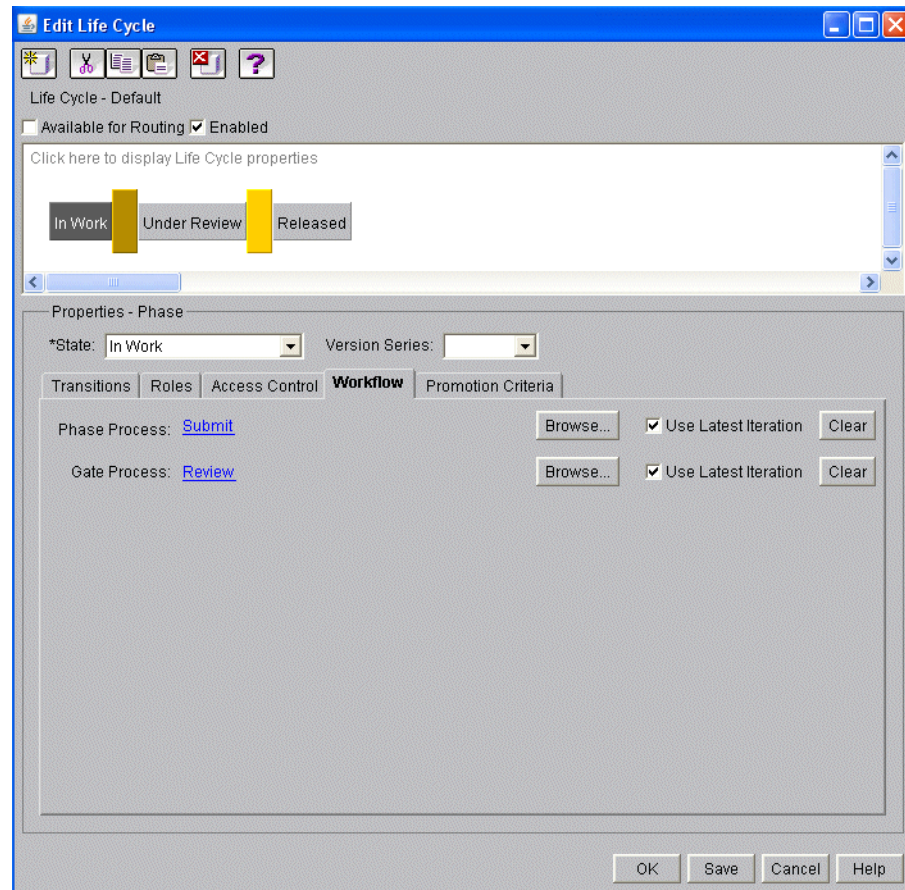
To change the workflow process that is associated with a phase or a gate, modify the following properties in the `wt.properties` file:

- `wt.lifecycle.defaultPhaseProcess`
- `wt.lifecycle.defaultGateProcess`

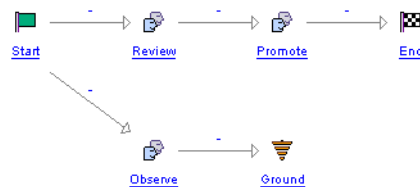
As shown in the figure that follows, the Submit process is automatically associated with the In Work phase. The Review process is associated with the gate by which an object moves from In Work to its next phase.



**Note: Use Latest Iteration** is selected, so the most recent iteration of the workflow process template is used at instantiation. If this check box is cleared, the specific iteration selected is used, even if it is not the most recent.



As a result, the Review workflow, which is shown in the following Workflow Process Editor figure, defines the process and activities that are part of moving an object forward from the In Work phase.



This workflow process has three defined activities: Review, Observe, and Promote. You can view the properties of each link and activity within the process on the Workflow Process Editor. For example, for the Review workflow, the participant to be assigned the Review task is the Reviewer role.

Therefore, when an object is submitted for promotion from the In Work phase and the Review workflow process is started, the Reviewer role is mapped to an actual user, based on role mappings in either the life cycle or a team template. The Review task is added to the Assignments table for the user.

The Submit and Review workflow processes are predefined and available for your use when Windchill is installed; however, your organization may have a number of additional workflow processes in place. To associate a specific workflow process with a phase or gate, click **Browse** to locate and select a process from the shared location.

## Defining Promotion Criteria

**Note:** This section applies only to advanced life cycle templates.

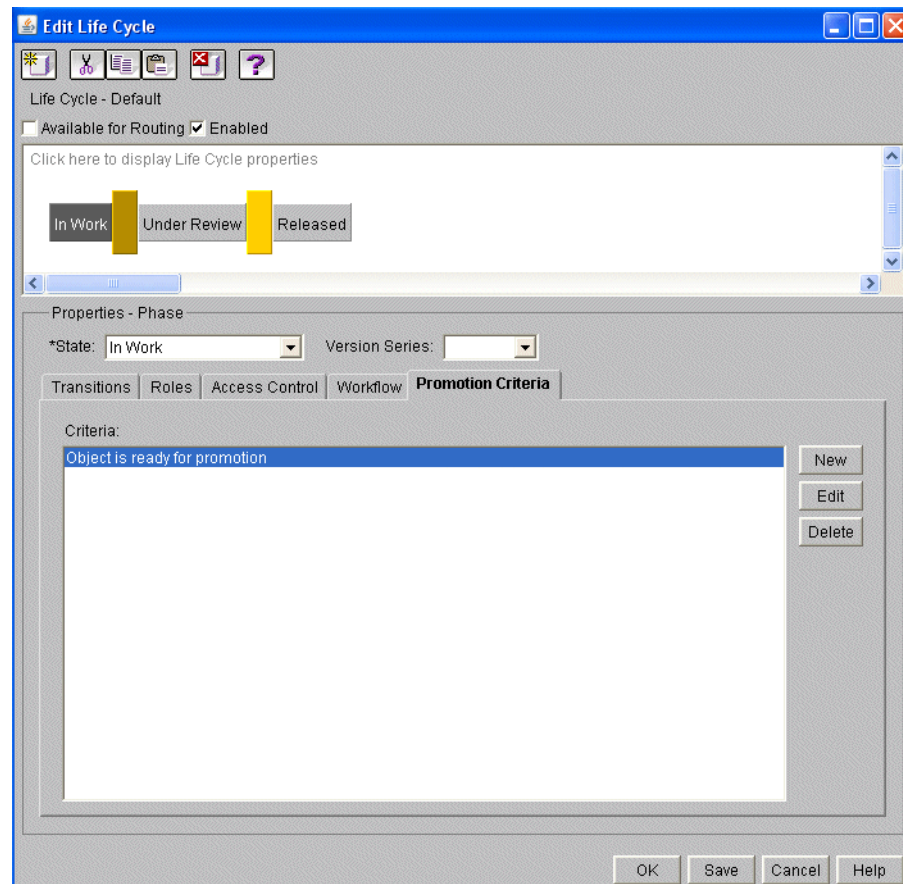
Life Cycle Management does not enforce the satisfaction of promotion criteria. For example, reviewers' votes are not tabulated, and any reviewer is allowed to check off one or more promotion criteria. The promoter can choose to promote the object to its next life cycle phase regardless of whether all reviewers have voted or all promotion criteria have been satisfied.

However, the following specifies promotion criteria guidelines you can use to help reviewers and promoters make appropriate decisions, based on your site's processes:

- Click the **Promotion Criteria** tab to view the existing criteria set.
- Click **New** or **Edit** to create or modify the criteria.
- Click **Delete** to remove a criterion.

The online help contains detailed instructions to help you perform these actions.

The following graphic displays the **Promotion Criteria** tab panel.



When you click **New** the **New Criteria** window opens. When you click **Edit** the **Edit Criteria** window opens. Type a criterion statement in the field provided. For example, you could enter the following statement for a given phase of a life cycle:

All reviewers have voted for promotion.

This statement would then serve as a criterion for promoting an object from the current life cycle phase to the next.

## Importing and Exporting Life Cycle Templates

Before you begin importing or exporting life cycles, you should be familiar with the following information:

- Upgrade to the latest maintenance only release (MOR) as it becomes available, to ensure you have the latest enhancements to the import and export functionality.
- You can import a life cycle into a later version of Windchill; but not to an earlier version. That is, importing and exporting are not backwardly compatible. To import from versions prior to Windchill 7.0, you must convert the CSV files to XML format. Use the CSV2XML utility to convert CSV files to XML files. The syntax is as follows:

```
windchill wt.load.util.CSV2XML -input <input file or directory>  
-output <output directory> -root <root directory> -help
```

- Importing or exporting life cycles creates objects in a JAR or ZIP file format. (This is the same format that the load.Installer functionality uses.)

Use a zip utility like WinZip to create a ZIP file consisting of XML files. To create a JAR file, use the following syntax:

```
jar -cf <jar file name> <xml files to be included>
```

or

```
jar -cvf (for verbose output to print each file name as it is  
added)
```

- Importing a JAR or ZIP file with one or more XML files creates one or more life cycle templates (depending on how many templates were defined in the XML files).
- There is no limit to the number of life cycles that you can export. You can export multiple life cycles into a single JAR or ZIP file. Select them (on the **Life Cycle Administrator** page) and click **Export**. All the selected life cycles are exported to the same JAR or ZIP file.
- Errors can occur, especially when importing life cycles. Some errors result in messages displayed; others cause a loss of data. Check the method server log for error information.

- When you export a life cycle, only the life cycle itself is exported. This includes references to underlying objects, such as principals, roles, and actor roles. However, the underlying objects themselves are not exported. If the export file is used to import the life cycle into another system, the underlying objects must first exist in the system, or the import fails and errors appear in the method server log. This can occur, especially when importing the object into a different system. Be certain that all underlying objects referenced in the XML representation of the life cycle exist.
- If a life cycle is imported and a life cycle with the same name already exists in the Import directory, the results depend on the Iteration On Import setting in the wt.properties file. If it is set to true, the imported life cycle is appended to the existing life cycle as a new iteration. If it is set to false, the imported file causes a method server exception, stating that there is a duplicate name, and the life cycle is not imported.

## Importing

Use the **Import** button in the **Life Cycle Administrator** window to import a life cycle template.

You can import one or more life cycles from a JAR or ZIP file in the life cycles export directory. Select a file from the **Import** dialog box, and click **Import**.

## Exporting

You can export one or more life cycles into an XML representation of the life cycle in the life cycles export directory.

Use the following procedure to export life cycle template(s):

1. Select a template from the **Life Cycle Administrator** window, and click **Export**.

The **Export** button is disabled if you do not have a life cycle selected.

2. A grant permission window may appear asking for permission to access the local file system and to write a file on that system. If you select the remember selection check box, permission need only be granted once. Once permission is granted, a **Browse for file** window opens, defaulting to the system **temp** folder.
3. You can pick a file that exists or type in a new name. If the file name exists, you are asked to confirm to overwrite the file. You must click **Yes** to continue the export. If the file name does not exist, a new file is created.

There is no confirmation that the export is completed. When the progress bar and the hourglass on the life cycle administrator applet disappear, the export is complete.

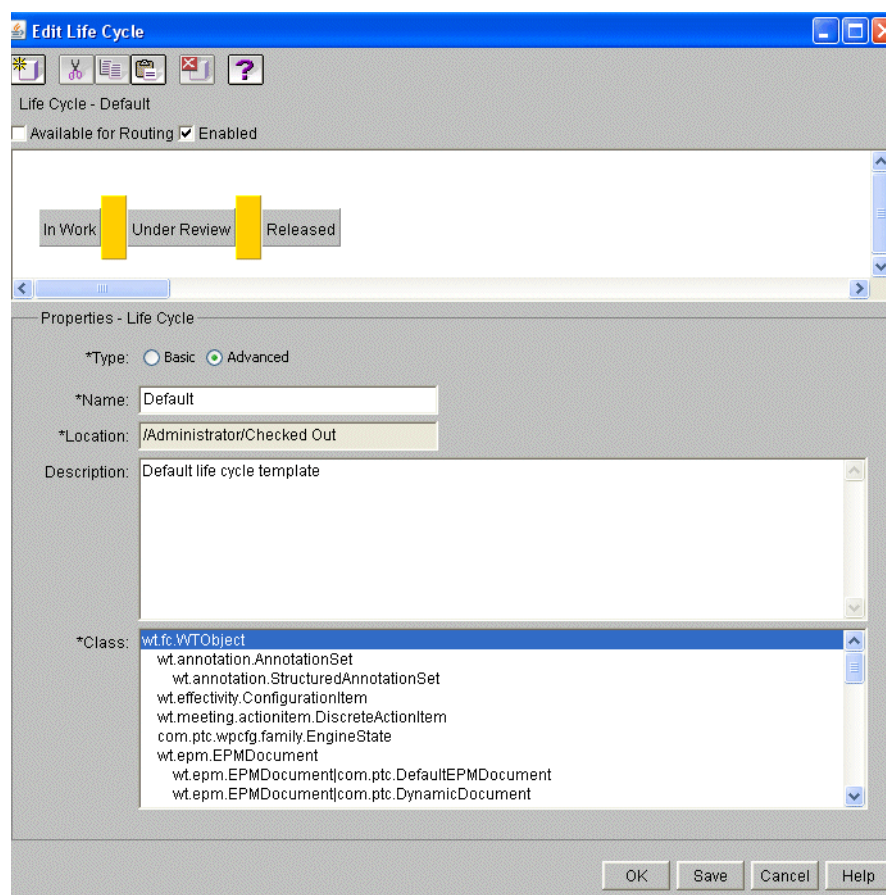
If you want to use a template in another context (such as between two organizations, two application contexts, or an application context and organization context), export the template from the source context and import it into the target context.

You can edit a template to make it specific to an organization or application context by copying them (using **Save As**) from parent contexts.

## Overriding and Reassigning Life Cycle and Team Templates

In Windchill PDMLink, during the creation of a part and a document, the life cycle template and the team template can be manually selected. It is also possible to reassign a life cycle and reset a team template on an existing part or document. In most cases, the manual selection can be done from either a drop-down list or a Search function.

The life cycle drop-down list is automatically generated based on the object type being created and the class attribute value that is assigned in a life cycle template. A life cycle template contains an attribute that specifies for which class the life cycle is valid. The class attribute can be set to a value of an abstract class, a parent class, or a leaf node class. See the **Class** field in the graphic below.



Based on the object type being created, all enabled life cycle templates which have a class attribute value matching the selected object type being created, or a parent type, or an abstract class, will be included in the drop-down list.

The Search function allows the user to search for any enabled life cycle template that the user has access to, independent of the life cycle templates class attribute value.

## Manual Selection of Life Cycle and Team Templates

The action to manually select the life cycle and the team template during the creation of a part and a document is available to users that have administrative privileges set on the object. For more information, see the [Access Control](#) chapter.

## Reassigning Life Cycles and Resetting Team Templates

The action to reassign life cycle templates and reset team templates on existing parts and documents is only available in Windchill PDMLink. Users must have administrative privileges set on the object for either of these actions to be available. For more information, see the [Access Control](#) chapter.

Additionally, resetting the team template is only available for objects with advanced life cycles.

## Defining Additional Life Cycle States

Windchill includes many predefined life cycle states and roles in the out-of-the-box life cycle templates. You can define additional states and roles by customizing the StateRB.rbInfo and RoleRB.rbInfo resource files as described in the *Windchill Customizer's Guide*. These resource files are located in subdirectories under the <Windchill>/src directory. Newly defined states and roles are added to the existing states and roles when you recompile the resource files and deploy the class files to your production environment. For additional information, see the enumerated types information in the *Windchill Customizer's Guide*.



**Caution:** Removing a value you previously added to an enumerated type (for example, removing a state in the StateRB.rbInfo resource file), could result in serious runtime error. Do not remove a state unless you are certain there is no reference to it within the system.



## Access Control for Life Cycle Administration

As described in this chapter and [Teams](#), administrators create life cycles and teams templates.

The access control rules listed in the following table provide life cycle administrators with permissions needed to manage life cycles and teams, and to move them after they have been created. These rules need to be defined for the domains (or ancestor domains) associated with folders in which life cycle templates and teams will reside.

Object Type	Permissions Required
AdministrativeDomain	Read
LifeCycleTemplate	Read, Modify, Create, and Delete
Team Template	Read, Modify, Create, and Delete
Cabinet	Read and Modify
SubFolder	Read and Modify
WTContainer	Read

## Best Practices

Site, organization, and application context managers manage life cycle templates.

- Site administrators create, modify, delete, and view life cycle templates in the site context.
- Organization administrators create, modify, delete, and view life cycle templates in the given organization context. Organization administrators can view life cycle templates from the site context.
- Application context managers create, modify, delete, and view life cycle templates in the given application context. They can view life cycle templates from the parent organization context and the site context.

**Note:** The Life Cycle Administrator is not available to project or program application context managers.

The Life Cycle Administrator client displays a table that lists all life cycle templates belonging to the given context, plus those belonging to its parent contexts. A column in the table identifies the context owning each life cycle template.



When you create a life cycle template, the system saves the new life cycle template in the System cabinet or folder of the context in which it is created. Consequently, when creating a new life cycle, the **New** dialog for life cycle templates disables the location field since it is the system, not the user, that decides where the new life cycle template is to be located.

**Note:** When assigning a workflow template to a life cycle template, you see a list of valid workflows. The list of valid workflow templates includes the ones defined in the given application context, plus those defined in the parent organization and the site contexts. Workflow templates defined in a sub-context override and filter out the workflow templates defined in parent contexts having the same name.

The search scope used to locate user-defined groups is determined by the type of administrator doing the search. For more information about the search scope, see [Searching for Principals in Administrative Clients](#) in the [Principals \(Users, Groups, and Organizations\)](#) chapter.

## Life Cycle Support in Windchill ProjectLink

The following table shows the life cycle support in Windchill ProjectLink.

Feature	Support Level	Notes
Life cycle template definition	Full	Organization administrators may create, edit, and delete life cycles for their organization.
Life cycle-managed object support	Partial	Only document and part items are supported.
Phases	Full	
Gates	Full	
Default life cycles	Partial	The following default life cycles are supported out of the box: Default, Basic, Approval, Review, and Release.
Workflow association	Full	Workflow templates must be defined in the organization, and are available for reference at a life cycle phase or gate.
Roles	Partial	Roles must be defined in the project or program instance.

Feature	Support Level	Notes
Role-based access control	None	Access rules defined for objects in the project or program instance take precedence. By default, members of the project or program manager group have full access to all objects in the project or the program. The life cycle access rules by role for each phase and gate are ADDITIVE. This means access rights can be added for each role only at each phase and gate.
Criteria	Full	
Life cycle template import export	Partial	There is no organization association with the imported or exported template so the life cycle import/export folder should be cleared between uses.
Access to current life cycle state	Partial	The life cycle state is provided on the Details page for the item.
Set life cycle state	Full	Object owners and project managers can set the life cycle state for a document or part with the <b>Set State</b> action. For more information about overriding the life cycle state, see <a href="#">Overriding and Reassigning Life Cycle and Team Templates</a>
Access to maturity history	None	The maturity history for a part or document is not available in the user interface.
Queries based on life cycle state	None	No user interface is provided to support life cycle-based queries, but this may be supported in a future release.
Access to running process through life cycle	None	

## Managing Promotion Processes in Windchill PDMLink

The decision to promote an object from one state to another often pertains to a business decision. As such, it is important to make sure that the Products and Libraries are set up correctly to allow for the proper approvals to be obtained prior to promotion. See [Products and Libraries](#) for detailed information. For any process template used in the promotion process, users assigned to the particular tasks are obtained from the following:

- The process template itself (the process has a member defined for a specific task)
- The state of the life cycle of the promotion request
- The team instance of the promotion request

The out-of-the-box templates do not define users for the roles defined in the process templates or life cycle templates used by the promotion request.

The team template provided for the promotion request defines the roles Approver and Reviewer; however, no default users are assigned to these roles. The Approver and Reviewer roles should be populated by the context team. For information about context teams, see [Context Teams](#) in the [Teams](#) chapter.

If using a process template for the promotion process that defines roles in addition to Approver and Reviewer, add the roles to the team template used for promotion request creation. You can modify team templates through the **Team Administrator** found on the **Utilities** page of each tab. Create a copy of the team template (using the Save As functionality), edit the new template to add any new roles, and then modify the appropriate object initialization rule for the promotion request to point to the new team template in the context of interest.

The primary method for allocating users to roles is by using the context team. You can find the team on the **Team** page of the **Product** or **Library** tab. Both out-of-the-box product templates define two context roles, Promotion Approvers and Promotion Reviewers, to work with the promotion process. These context team roles, along with role mapping copies any users and user-defined groups from these roles into the team instance of the promotion request at creation. When a new product is created, the Promotion Approvers role is empty; any approval tasks are sent to the promotion request creator unless this role (or another mapping) is properly defined. For each product, resolve at least one user into the promotion request team instance for the Approver role to ensure that a proper approval is obtained prior to promotion. Out-of-the-box, this can be done in the following ways:

- Adding the approver to the Promotion Approvers role of the context team
- Modifying the team template to have a default participant for the Approver role
- Modifying the preference mapping to add additional roles for the Approver role, ensuring that at least one user is assigned to at least one of the roles in the context team.

## Life Cycle Teams in ProjectLink

When routing a life cycle-managed object in Windchill ProjectLink, only the members of the context team roles that match the life cycle phase roles will be pulled into the life cycle team. The entire context team is no longer pulled into the life cycle team. This is the default behavior for the wt.property file (wt.team.addEntireContainerTeam). This wt.property file can be used to toggle between the default behavior and the previous behavior that pulled the entire context team into the life cycle team.

A message will be thrown if a user tries to create an advanced life cycle template via the Life Cycle Administrator and does not associate a workflow template to the life cycle phases or gates. In this case, the message requests the user to change the life cycle template from advanced to basic for better performance and scalability.

## Restrictions on Moving Objects Between Contexts

If you create a life cycle template in a specific product or library and the template is not available in other products or libraries, then you cannot move any object that uses the life cycle template.

**Note:** For demonstration purposes, the Product Design template (available by default when you create a product) includes object initialization rules that set different default life cycle templates and versioning schemes for parts, documents, and CAD documents than is set out-of-the-box at the site level. This template demonstrates the use of simplified life cycles and state-based versioning. Unless you have overriding business practices that require unique rules, PTC recommends that if you want to use these features, you modify the organization or site object initialization rules and remove them from the Product Design template.

# 16

## Workflow

This chapter provides information about workflow processes and the Workflow Administrator.

Topic	Page
Overview .....	16-2
Managing Workflow Security .....	16-3
Workflow Iteration .....	16-8
Using the Workflow Process Editor .....	16-8
Importing and Exporting Workflow Templates .....	16-39
Process Manager Toolbar Access Control .....	16-41
Viewing Workflow History .....	16-42
Workflow Instance States .....	16-49
Domain-based Workflows .....	16-50
Out-of-the-Box Workflow Templates .....	16-51
Workflow Template Execution Flags .....	16-60
Using Task Form Templates in a Workflow .....	16-71
Electronic Signatures .....	16-72
Best Practices .....	16-75

## Overview

A *workflow system* gives you the ability to automate procedures in which information, tasks, and documents are passed among participants. This procedure is based on a process composed of well-defined rules, designed to efficiently accomplish your business goals.

This chapter provides information on creating and editing workflow templates, importing and exporting templates, and viewing workflow history. See the end of this chapter for detailed information concerning configuring worklist fields and object subscription code.

The Windchill workflow system, implemented by the Windchill Workflow application, consists of four components:

- The Workflow Process Editor, which allows you to define a workflow process and save your definition as a process template. This graphical editor and its use are the focus of this chapter.
- The workflow runtime system, which executes a defined workflow process within the context of a specific business object (for example, a part or a document). Process execution includes delivering work assignments to users participating in the process, opening applications (for example, automatically interacting with the Windchill Explorer to check a business object out of the database), initiating subprocesses, and so on. The way in which the end user interacts with a running workflow process is described in detail in the *Windchill User's Guide*.
- The Workflow Process Manager, a graphical tool for monitoring and reporting on workflow processes. (For access control information, see [Process Manager Toolbar Access Control](#), in this chapter.)
- The Workflow History Viewer, which provides a simple ASCII interface used to access recorded workflow events, such as state changes, data transfers, or process start. The information in [Viewing Workflow History](#) can assist you in optimizing or streamlining a workflow process.

## Accessing the Workflow Administrator Utility

You can access the **Workflow Administrator** on the **Utilities** page from the **Site**, **Organization**, **Product**, and **Library** tabs. Site, organization, and application context administrators manage workflow templates, as described below:

- The **Site** tab provides the site administrator with unrestricted access to all workflow templates in the site context.

Site administrators create, modify, delete, and view workflow templates in the site context.

- The **Organization** tab provides access to the workflow templates in the site context and the templates that are within the active organization context.

Organization administrators create, modify, delete, and view workflow templates in the given organization context. Organization administrators can view workflow templates from the site text.

- The **Product** and **Library** tab provides access to the workflow templates that are in the active context, plus access to the templates that are inherited from the organization and site contexts.

Product and library administrators create, modify, delete, and view workflow templates in their respective context. They can view organization templates from the parent organization context and the site context.

**Note:** The Workflow Administrator is not available on the Project or Program tab to administrators of project or program contexts. For more information about the specific workflow process support in Windchill ProjectLink, see the [Workflow Process Support in Windchill ProjectLink](#) section.

The Workflow Administrator client displays a table that lists all workflow templates belonging to the given context, plus those belonging to its parent contexts. The Context column in the table identifies the context owning each workflow template.

When you create a workflow process in Windchill PDMLink, the system saves the new workflow process in the System cabinet of the context in which it is created. The system, not the user, decides where the new workflow template is to be located.

## Managing Workflow Security

Workflow creators are permitted to write workflow-embedded Java code to facilitate the execution of the workflow process. This embedded Java code is executed on the server, and there are no restrictions on the APIs available for use.

Considering this capability, an additional level of control has been provided for the site administrator to prevent a user who is not a member of at least one of three specific site context groups (Administrators, Workflow Administrators, or Workflow Authors) from embedding Java code in workflows.

A user with permissions to create workflow templates (for example, a project manager) could potentially add malicious code in one of the workflow expressions, causing a possible security threat. For this reason, workflow templates that contain Java expressions must be written, reviewed, and thoroughly tested by individuals that are trusted by the organization.

The **Workflow Administrator** link is available on the Site, Organization, Library, and Product Utilities pages. Although it is not available on the **Project Utilities** tab, a Project Manager could potentially obtain the URL for and access the user interface by typing it directly into a browser address bar.

The following sections provide more detail about the roles that can author workflow processes, the site context groups that allow Java code to be embedded, and the areas of the user interface that are disabled when a user is prevented from embedding Java code.

## Workflow Creators

Windchill PDMLink and Windchill ProjectLink permit all users with the role of Library, Product, and Project Managers, as well as Organization and Site Administrators to author (create and modify) workflows within the contexts (Organizations, Products, Libraries, or Projects) they manage. This is as-designed, and is a necessary component of the distributed administration model enabled by these solutions.

The embedded Java code that workflow creators are permitted to write to a workflow process includes embedding Java code in Execute Expression robots, Synchronize robots, Conditional gates and activities to perform conditional branching or execute custom application logic within the context of the workflow. This embedded Java code is executed on the server, and there are no restrictions on the APIs.

Application context managers for Libraries, Products, Projects, Programs, and Organizations are granted "Full Control (All)" permission on the WXObject class. This permits Organization Administrators and application context managers to effectively administer their contexts, through granting them the ability to modify access control policy rules, workflow and life cycle templates, and other contextual administrative objects. However, through granting them the ability to author workflow templates they are also effectively being granted the ability to write Java expressions which will be run on the server when the workflow is executed.

With the introduction and adoption of the distributed administration model, the number of users with broad access permissions in the application contexts expanded significantly. PTC is no longer assuming that a user who is responsible for administering an individual context is trusted to author Java code which may provide access to information outside of the context they manage.



Considering this capability, a user with permissions to create workflow templates (e.g., Project Managers) could potentially add malicious code in one of the expressions, causing a possible security threat. For this reason, workflow templates that contain Java expressions must be created and thoroughly tested by individuals that are trusted by the organization.

## Restricting Workflow-Embedded Java Code

An additional level of control has been provided for the site administrator to restrict which users can create and modify Java expressions within workflow templates. This is done by specifying membership in at least one of three of the following site context groups:

- Administrators
- Workflow Administrators
- Workflow Authors

A user must be part of one of the above listed groups to have permission to embed Java code in workflows templates.

When a user attempts to add or modify expression code, the system will perform an additional check to verify that the user is a member of at least one of these three groups defined in the Site context. If the user is not a member of at least one of these three groups, all expression code input fields in the Workflow Process Editor user interface will be disabled. Furthermore, the workflow service will prevent saving a template with modified expression code, in cases where the workflow template may be created through other means.

## Administrative Groups

The Administrators and Workflow Administrators groups are part of the base data loaded with every Windchill installation. However, members of these groups are also granted the permissions necessary to create and modify Workflow Templates within the Site context. Even though these permissions are not inherited by the domains used for organizations or other contexts, a member of one of these groups could potentially obtain the URL for and access the Site-level Workflow Administrator user interface by typing it directly into a browser address bar.

For this reason, a new group is available in the Site context called **Workflow Authors**. This group will not be granted any access permissions by default, but members of this group will be considered trusted to author Java expressions in locations where they are otherwise granted the permission necessary to author workflow templates.

For example, if a library context manager launches the Workflow Administrator user interface and chooses to create a new workflow template in the context of that library, they will be permitted to do so because they have been granted the necessary access control permissions through their membership in the library manager role (full control on WTOBJECT). However, when the library context manager attempts to access any of the input fields for expression code (for example, the **Transitions** tab for a workflow activity) in the workflow template, the system will perform an additional check to verify that the user is trusted to author expression code by confirming that the user is a member of either the Workflow Authors, Workflow Administrators, or (Site) Administrators groups. If so, the expression code input fields will be enabled and the user will be able to edit the embedded expression code. If not, the expression code input fields will be disabled and the user will be able to view but not edit the expression code.

Thus, membership in one or more of these three groups serves as a second layer of permission controlling who is able to create and edit workflow-embedded Java expressions.

### Disabled Areas of the User Interface

The following areas within the Workflow Administrator user interface will be disabled for users that do not have permission to create or edit Java expressions:

- Import button
- Export button
- Save As button

The text area used to enter Java expressions in the Workflow Process Editor will be disabled for the following components (for detailed information about these components, refer to the Workflow online help):

- Routing tab in the Assigned Activity / Ad hoc Activity process node
- Transition tab in the Assigned Activity / Ad-hoc Activity process node
- Routing tab from the Process Properties link
- Transition tab from the Process Properties link
- Expression Robot
- Block
- All connectors (i.e., And connector, Or connector, Threshold connector, and the Conditional router)

The following components will also be disabled in the Workflow Process Editor:

- Synch Robot
- Application Robot
- Expression Robot

No restrictions have been put in place when users create context templates (e.g., project, product) that load workflow templates. Context Creators will be allowed to load the workflow templates containing expressions, even if they are not members of the site level workflow administrators group.

## Enabling Restricted Access to Workflow Administrators Group

There is a property in the wt.properties file called wt.workflow.definer.checkAdminAccess. The default is false. That means you can access the workflow clients through the URL alias. If you want to restrict access to the Workflow Administrator, perform the following steps:

1. Set wt.workflow.definer.checkAdminAccess to true.
2. Edit the httpd.conf file to require authentication when accessing Web pages under /windchill/wt/clients/workflow (see below).
3. Restart the method server.

Any user attempting to access the **Workflow Administrator** page is then required to authenticate as a member of the Workflow Administrators group. If the authentication fails, an error page indicates that they are not authorized to access that page.

An example of setting the web server (Apache) to authenticate access to workflow clients is as follows. This change is made to the apache/conf/httpd.conf file.

```
# Admin level authentication
<Location /windchill/wt/clients/workflow/>
    AuthName Windchill
    AuthType Basic
    <IfModule auth_ldap.c>
        AuthLDAPAuthoritative off
        AuthLDAPURL
            ldap://ldap.company.com:389/ou=People,ou=wc62,l=city,o=company
    </IfModule>
    AuthUserFile "C:/Program Files/Apache Group/Apache/conf/wtpasswd"
    require valid-user
</Location>
```

*Italicized items are installation-dependent.*

## Workflow Iteration

Administering workflow templates is an iterative process. Like version-controlled objects, iterated objects are checked in and out of shared locations. However, unlike version-controlled objects, they cannot be revised. Instead, any change to an object creates a new and separate iteration when it is checked in. Earlier iterations, which may still be in use, are unchanged and unaffected by the new iteration. Only the latest iteration is available for new uses.

To make changes to a workflow template, you must check out a copy. (Clicking **Edit** on the Workflow Administrator page automatically checks out a copy of the selected template.) While it is checked out, no one else can check out a copy, but processes can still be initiated based on the current template. When you have completed changes to the checked-out copy, you must save it and check it in to make it available to others. It then becomes the latest iteration. Running processes that use an earlier iteration continue to run, unaffected by the newer iteration.

## Testing an Edited Workflow Process Template

Under normal circumstances, when you initiate a process template, the latest iteration in the shared location is used; however, if you have the template checked out and stored in your personal cabinet, the working copy is used. This makes it possible to test a workflow process template before checking it in.

To edit and test a workflow process template, follow this procedure:

1. Check out a copy of a workflow template.
2. Edit the copy, and save it to your personal cabinet.
3. Initiate a workflow process, based on that template.

The edited copy of the template in your personal cabinet is then used, rather than the current, checked-out iteration in the System cabinet.

You must either complete the running process, or stop and delete it before you can check in the edited workflow process template or undo the check out.

If you attempt to check in the edited process template or undo the checkout while the process instance is running, the following error message is displayed:

```
Can't modify or delete template, <template name>; there are open instances.
```

## Using the Workflow Process Editor

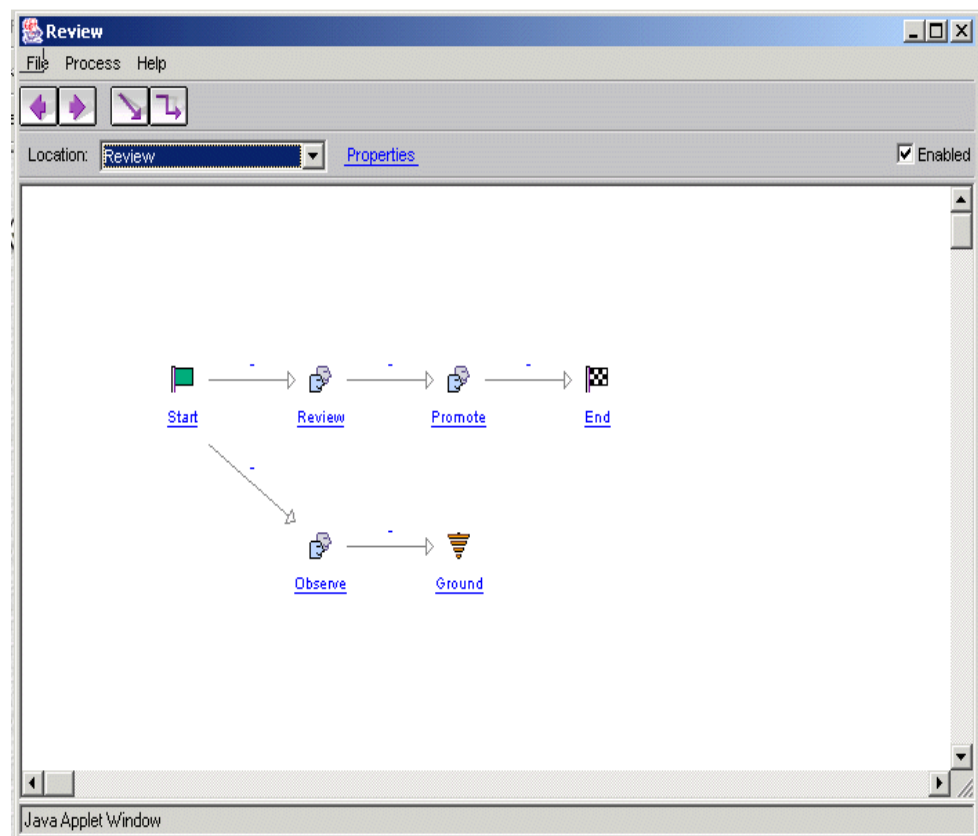
The Workflow Process Editor is a graphical interface for defining workflow processes that range from the simple to the highly complex. It features a large set of predefined activity nodes that you can place and connect. The Process Editor supports nested processes, branching, merging, loops for iterative activities, and defining assigned activities.

## Working with Workflow Templates

The **Workflow Administrator** page displays a list of existing workflow templates, with their locations enabled status, and context. Using the buttons on this page, you can create new templates, edit and view existing templates, and delete templates, as well other activities, including importing and exporting workflows. Click **New** or **Edit** to access the Workflow Process Editor. When you edit a workflow process, it is automatically checked out if it is in a vault.

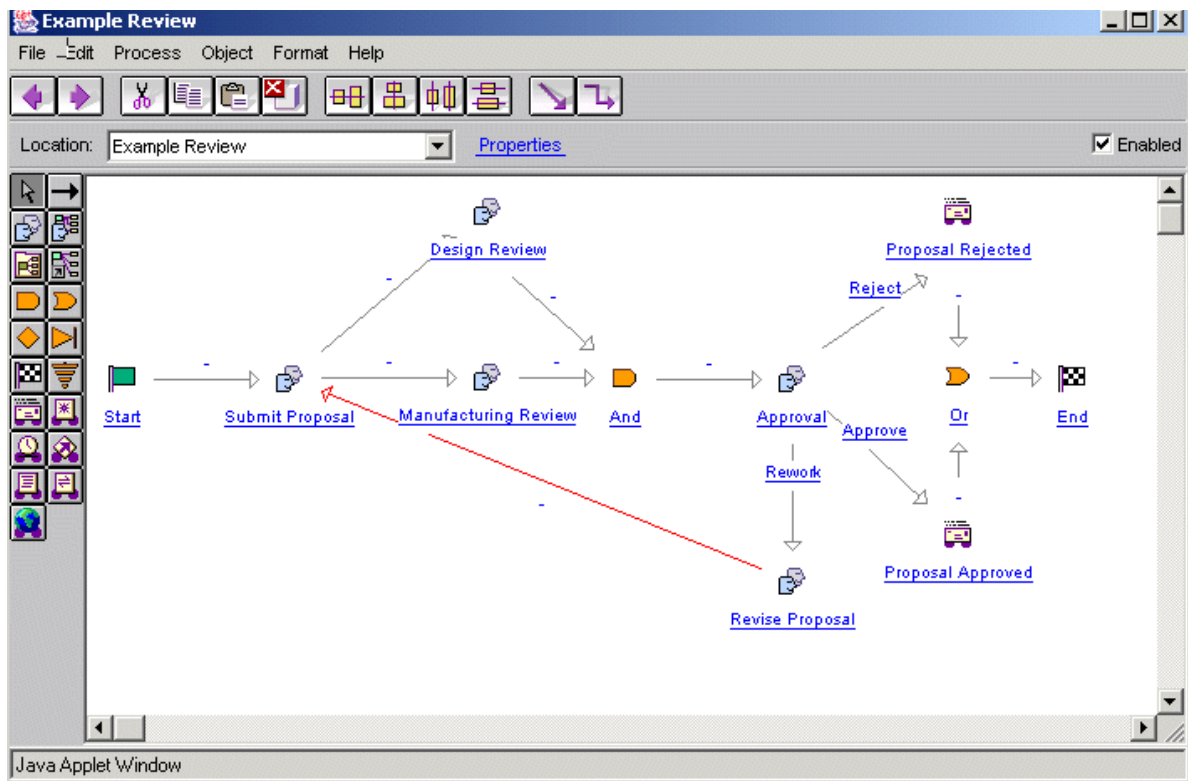
For detailed help on using the **Workflow Administrator**, click **Help**.

The following figure shows the Workflow Process Editor displaying a Review process, (one of the default workflow templates included in an out-of-the-box system):

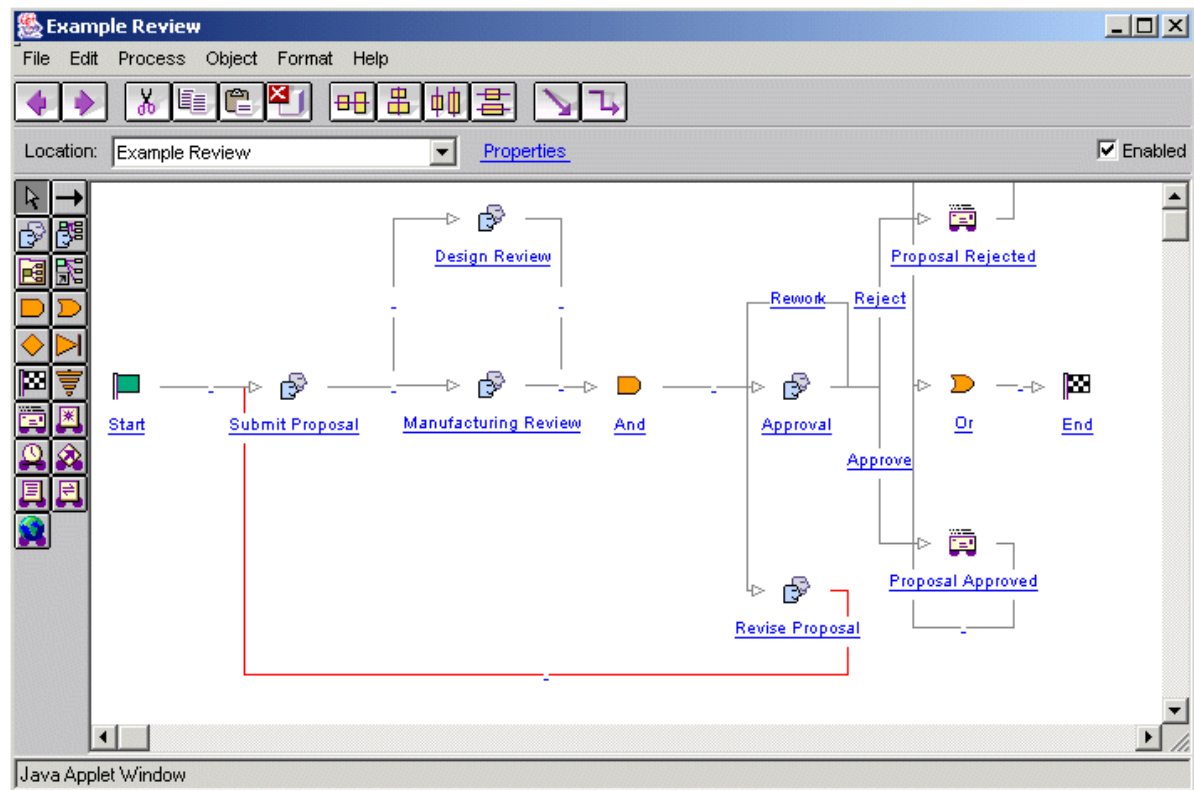


If you click **New**, the Workflow Process Editor displays only the Start node.

You can display the workflow process with *straight lines* at any angle or with *square lines* – horizontal and vertical lines and right angles. Click **Straight Lines** to use straight lines at any angle, as in the following figure:



Click **Square Lines** to use horizontal and vertical lines at right angles, as in the following figure:



To verify that your process definition is correct, select **Validate All** from the **Process** menu. The **Validate** window either confirms the process definition or identifies dangling activities or malformed processes.

A Java compiler is integrated with the Process Editor to support expressions of arbitrary complexity. Workflow routing functionality includes links and event triggers. Events that cause a link to *fire* are displayed on the link itself, so anyone viewing the process definition can easily understand and verify the process behavior. For example, the Approve and Revise events in the previous straight line and square line examples are events that cause a link to fire.

When you have completed your process definition, it is saved in your personal folder. To change a process definition that has been saved in a vault, it must be first checked out, and then checked in.

Updating workflow processes is an iterative process. A new iteration is created when an edit is checked in. You can view iterations on the **Workflow Administrator** page.

The following sections describe the tools and components available to help you define a workflow process in Windchill.

## Navigating a Process Diagram

The Workflow Process Editor is designed for easy navigation of processes and their subprocesses, using common Web navigation techniques. For example, you can edit a subprocess diagram, by clicking a subprocess hyperlink.

To navigate between a parent process and a subprocess, use either the **Back** and **Forward** buttons, or the drop-down list in the **Location** field.

The title bar of the Workflow Process Editor displays the name of the process or subprocess you are currently editing.

Hyperlinks display the properties of each activity type and link, for example:

- Click an activity node hyperlink to open the properties window for that activity. You can then create, edit, or view the properties that define the node's behavior.
- Click the Properties hyperlink (at the right of the **Location** drop-down menu) to view and edit the properties of the process itself.
- Double-click the link that connects a node to open a window to map events that are *broadcast* (or *emitted*) from the preceding activity to actions in the succeeding activity. By default, the completion event for a given task triggers the start of all successor tasks.

## Placing Process Nodes

You can build a process definition by adding, selecting, and linking nodes that are represented by icons at the left of the workflow process editor.

- Add nodes to the process definition by selecting the appropriate icon and then click on open space within the process diagram.
- Select a node and display its properties by clicking the node hyperlink.
- Link two nodes by clicking the **Action** icon (a right arrow). Click and drag from the first node to the second node and release the mouse button. (A line with an arrow appears linking the nodes. The first node does not move to the second.)
- When deleting workflow objects like assignments, use the supported workflow helper APIs in the workflow expressions to avoid data corruption. Never delete the object directly from the expression using the persistence manager APIs.



The following list describes the process nodes that can be added to your process definition. The list is displayed in order of each icon's appearance on the Process Editor.

- The *Assigned Activity* is an activity assigned to one or more users or user-defined groups or an actor to perform.
- The *Ad Hoc Activity* is assigned to a user to define a group of activities at runtime. The group of activities is similar to a simple block.
- The *Block* represents a group of activities, connectors, or robots. You can reduce the complexity of a process by creating blocks of activities that can be expanded when needed.
- The *Proxy Process* is a subprocess embedded within the main parent process, which can be nested to reduce complexity and provide reuse.
- The *And Connector* fires when all the predecessor links have fired, but not before.
- The *Or Connector* fires when any one of the predecessor links has fired. Preceding activities are terminated if **Terminate Open Predecessor Activities when Fired** has been selected.
- The *Conditional Router* allows you to branch a process based on a conditional expression.
- The *Threshold Connector* fires when a user-defined number of predecessor links have fired. Preceding activities are terminated if **Terminate Open Predecessor Activities when Fired** has been selected.
- The *End* stops the process. All process activities should eventually be connected to an end.
- The *Ground* stops a parallel branch of activities within the process, but it does not stop the process.

- The *Notification Robot* notifies the appropriate user with a user-defined e-mail. You can define notification robots containing an attachment or metadata of the primary business object associated with the workflow. For a detailed procedure, see the online help. You can use braces to delimit variables created for the process or node, for example, {varname}. Use back slashes to escape the delimiter, for example, \{{varname}}\.
- The *Method Robot* represents one of several single actions performed when adding the robot to the process. No other configuration is required. The following table lists the robot actions:

Robot	Description
Checkin	Checks in the primary business object to the Windchill database.
Checkout	Checks out a business object to the specified user. For example, you can use the Checkout robot to automatically check a part out to the engineer assigned the task of applying changes after a design review cycle is complete.
Demote	Causes the primary business object to transition to a predecessor phase, with an associated state change, and the application of new business rules (such as those for access control).
Deny	Removes the primary business object from the gate and returns it to the phase of the same state.
Drop	Causes an object to be removed from its current life cycle and sets its state to <i>dropped</i> . For example, you could have a process branch in which two vendors submit bids for review. These bids could be entered into the database as Windchill documents, which would move through a review and approval process by application of a process definition. In this case, your process may require that, when one bid is approved, its document object is automatically promoted to its next life cycle phase, while the document containing the rejected bid is dropped from its life cycle and goes no further.

Robot	Description
Promote	Causes the primary business object to transition to a successor phase, with an associated state change and the application of new business rules, such as those for access control. For example, you could define a process in which an object is automatically promoted to the next phase in its life cycle, if a specific user approves the object. In this case, you could add the Promote robot to your process definition to execute all of the actions associated with an object's promotion.
Set State	Sets a Life Cycle–managed object to an ordinal state or a specific state. The ordinal state is entered as any non-zero integer. The specific state is selected from those defined in the <code>wt.lifecycle.StateRB</code> enumerated type.
Submit	Moves the business object associated with this process to the gate for its current life cycle phase. After a submit, an object awaits promotion to the next life cycle phase. For example, you could add the Submit robot to a process definition to indicate that, when a user creates a change request, it is automatically submitted for promotion to the Open state.

- The *Timer Robot* delays the start of an activity by a specified amount of time, based on the time it is fired or the time the parent process is begun.
- The *Launch Application Robot* executes system commands on the server. These commands are executed using the Java `runtime.exe` command. The execution can be either synchronous or asynchronous.
- The *Execute Expression Robot* enters a synchronous Java expression to be executed in a workflow. By default, the expression returns `true`. A return of `false` indicates a problem during execution, and an exception is thrown on the server.
- The *Synchronization Robot* synchronizes the start of an activity or process with events that are not time related. You can set the robot to start an activity when certain generic external or Windchill-keyed events occur.

If synchronization does not work, make sure the Windchill time is set to the correct time zone. See the Managing User Preferences section of the *Windchill System Administrator's Guide* for more information.

- The *URL Robot* executes a URL to communicate with another server, for such purposes as initiating various Info\*Engine tasks or providing information necessary to complete workflow tasks. It can initiate an operation or retrieve status information to be collected in a string variable. HTML links to binary objects, such as graphics, can be retrieved, although the objects, themselves, cannot.

You can specify the results of a failure by the robot to execute the URL. The following list of error codes may be helpful. For instructions, see the help file.

- *400 Bad Request*: Request has not been recognized by the server, because of incorrect syntax. The client should not repeat the request.
- *401 Unauthorized*: Request requires user authentication. Under normal usage, the URL robot does not support authentication. Request should not be repeated.
- *403 Forbidden*: Request has been recognized but the server has refused to honor it. Authentication was not the reason. The request should not be repeated.
- *404 Not Found*: Server has found no match for the Request-URI. May be temporary or permanent. Repeat of request may be appropriate.
- *500 Server Error*: Server has encountered an unexpected condition, which prevents it from fulfilling the request. Repeat of request may be appropriate.
- *501 Not Implemented/Internal Error*: Server does not support the functionality required to fulfill the request. Request should not be repeated.
- *503 Service Unavailable*: Server is temporarily unable to handle the request. Repeat of request may be appropriate.
- *504 Gateway Timeout*: Server has not received a timely response from the upstream server specified by the URI. Repeat of request may be appropriate.

For more information on error messages, see the Internet standards at W3C HTTP RFC (<http://www.w3.org>).

## Declaring Variables

When you define a process, variables can be used within transition condition or automatic routing expressions. Variables can be either global (applicable to the process itself) or local (applicable to an assigned activity or a subprocess).

**Note:** When the Windchill 8.0 out-of-the-box change management workflows are run in Windchill 9.0, the user will see the activity variables in the task details page that are defined in the 8.0 workflow templates, instead of the activity variables that are defined in the 9.0 workflow templates.

Variables can be declared as any Java type or as any Windchill class. The only restriction is that the variable must be Serializable. If the variable is typed as a Windchill business object, attributes of that object can be referenced through standard getter APIs. Variables can be declared as follows:

- Visible or invisible
- Required or optional
- Read only or read/write
- Resettable or Static

Variable values can be initialized from parent process variables when an activity or subprocess is started and can also be copied into parent process variables when the activity or subprocess is complete.

**Note:** Although you can declare a workflow variable to be any Java type or Windchill class (provided it is Serializable), PTC encourages you to consider the following guidelines:

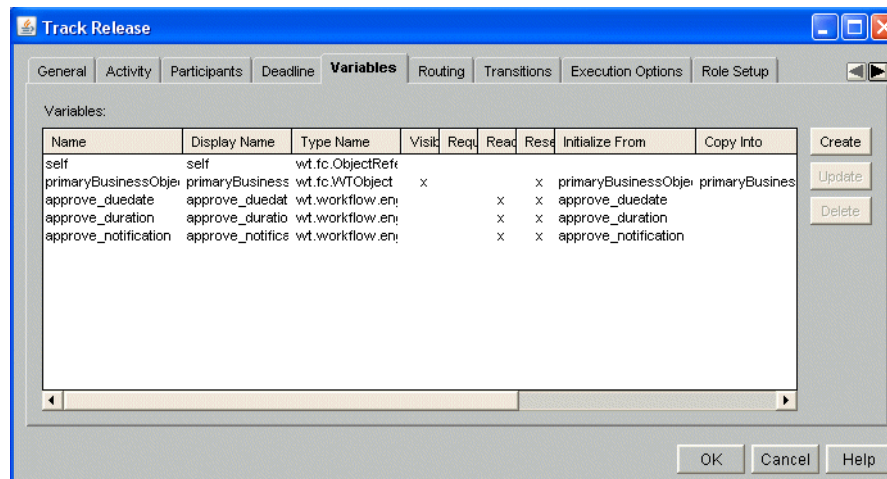
- Workflow variables are composited within a ProcessData object, which in turn, is persisted in the Oracle database in a BLOB column. If the persistent signature of a BLOB object changes after it is stored in the database, an error may be thrown at the time it is retrieved from the database and de-serialized; therefore, PTC recommends that Workflow Variable types be restricted to the following:
  - java primitives
  - Serializables with stable persistent signatures
  - Evolvables
  - Persistables

Evolvables provide methods to handle changes in their persistent signatures; and Persistables are stored in workflow variables as ObjectReferences (not the fully inflated object).

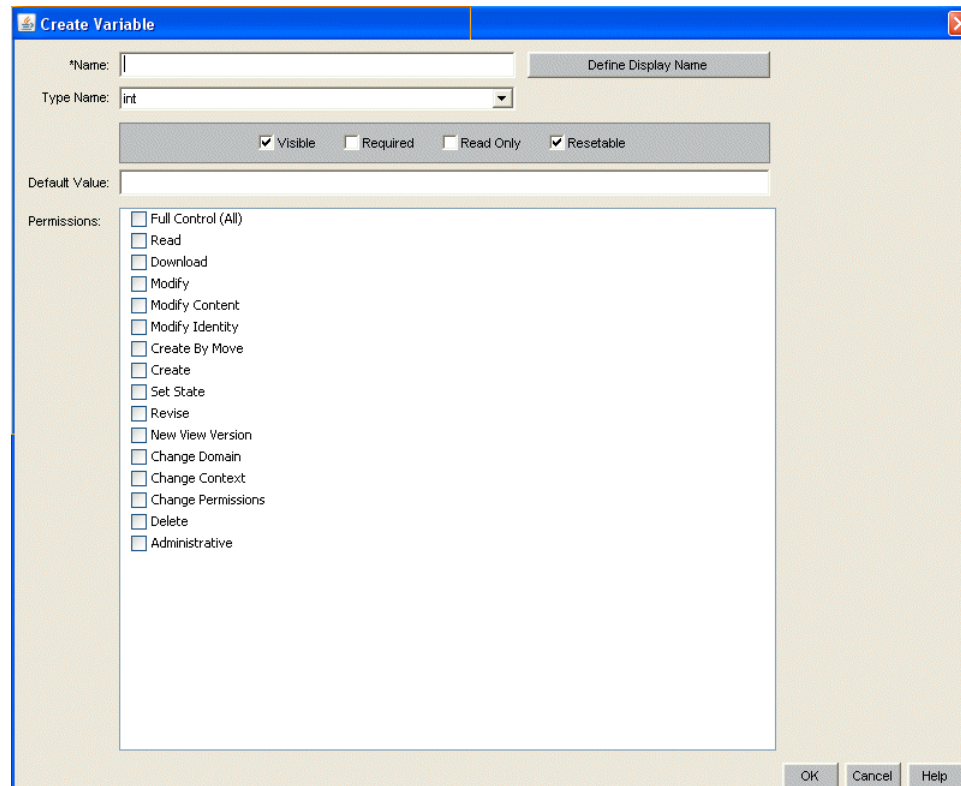
- Persistables should not be composited within another class as this may lead to data inconsistency or corruption. Doing this will also lead to a `wt.pds.NotEvolvableException` during migration.
- Persistables may be an element in a `java.util.Vector` (this is because the `Vector` class receives special handling by the `WfVariable` class), but this should be avoided if possible. Use of a `CachedObjectReference` or `ObjectReference` pointing to a `Persistable` is the preferred practice.
- A Workflow Variable can contain, composite, or otherwise encapsulate a `ObjectReference` to a `Persistable` as necessary.

For information on the Windchill Evolvable classes and interfaces, see the *Windchill Customizer's Guide*.

The following figure shows the **Variables** tab panel on the properties window for an assigned activity named Activity 1.



To define additional activity variables, click **Create**. The **Create Variables** window opens:



To define a display name to be displayed for the variable, click **Define Display Name**.

The online help file for this window provides detailed instructions for variable declaration.

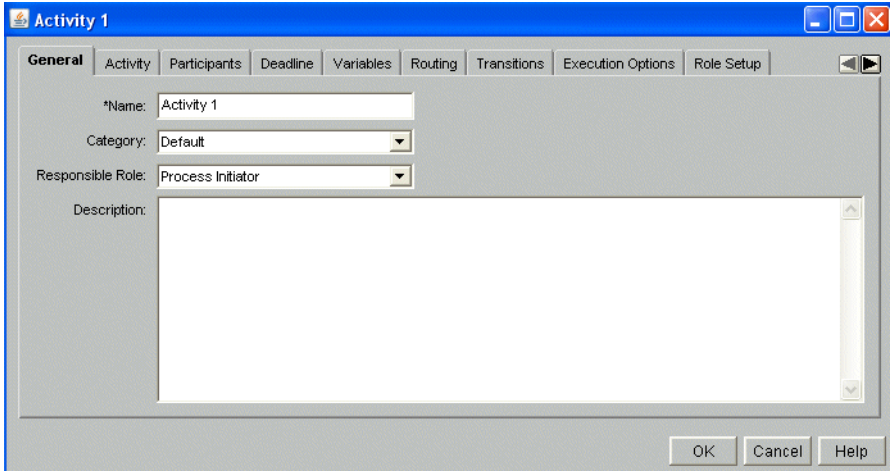
## Defining an Assigned Activity

An assigned activity is a task that gets assigned to a specific user or group of users when an instance of this process definition is running. When you define an assigned activity, you specify a task that the selected user is to perform as part of a workflow process. The task is then added to the user's Assignments table. See the appropriate user's guide for more information about the Assignments table.

The assigned activity properties are described in the sections that follow.

### General Tab

The following figure shows the properties window for an assigned activity, with the **General** tab active:

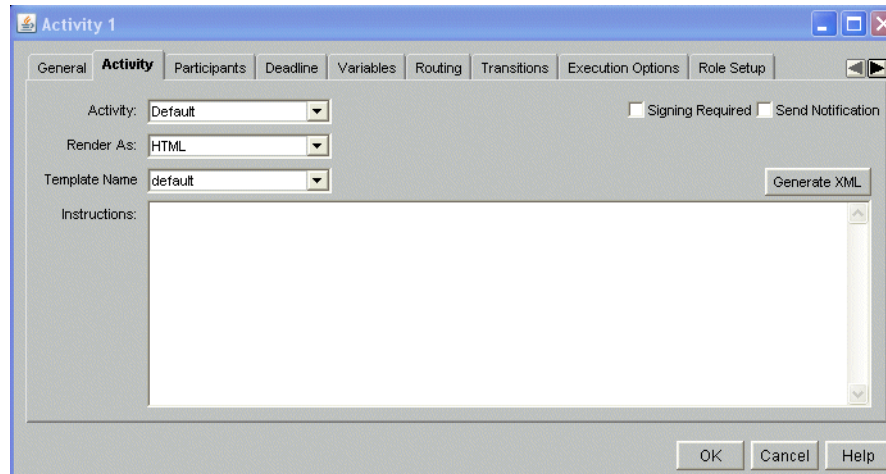
The image shows a software window titled "Activity 1". It has a tabbed interface with the "General" tab selected. Other tabs include "Activity", "Participants", "Deadline", "Variables", "Routing", "Transitions", "Execution Options", and "Role Setup". In the "General" tab, there are four fields: "\*Name:" with the value "Activity 1", "Category:" with a dropdown menu showing "Default", "Responsible Role:" with a dropdown menu showing "Process Initiator", and "Description:" with a large empty text area. At the bottom right of the window are three buttons: "OK", "Cancel", and "Help".

On the **General** tab, shown above, you can specify a name, a category, a responsible role, and a description for the assigned activity. Name is the only required property.

From the **Category** drop-down list you can categorize the activity. For example, you can have activity categories that reflect a team or product type. Windchill provides several predefined categories. Categories are an enumerated type, and you can define additional categories in the `wt.workflow.definer.WfTemplateCategory` file. For more information about enumerated types, see the *Windchill Customizer's Guide*.

## Activity Tab

On the **Activity** tab, shown below, you can specify the type of task that will be assigned to a user or user-defined group, or an actor to perform.



The screenshot shows a dialog box titled "Activity 1" with a tabbed interface. The "Activity" tab is selected. It contains the following fields and controls:

- Activity:** A dropdown menu with "Default" selected.
- Render As:** A dropdown menu with "HTML" selected.
- Template Name:** A dropdown menu with "default" selected.
- Instructions:** A large text area for entering instructions.
- Buttons:** "OK", "Cancel", and "Help" at the bottom right.
- Checkboxes:** "Signing Required" and "Send Notification" (both unchecked).
- Generate XML:** A button next to the Template Name dropdown.

From the **Activity** drop-down list, select a task type from the following options:

Activity	Description
Default	The default activity type.
Update Content	The task associated with this type of activity allows the user to upload attachments to the business object.
Set Up Participants	<p>The task associated with this type of activity allows the user the capability to modify the team role membership present in the workflow process template. Participants of a role can be added or removed depending on the permissions available to the user.</p> <p>For more information, see the <a href="#">Set Up Participants Activity Type</a>.</p>



Activity	Description
Review	<p>The task associated with this type of activity presents voting options to the user. The options include approve or do not approve. The users voting record is visible when displaying the life cycle history of the object from the details page.</p> <p><b>Note:</b> The Review and Promote activity types for life cycle-managed objects are associated with voting options. These voting options are different from the routing options defined on assigned activities of the workflow process templates. The voting options of the Review and Promote activity types are defined and displayed in the task details page of an activity in both the Windchill PDMLink and Windchill ProjectLink task details page.</p>
Promote	<p>The task associated with this type of activity presents voting options to the user. The options include:</p> <ul style="list-style-type: none"> <li>Promote will cause the object to advance to the next life cycle phase.</li> <li>Deny removes the primary business object from the gate and returns it to the phase of the same state.</li> <li>Demote will cause the object to get demoted to the previous life cycle state.</li> <li>Drop will cause the object to get removed from the life cycle so that it cannot proceed further.</li> </ul>
Observe	<p>The task associated with this type of activity allows the user to update only the business object depending on the permissions available to the user on that object.</p>
Submit	<p>The task associated with this type of activity allows the user to present the business object to the life cycle gate.</p>
Define Teams	<p>The task associated with this type of activity allows the user to select the values of activity variables of the type wt.team.Team and wt.team.TeamTemplate. A drop-down list for team and team templates is presented to the user to select the variable types.</p> <p>For more information, see <a href="#">Define Teams Activity Type</a>.</p>

From the **Render As** drop-down list you can determine how you would like the task details to be rendered by selecting one of the following options:

- HTML - (default) renders the task details page in an HTML browser page.
- PDF - renders the task details page as a PDF using the Adobe Acrobat Reader. Adobe Designer is required to render tasks in PDF format. For more information, see the [Using Task Form Templates in a Workflow](#) section.

The PDF option is not available with an ad hoc activity.

Select the name of the task template from the **Template Name** drop-down list that should be used to render the task details page. For more information about managing templates, see the Templates online help.

If desired, provide instructions in the **Instructions** text area. You can use braces to delimit variables, for example, {varname}. Use back slashes to escape the delimiter, as shown in the following example:

```
\{{varname}}\
```

If you want users to be notified by e-mail of the task assignment, check **Send Notification**. E-mail notification is optional, as tasks are always added to the appropriate user's Assignment table.

Select the **Signing Required** check box if you want an electronic signature required for the activity to be complete. For more information about electronic signatures, see [Requiring Electronic Signatures in a Workflow](#).

### Set Up Participants Activity Type

When the specific participants that are to complete the tasks are unknown at the time a workflow template is created, you can define a *set up participants activity* by selecting **Set Up Participants** in the **Activity** drop-down list. This activity type gives a specifically designated user the permission to change the roles on the Set Up Participants table located in the Assignments table on the Home tab.

When defining a set up participants activity, you must also define a role permission map on the [Role Setup Tab](#). The pool of users should first be selected on the [Resource Pool Tab](#). For detailed procedures for creating a Set Up Participants Activity, see the Workflow Administrator online help.

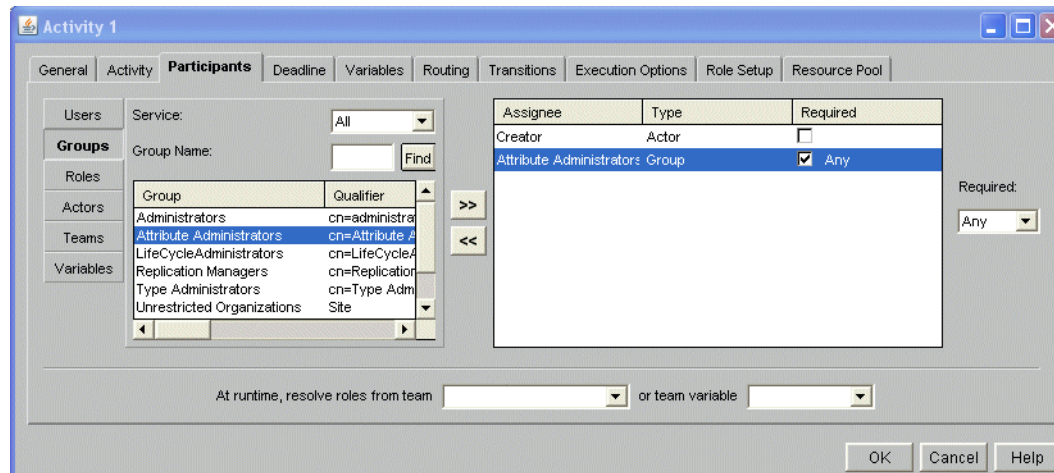
### Define Teams Activity Type

Create a *define teams* activity by selecting **Define Teams** from the **Activity** drop-down list. The workflow activity must also have a wt.team. Team variable created for use by this activity. Generally, you can do this by creating a process level wt.team.Team variable and copy to assign the process level variable to the wt.team.Team variable defined in the activity.

When the workflow is assigned, you can create a new instance of a team that gets assigned to the wt.team.Team variable you created. You must determine the usage of the variable when it is assigned the instance of the Team object.

## Participants Tab

From the **Participants** tab, shown below, you can assign users, user-defined groups, roles, actors, teams, or variables to complete the assigned activity:



Although you have the option to assign an activity task to actual users or user-defined groups as part of the process definition, you may find it more useful to select actors, roles, or teams as participants, which are then mapped to users or user-defined groups when an instance of the process definition is instantiated. Assigning participants in this way provides more flexibility and promotes reuse of a process definition in a variety of contexts.

If you choose a role, such as Submitter, as the participant to whom the task will be assigned, the role is resolved at runtime by mapping it to a participant specified in a life cycle or a team (usually a team). For example, if the workflow process is applied to a specific document, the Submitter role in the relevant phase of the document's life cycle can be mapped to a Team Leader role in the team to which the document is assigned. The actual user to whom Team Leader is mapped then receives the task in his or her Assignments table when this activity is fired. For more information on role mapping, with an extensive example, see [Teams](#).

The selected assignees are displayed on a table, which specifies the participant types. You can designate whether a specific participant is required to complete the task or what conditions must be met to meet a requirement. (That is, any participant, all participants, or a specified number, must complete the activity.) The **Required** drop-down list is not displayed for users or actors or if the assignee is not required to complete the activity.

If a task is assigned to a user-defined group, the task appears in the worklists (or Assignment table) of all group members. (However, one group member can choose to accept the task on behalf of the group.) By default, the person creating the assigned activity is defined as the user who is assigned the task.

To find a specific user, you can search the entire system, or narrow your search to your local system or a specific federated services or user-defined group.

If you want to map a role to a team other than the one associated with the process, you can select the team from the drop-down list at the bottom of the panel. This exception applies only to this specific activity.

You can also map a task role to an actor, that is, someone who performs a specific action within the context of the business object. Currently, Creator is the only actor defined.

**Note:** Click **Help** on the **Participants** tab panel for a help file, which documents specific procedures for assigning participants.

## Deadline Tab

On the **Deadline** tab, shown below, you can set the time that the activity is due. You can set the deadline in relation to the start of the activity or the start of the parent process. If you define them both, the earliest deadline applies. You can also designate the consequences and be notified if the deadline is overdue.

Activity 1

General Activity Participants **Deadline** Variables Routing Transitions Execution Options Role Setup Resource Pool

Set deadline to  days,  hours, and  minutes from the start of the activity, or  
 days,  hours, and  minutes from the start of the parent process.

Overdue Consequences

☐ Skip ☐ Mark complete ☐ Reassign to the responsible role

Overdue Notifications

☒ Notify the selected roles:

Approver  
Approver - Level I  
Approver - Level II  
Auditor

☐ Notify the responsible role:

days,  hours, and  minutes before the deadline

☐ Notify the responsible role:

days,  hours, and  minutes after the deadline

OK Cancel Help

To designate the consequences of a missed deadline, select one of the following check boxes:

- Skip
- Mark complete
- Reassign to the responsible role

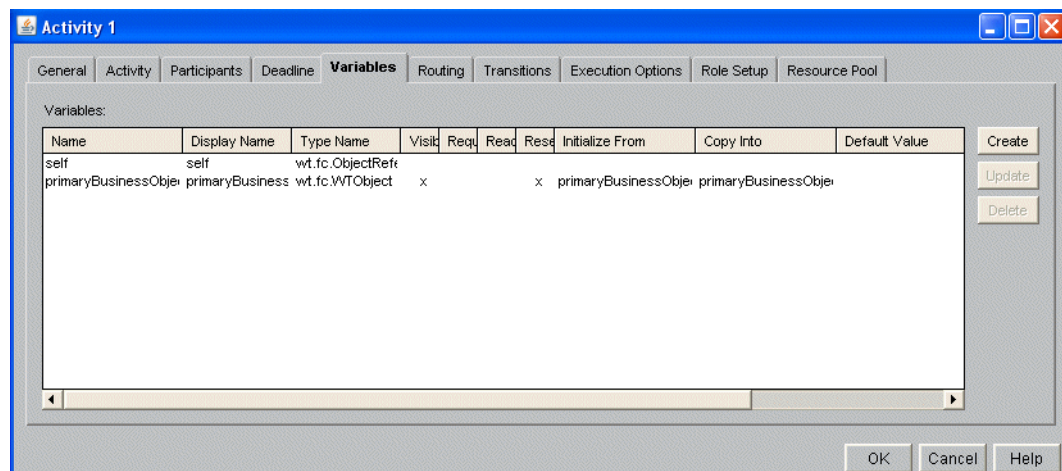
To designate who is notified and when, select any number of the following check boxes:

- To notify the user assigned to a specific role (in addition to the responsible role) when the activity is overdue, select the check box labeled **Notify the selected roles**. Then select any number of roles.
- To notify the responsible role before the activity deadline, select the upper check box labeled **Notify the responsible role** and fill in the amount of days, hours, and/or minutes before the deadline that you want notification sent.
- To notify the responsible role after the activity deadline, select the lower check box labeled **Notify the responsible role** and fill in the amount of days, hours, and/or minutes after the deadline that you want notification sent.

If you do not designate the time, notification is sent to the responsible role when the deadline is reached, even if you do not select a check box. If you select both responsible role notification check boxes and fill in times, notification is sent both before and after the deadline.

## Variables Tab

On the **Variables** tab, shown below, you can declare additional variables, or you can edit or delete existing variables. The variables defined for this assigned activity are listed in the **Variables** text area.



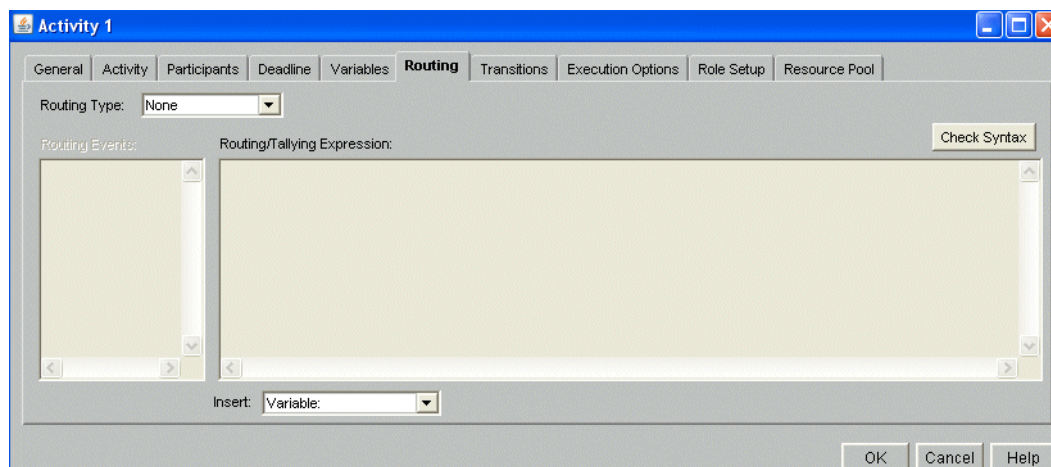
There are two default variables for assigned activities: *self* and *primaryBusinessObject*. The variable *self* refers to the assigned activity object at runtime. The variable *primaryBusinessObject* holds the business object associated with the workflow process at runtime.

See [Declaring Variables](#) for information about adding other variables in this tab panel.



## Routing Tab

On the **Routing** tab, you can specify custom routing events. These events are used to control the process flow by mapping one of the events you define, to an action that will be performed in a successor activity via a link.



**Note:** Windchill ProjectLink workflows present routing choices (either manual or exclusive) only to required participants. Non-required participants can see the task and all its information, but the task does not have check boxes or radio buttons or any of the associated text with the routing selection. The task complete button is available for all assignees.

Routing events are defined in the **Routing Events** text area, where you enter the name of the events (one event per line). Click **Check Syntax** to verify that the Java code you have entered is correct.

You can select any of the following routing types from the drop-down list:

- **None** for no routing. The **Routing Events** and **Routing Expression** text areas are not available.
- **Conditional** to automatically trigger the specified event.
- **Manual** to allow a user to select one or more of the routing events specified on this tab.
- **Manual exclusive** to allow a user to select one and only one of the routing events specified on this tab.

Conditional events require a firing expression to automatically fire the event. As shown in the following example, this expression is a fragment of Java code that sets a variable *result* to one of the custom routing events. The *result* variable is required and must return a string matching one of the event names. The expression can reference any variable defined in the **Variables** tab. If this information on the example were entered in the **Routing Events** panel, the >1000 event would be emitted if the result variable for the cost of the object associated with the process (for example, a change order) has a value greater than \$1000.

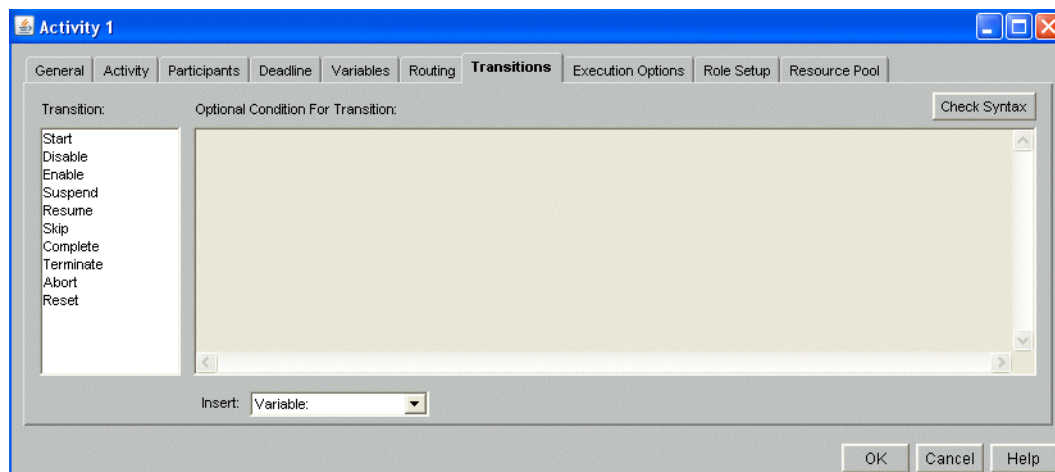
Likewise, the <1000 event would be emitted if the *result* variable has a value less than or equal to \$1000. One link coming from the activity can be configured to start another activity that would be assigned to a user who would review costs when the >1000 event was emitted (that is, when a change order required an expenditure greater than \$1000). Another link can be configured to simply continue with the process when the <1000 event is emitted.

If you do not specify an automatic event firing expression, the user who is assigned the task chooses a custom routing events. The activity emits the event that the user chooses.

**Note:** The online help file, accessed from the **Routing** tab panel, contains an extensive set of tally examples and a link to an HTML file that contains additional code samples.

## Transitions Tab

On the **Transitions** tab, shown below, you can define the conditions necessary for moving from one internal state to another within a workflow process. Each assigned activity defines transitions.



For example, initiation of a particular assigned activity represents a transition. A state transition can result from a routing decision made by the workflow process while it is running.

Each transition can have an associated condition. If this condition is TRUE, the transition succeeds. Otherwise, it does not. These conditions are defined in the **Transitions** tab panel.

To add a condition to a transition, select it from the **Transition** list, and type an expression in the condition text area. The condition is a standard Java expression that sets the *result* variable to TRUE if the transition should proceed, or to FALSE if it should not. Click **Check Syntax** to verify that the Java code you have entered is correct.

For example, you might want the process to start an assigned activity only if a variable (i) is set to a certain value. Therefore, you would select **Start** from the **Transition** list and enter the expression shown as the condition.

## Execution Options Tab

On the **Execution Options** tab, shown below, you can specify the consequences of an error, including who is notified.

The following table describes the check boxes you may want to select:

Check box	Description
<b>Notify the responsible role if there is an error</b>	Sends a notification to the responsible role identified for the activity in the case in which an unexpected error occurs in the activity instance. This value is checked by default and the initiator of the workflow process is defined as the user associated with the responsible role if no other role is identified.



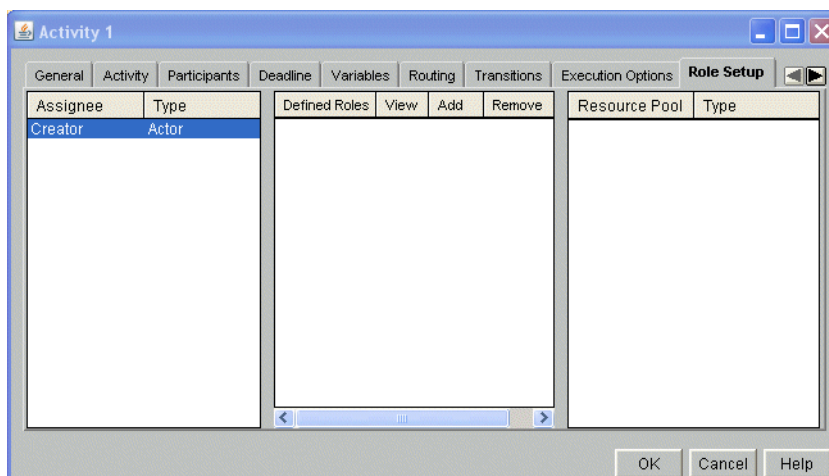
Check box	Description
<b>Notify the responsible role on abort</b>	<p>Sends a notification to the responsible role identified for the activity in the case in which an unexpected error occurs in the activity instance when the activity is aborted. This value is checked by default and the initiator of the workflow process is defined as the user associated with the responsible role if no other role is identified.</p> <p>If the activity is aborted, and there are following activities in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start.</p>
<b>Abort if there is an error</b>	<p>Automatically aborts the activity if an unexpected error occurs when the activity is executing. If the activity is aborted, and there are following activities in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start. This option is checked by default.</p>
<b>Abort the parent process on abort</b>	<p>Automatically aborts the parent process for an activity if an unexpected error occurs when the activity is executing. This option is checked by default.</p> <p>If the activity is aborted, and there are activities that follow in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start.</p>

Check box	Description
<b>Notify the responsible role on abort</b>	<p>Sends a notification to the responsible role identified for the activity in the case in which an unexpected error occurs in the activity instance when the activity is aborted. This value is checked by default and the initiator of the workflow process is defined as the user associated with the responsible role if no other role is identified.</p> <p>If the activity is aborted, and there are following activities in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start.</p>
<b>Abort if there is an error</b>	<p>Automatically aborts the activity if an unexpected error occurs when the activity is executing. If the activity is aborted, and there are following activities in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start. This option is checked by default.</p>
<b>Abort the parent process on abort</b>	<p>Automatically aborts the parent process for an activity if an unexpected error occurs when the activity is executing. This option is checked by default.</p> <p>If the activity is aborted, and there are activities that follow in the process that depend on the completion of this preceding activity, the following activities linked to this preceding activity will NOT start.</p>

Check box	Description
<b>Record variable changes</b>	Saves the variable values entered by an individual assignee in the activity. This applies only to variables that are displayed in an activity and are editable by an assignee for the activity. These values are captured during workflow execution for each activity instance and for every time the activity is executed.
<b>Record task reassignment</b>	Keeps a record of task reassignments associated with each task. When you select this option for an activity, a record of the original assignee, the new assignee, the reason for reassignment, and the reassignment date is saved. These values are captured during workflow execution for each activity instance and for every time the activity is executed.
<b>Record voting</b>	Keeps a record of routing votes made by each participant in workflow activity with manual routing choices. These values are captured during workflow execution for each activity instance and for every time the activity is executed.
<b>Ignore unresolved role</b>	Ignores non-required role participants if no users are mapped to that role in the associated team. If no users are mapped from the team, for a required role, that role participant is automatically mapped to the responsible role (by default, the user that created the business object to which the workflow process, via the life cycle, is associated).

## Role Setup Tab

On the **Role Setup** tab, shown below, you can specify the roles for assigned activities and the permissions that apply to the roles. These roles apply to the activity type that is selected on the **Activity** tab.



For example, a *set up participant* activity enables the activity assignee to modify the team role membership during workflow execution. The role permission map defined in the **Role Setup** tab defines which activity assignees will be able to alter which role memberships from the task details page (also referred to as the information page for the assignment).

The **Assignee** table, on the left side of the **Role Setup** tab, displays the list of participants who will be assigned the set up participants activity during process execution. This list is derived from the **Participants** tab for the activity.

The **Defined Roles** table, in the center of the **Role Setup** tab, displays a list of all of the roles used anywhere in the workflow template. At runtime, the roles that are listed in the **Defined Roles** table can be altered by the assignee in the details page of the assignment which is generated for this activity. The permissions for each role are defined by selecting **View**, **Add**, or **Remove** in the check boxes adjacent to each role.

Specify which roles that the user selected in the **Assignee** table can view and alter in the Set Up Participants table for the specific activity being defined, by selecting any of the following check boxes:

- **View** displays the associated role in the Set Up Participants table. If this check box is disabled, the Assignee will not see this role in the Set Up Participants table and therefore is unable to alter the participants.
- **Remove** gives the Assignee permission to remove participants from this role in the Set Up Participants table.
- **Add** allows the Assignee to add participants that are selected in the Resource Pool, to the role in the Set Up Participants table.

The **Resource Pool** table, at the right of the **Role Setup** tab, displays the members available for role assignment, as identified in the **Resource Pool** tab.

When a user is assigned a set up participants activity, the list of roles they are permitted to edit is determined by the **Defined Roles** section, specified for the assignee. The list of users or user-defined groups the assignee is permitted to add to the role(s) is determined by the team members specified for the assignee in the **Resource Pool** tab. Thus, different assignees of the same set up participants activity may be permitted to modify a different subset of roles, and may be presented with a different list of resources which may be assigned to those roles.

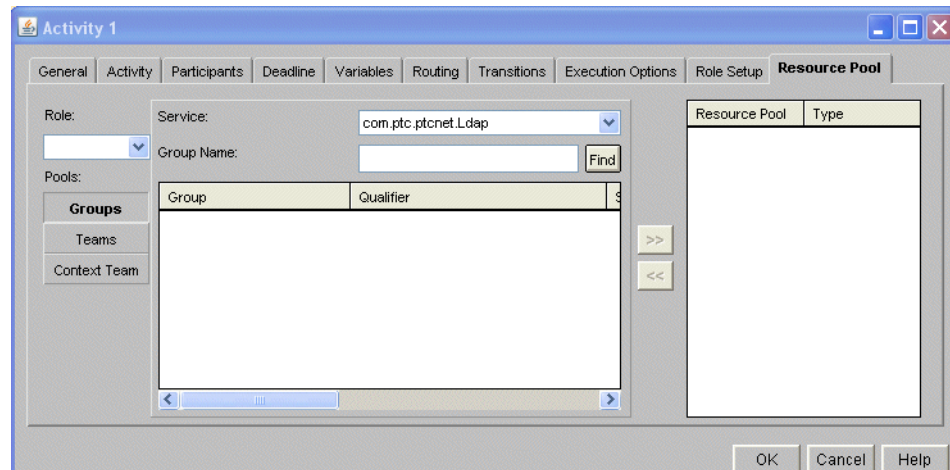
## Resource Pool Tab

On the **Resource Pool** tab, shown below, you can identify the pool of users which will be available when defining a role permission map for a *Set Up Participants Activity*. This pool of users is the superset of participants available to be added to any role via the Set Up Participants activity.

You can select from user-defined groups, team templates, and context teams to form the resource pool for each role individually. The users identified in the pool are then displayed in the **Role Setup** tab, from which the role permission map is defined.


When the **Set Up Participants** table is displayed in the Task Details page, members identified on the **Resource Pool** tab will be displayed. Depending on the role permission map defined in the **Role Setup** tab, the assignee of the task can alter the role membership.

To add to the **Resource Pool** table, select it on the left pane, and click the >> button. To remove an assignee from the table, click the << button.



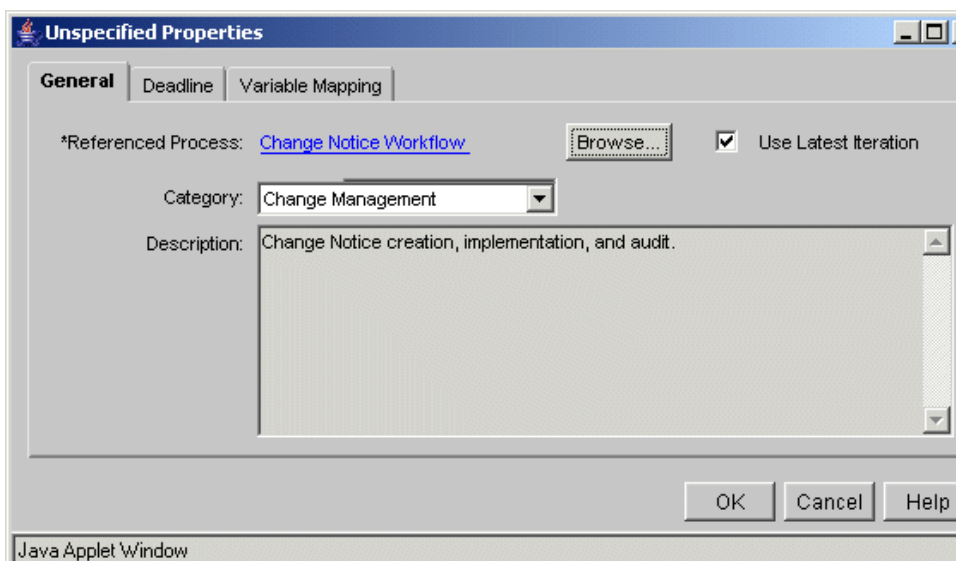
## Defining a Subprocess

A subprocess, or proxy process, can be included as a node of another process; the parent process.

To add a proxy process to the workflow template, click on the Proxy Process icon () and then click inside the process diagram to place the node. The properties window opens. Click the **Browse** button to select an existing workflow process that will then become the proxy process. The selected process name appears as a hyperlink next to **Referenced Process**.

You can designate a category and enter a description for the proxy process. To ensure that the proxy process is edited when the parent process is changed, select **Use Latest Iteration**.

If you do not select a process, you cannot return to the process diagram. Instead, you must click **Cancel**; the proxy process node disappears from the process diagram.



The image shows a Java Applet Window titled "Unspecified Properties". It has three tabs: "General", "Deadline", and "Variable Mapping". The "General" tab is selected. Inside the "General" tab, there is a label "\*Referenced Process:" followed by a text field containing the hyperlink "Change Notice Workflow". To the right of this text field is a "Browse..." button. Further right is a checked checkbox labeled "Use Latest Iteration". Below these, there is a "Category:" label followed by a dropdown menu showing "Change Management". Below the category is a "Description:" label followed by a large text area containing the text "Change Notice creation, implementation, and audit." At the bottom right of the dialog are three buttons: "OK", "Cancel", and "Help".

The assignments that are generated from an activity defined in the proxy process will not display roles that have been defined in the parent process (the outer process nodes). However, activities that are defined in the parent process nodes display a list of all roles used anywhere in the (outer) workflow process. This means that an assignee of an activity that belongs to a proxy process won't be able to modify the participant roles that belong to the parent process. But the assignees of activities that belong to a parent process can modify the participant roles that belong to the proxy process.

Roles are defined in the outer process activity nodes on the [Role Setup Tab](#). The Set Up Participants table (accessed from the Assignments table of the assignee) will display roles based on the permissions that are granted by the workflow administrator in the **Role Setup** tab of the workflow process.



For example, if the Approval Process workflow is selected when defining the proxy process, only the roles that are defined in the Approval Process workflow will display in the generated assignments of the activities in defined in the proxy process. The assignments that are generated from the outer process activity nodes will display all of the roles that have been defined anywhere in the outer process (also referred to as the parent process), plus all the roles that are defined in the inner process (in this example, the Approval Process). For more information about defining roles and specifying permission for each role, see the [Role Setup Tab](#) section.

Set a deadline for the proxy process by clicking the **Deadline** tab. For more information, see the [Deadline Tab](#) section.

To map variables, click the **Variable Mapping** tab. For more information, see [Variables Tab](#) section.

**Note:** Ad hoc activities and blocks are similar to proxy process, in that they are composed of a group of activities. A block is a way of simplifying the graphical representation of the process, by combining a number of activities under one icon. An ad hoc activity is a group of activities defined at runtime.

## Defining Connectors

The Workflow Process Editor supports the following connector types:

Connector Type	Description
Start	The Start connector represents the starting point in a process. Each process has only one Start connector, which cannot be removed or duplicated.
And	An And connector does not fire until all predecessor links have fired. That is, it waits for all preceding activities to complete before allowing the process to continue. For example, a Promote activity can be connected to multiple review activities by an And connector.

Connector Type	Description
Or	An Or connector fires if any of its predecessor links fire. That is, it allows the process to continue if any of the preceding activities have completed. For example, a Revise activity can be linked to multiple review activities by an Or connector.
Threshold	<p>A Threshold connector fires if a user-defined number of predecessor links fire. That is, it allows the process to continue only when the user-defined number of preceding activities have completed. To set the number of activities that must complete before a Threshold connector fires, enter a number in the <b>Firing Threshold</b> text box, on the <b>Threshold Properties</b> window.</p> <p>To set a dynamic threshold, in which the firing threshold is set to 0 at runtime and is reset to the number of started predecessor activities, select <b>Add One</b> from the drop-down list in the <b>Action</b> list on the preceding <b>Link Properties</b> window.</p>
Conditional Router	<p>A Conditional Router fires user-defined events based on an automatic event firing expression. Because you can define user events and fire them with an expression for all connectors, the Conditional Router is essentially an Or connector identified by a special icon.</p> <p><b>Note:</b> Whenever possible, expressions should be inside a <i>try catch block</i>. Inside the catch block, take corrective action for the exception.</p>

You specify custom routing events, which can be used to control process flow, on the **Routing** tab panel on the Connector Properties window. For more information, see the [Routing Tab](#) section.

**Note:** Because the Or connector and the Threshold connector do not require all activities to be completed before allowing the process to continue, unnecessary activities can remain open. To terminate these activities, click **Terminate Open Predecessor Activities When Fired** on the **Or Properties** window, or the **Threshold Properties** window.



## Defining Links

Links define the flow of control among nodes within a process definition. They also determine which actions are performed in an activity when a predecessor activity broadcasts (or emits) events. For example, when a user completes a review task (indicating completion by clicking a button on that task page), you can specify that the completion event will cause a link to the next activity to fire.

Click the hyperlink associated with a link representation, or double-click the rule between nodes in your diagram, to access the **Link Properties** window, which allows you to map events to actions.

On the **Link Properties** window, you can map events to actions.

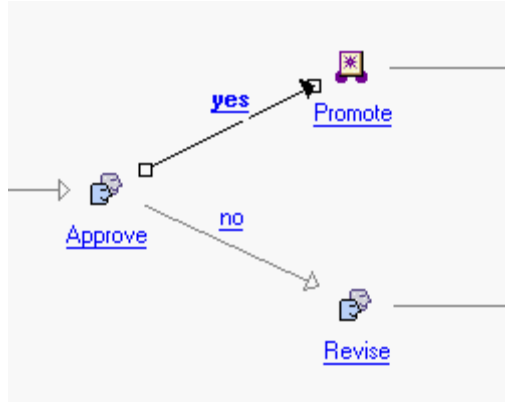
To map an event (in the left column), select a successor action from the drop-down list in the right column. You can specify more than one event to cause the same action to occur. To indicate that an event will be ignored, leave the field in the **Action** column blank. If an event is ignored, no action is performed when that event is emitted.

To reset all connectors in an events path when it is fired, select the **Loop Link** check box. Selecting the Loop Link check box ensures that connectors that are in the loop get reset prior to the workflow loop getting traversed again. The connectors that have already fired are reset, and they can be fired again. Loop links are designed to reset all connectors in the path of the workflow activity that is the target of the loop link. A connector is considered in the path of this activity if it is reached by going forward from the activity AND going backwards from the activity. Loop links appear in red.

You can define custom routing events for most activities and processes. When you do so, these events are also displayed in the **Link Properties** window, and you can map them to actions in a successor activity. For example, you could include in a process definition the following assigned activities:

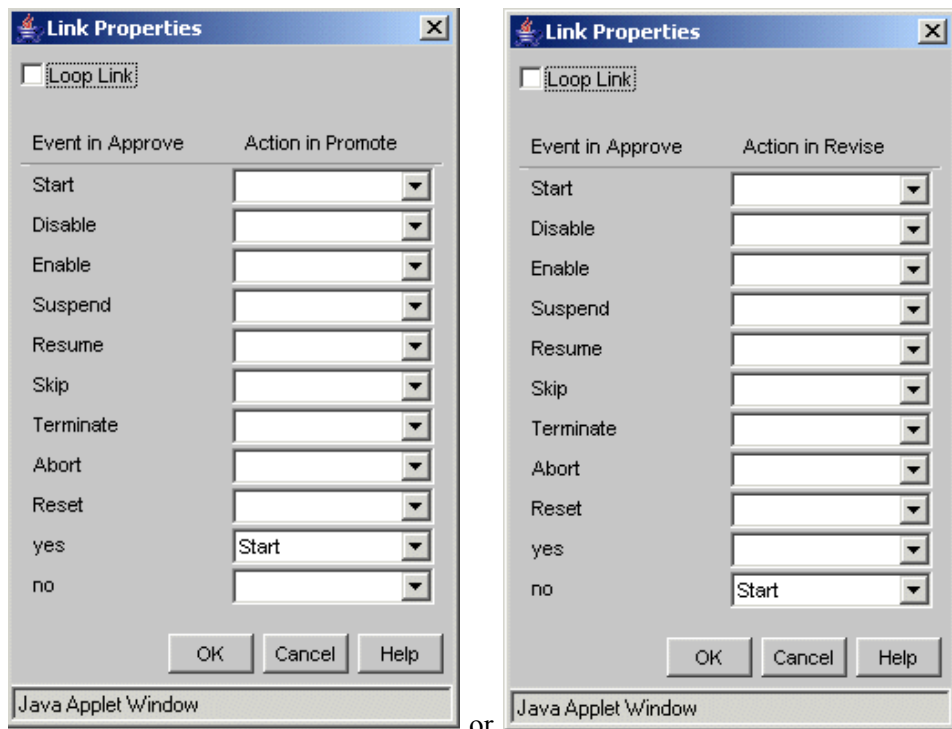
- Approve
- Revise
- Promote

See the following figure:



The Approve activity defines two custom routing events, yes and no. This activity has two links: one link connects to the Revise activity, and another connects to a Promote robot. The link to the Revise activity can be configured to perform the Start action in Revise when the no event is emitted from the Approve activity. The other link can be configured to perform the Start action in the Promote robot when the yes event is emitted from Approve. In this way, the flow of control to the Revise or Promote activity is controlled by the event emitted from the Approve activity.

The figures that follow show possible results:



# Importing and Exporting Workflow Templates

Before you begin, you should be familiar with the following information regarding importing and exporting:

- Be sure to upgrade to the latest maintenance only release (MOR), as it becomes available, to ensure you have the latest enhancements to the import and export functionality.
- You can import templates into a later version of Windchill; importing to an earlier version may not work. The data you are importing must comply with the database integrity and structure of the older version. To import templates from versions prior to Windchill 7.0, you must convert the CSV to XML format. Use the CSV2XML utility to convert CSV files to XML files. The syntax is as follows:

```
windchill wt.load.util.CSV2XML -input <input file or directory>  
-output <output directory> -root <root directory> -help
```

- Importing or exporting workflow templates creates objects in a JAR or ZIP file format. (This is the same format that the load.Installer functionality uses.) Use a zip utility like WinZip to create a ZIP file consisting of XML files. To create a JAR file, use the following syntax:

```
jar -cf <jar file name> <xml files to be included>
```

or

```
jar -cvf (for verbose output to print each file name as it is  
added)
```

- You can import any number of templates at a time. You can have many XML files, each representing one workflow template, in one JAR or ZIP file, or you can have one XML file representing many workflow templates. The order of the templates is important. For example, if template A has a subprocess template B, then B must be placed before A in the XML.

There is no limit to the number of workflow templates you can export. To export multiple templates into a single JAR or ZIP file, hold down the CTRL key, select the templates you want to export, and click **Export**. All the selected templates are exported to the same JAR or ZIP file.

- Iterations of a workflow template are not imported or exported. When you export a workflow that contains a fixed reference to a specific iteration that is not the latest iteration, the import fails. PTC recommends that you change all fixed references to floating references (that is, references to the latest iteration) before you export the workflow template.
- PTC recommends that you do not export data from one version of Windchill and import it into a different version. Instead, migrate the data in your database and then export the templates.

- Errors may occur, especially when importing workflow templates. Some errors result in messages displayed; others may be fatal. Check the method server log for error information.
- When you export a workflow template, only the template itself is exported. This includes references to underlying objects, such as principals, roles, and actor roles. The underlying objects themselves are not exported. If the export file is used to import the template into another database, the underlying objects must exist in the database, or the import fails and errors appear in the method server log. This can occur especially when importing the object into a different system. Be certain that all underlying objects referenced in the XML representation of the template exist.
- If a workflow template is imported and a template with the same name already exists in the Import directory, the results depend on the Iteration On Import setting in the wt.properties file. If wt.workflow.iterateOnImport=true, the imported workflow template is appended to the existing template as a new iteration. If it is set to FALSE, the imported file causes a method server exception, stating that there is a duplicate name, and the template is not imported.
- If you want to reuse an existing template that resides in another context, export the template from source context to your local client file system, then import to your target context.

## Importing and Exporting Workflow Templates

Use the **Import** and **Export** buttons on the **Workflow Administrator** page to import or export workflow templates.

To import one or more templates from a JAR or ZIP file, select the file from the local file system in the **Import** window, and click **Import**.

To export a workflow template into an XML representation of the template, use the following procedure:

1. Select a template from the **Workflow Administrator** page, and click **Export**. (The **Export** button is disabled if you do not have a template selected.)
2. A grant permission window may appear asking for permission to access the local file system and to write a file on that system. If you select the remember selection check box, permission need only be granted once. Once permission is granted, a **Browse for file** picker opens, defaulting to the system **temp** folder.

3. You can pick a file that exists or type in a new name. If the file name exists, you are asked to confirm to overwrite the file. You must click **Yes** to continue the export. If the file name does not exist, a new file is created.

There is no confirmation that the export is completed. When the progress bar and the hourglass on the workflow administrator applet disappear, the export is complete.

## Importing Across Releases

Importing across releases is denied by default due to the fact that it could cause data corruption.

The cross release import function can be enabled by setting a property file. However, any data inconsistency issues that arise after allowing such an import will not be supported.

Set the following wt.property file to true.

```
wt.ixb.import.allowCrossReleaseImport=true
```

Restart the method server to reload the property file.

## Process Manager Toolbar Access Control

Process Manager toolbar buttons give authorized users the ability to change workflow processes. An authorized user can make vital changes to an activity or a process, such as terminating or completing it. Access to Process Manager buttons is defined in the following file:

```
<install directory>/codebase/wt/clients/workflow/manager/process-manager.properties
```

Out-of-the-box, full access is given only to the system administrator and the workflow administrator, as defined in the following properties:

```
activityrestartAccessControl=Administrators
activitysuspendAccessControl=Administrators
activityresumeAccessControl=Administrators
activityterminateAccessControl=Administrators
activitycompleteAccessControl=Administrators
processsuspendAccessControl=Administrators
processresumeAccessControl=Administrators
processterminateAccessControl=Administrators
processcompleteAccessControl=Administrators
activityrestartAccessControl=WorkflowAdministrators
activitysuspendAccessControl=WorkflowAdministrators
activityresumeAccessControl=WorkflowAdministrators
activityterminateAccessControl=WorkflowAdministrators
activitycompleteAccessControl=WorkflowAdministrators
processsuspendAccessControl=WorkflowAdministrators
processresumeAccessControl=WorkflowAdministrators
processterminateAccessControl=WorkflowAdministrators
processcompleteAccessControl=WorkflowAdministrators
```

To add other user-defined groups, add additional AccessControl lines to the file, in the format:

```
<action>AccessControl=<group name>
```

When you have edited the properties file the, you must recreate the JAR file (wtWorkFCS.jar).

## Viewing Workflow History

To optimize and streamline your workflow process definitions, you should capture and record events, such as state transitions and variable updates that occur during process execution. The ordered sequence of these events is called a workflow history, and allows you to capture significant events.

In addition, you can specify that a keyed event be emitted, which allows synchronization of other Windchill managers and servers with process events. As described in this section, you can specify the following:

- Which events are to be ignored
- Which are to be recorded
- Which are to be emitted as keyed events
- Which are to be both recorded and emitted

**Note:** When there is a loop link in a workflow with a conditional route, only the last vote is recorded.

## Selecting Events

The following events can be generated during execution of a workflow process:

- Process creation, which occurs when a Start command is issued for an existing, enabled process definition.
- Change of state, which occurs when an execution object (a process or an activity) changes states (for example, from running to completed).
- Change of data value, which occurs when values are read into a process activity or written to process variables.
- Change of assignee, which occurs when assignment of a workflow task changes from one user to another.
- Error event, which occurs when an exception is thrown during process execution.

You can identify the events to be recorded in the workflow history or emitted as keyed events by setting the default event configuration in the wt.properties file.

**Note:** Workflows that are created before changes to the wt.properties are applied, are not affected.

When one or more of the following properties is set to true, the corresponding event is recorded or emitted:

- wt.workflow.engine.recordProcessStateChange
- wt.workflow.engine.recordProcessDataChange
- wt.workflow.engine.recordActivityStateChange
- wt.workflow.engine.recordActivityDataChange
- wt.workflow.engine.recordProcessCreation
- wt.workflow.engine.recordAssigneeChange
- wt.workflow.engine.recordException
- wt.workflow.engine.recordVoting
- wt.workflow.engine.emitProcessStateChange
- wt.workflow.engine.emitProcessDataChange
- wt.workflow.engine.emitActivityStateChange
- wt.workflow.engine.emitActivityDataChange
- wt.workflow.engine.emitProcessCreation
- wt.workflow.engine.emitAssigneeChange
- wt.workflow.engine.emitException

All running workflow processes record and emit events based on this default configuration. As described in the next section, however, you can use the Workflow History Viewer to change the events recorded or emitted by a given process.

## Using the Workflow History Viewer

You can view workflow-generated events on the Workflow History Viewer, a simple ASCII utility. Issue the following command to start the Workflow History Viewer:

```
windchill wt.workflow.engine.WfMonitor
```

The following is the main menu of the Workflow History Viewer:

Workflow Monitor - Main Menu

Existing Workflow Processes

-- -- -- -- --

>> 1. Simplest, key = 3347 (Running)

Audit events

-- -- -- -- --

No event retrieved

1. Select process.
2. Show selected process.
3. Delete selected process.
4. Refresh processes.
5. Change event configuration for selected process.
6. Show selected events for selected process.
7. Show all events for selected process.
8. Show selected events for all processes.
9. Show all events for all processes.
10. Select event.
11. Show event.
12. Show event source.
13. Refresh events.
14. Delete events for selected process.
15. Delete events for all processes.
16. Exit

>>> Choose an option:

This menu includes a list of all existing processes and a list of all retrieved events. The list of retrieved events is initially empty.

In addition to these lists, the main menu provides 16 options. A description of each follows:

### Select Process

Select this option to select a process, so associated events can be queried. The selected process is preceded by the characters >>.



### Show Selected Process

Select this option to display the following information about the selected process, as well as all of its activities and subprocesses that have been started:

- The process status (for example, running).
- The process creator.
- The team with which the process is associated.
- An indication of whether or not the process or activity is overdue (that is, whether it has exceeded its specified duration).
- The time at which the process or activity started.
- The time at which the process or activity ended (not shown if the process or activity has not been closed).
- The deadline for the process or activity, if one exists.
- The event configuration for the process.
- Suspend and alert times, in milliseconds.

The following shows a sample process display:

```
Simplest - Running
Creator: Administrator
Team: Default
Times: is overdue = false
      start time = 1998-11-20 14:08:53.0
      suspend time = 0, alert time = 0
Event configuration:
Process (RECORD/EMIT): Creation: R, State change: RE
Activity (RECORD/EMIT): Data change: E
Exception (RECORD/EMIT): Exception: R
Activities:
    *** act_simplest (wt.workflow.engine.WfTestActivity) -
Running
      Times: is overdue = false
      start time = 1998-11-20 14:08:56.0
      suspend time = 0, alert time = 0
```

### Delete Selected Process

Select this option to delete the selected process.

### Refresh Processes

Select this option to refresh all processes. The Workflow History Viewer does not automatically edit the status of a displayed process. You must explicitly refresh it in order to see new processes and changes of state since the last refresh.

## Change Event Configuration

Select this option to open an event configuration editor. With the editor you can select the events that should be emitted, recorded, or ignored, for the selected process.

Edit event configuration

Current configuration

Process (RECORD/EMIT): Creation: R, State change: RE

Activity (RECORD/EMIT): Data change: E

Exception (RECORD/EMIT): Exception: R

1. Create process - record: true
2. Create process - emit: false
3. Change process state - record: true
4. Change process state - emit: true
5. Change process data - record: false
6. Change process data - emit: false
7. Change activity state - record: false
8. Change activity state - emit: false
9. Change activity data - record: false
10. Change activity data - emit: true
11. Change assignment - record: false
12. Change assignment - emit: false
13. Execution error - record: true
14. Execution error - emit: false
15. Save configuration
16. Save configuration and return
17. Return (looses changes since last save)

>>> Choose an option:

As previously described, all processes are governed by the event configuration specified in the wt.properties file, unless this submenu is used to change that configuration for the selected process. Selecting an option on this submenu toggles the current setting. For example, if you select option 1, you will change the setting of **Create process** from true to false. Consequently, process creation would no longer be a recorded event.

### Show Selected Events for Selected Process

Select this option to open the submenu shown in the following figure. You can select an option from this submenu to specify the type of event in which you are interested. (In the example, option 8 is selected, indicating that the user is interested in all events associated with the selected process.)

```
Types of events to show
1. Process creation
2. Process state change
3. Process data change
4. Activity state change
5. Activity data change
6. Assignment change
7. Execution error
8. All
9. None (return)
```

```
>>> Choose an option: 8
```

When you have select an option from this submenu, the requested events are retrieved, and the list of events is refreshed. For each event, the list identifies the event type and the execution object associated with it:

```
Audit events
```

```
>> 1. PROCESS_STATE_CHANGED, process = Simplest
```

### Show All Events for Selected Process

Select this option to represent a shortcut to selecting **All**, (option 8) from the submenu of event types described in the preceding section. If this main menu option is selected, all events associated with the selected process are shown.

### Show Selected Events for All Processes

Select this option to display all events of the selected type for all processes.

### Show All Events for All Processes

Select this option to display all stored events for all processes. This list can be very large.

### Select Event

Select this option to select another event. When events are retrieved, the first event in the list is automatically selected.

### Show Event

Select this option to display detailed event information for a selected event, as shown in the following figure:

```
event type = PROCESS_STATE_CHANGED
timestamp = 1998-11-20 14:08:53.0
activity key = 0 activity name = null
process key = 3347
process name = Simplest
process template name = simplest
old state = Not started
new state = Running
```

### Show Event Source

Select this option to display the execution object (process or activity) that is the source of the selected event.

### Refresh Events

Select this option to refresh all events corresponding to the selected process and event type. This is necessary to view events generated since the last refresh.

### Delete Events for Selected Process

Select this option to delete all the events of the last selected type associated with the selected process.

### Delete Events for All Processes

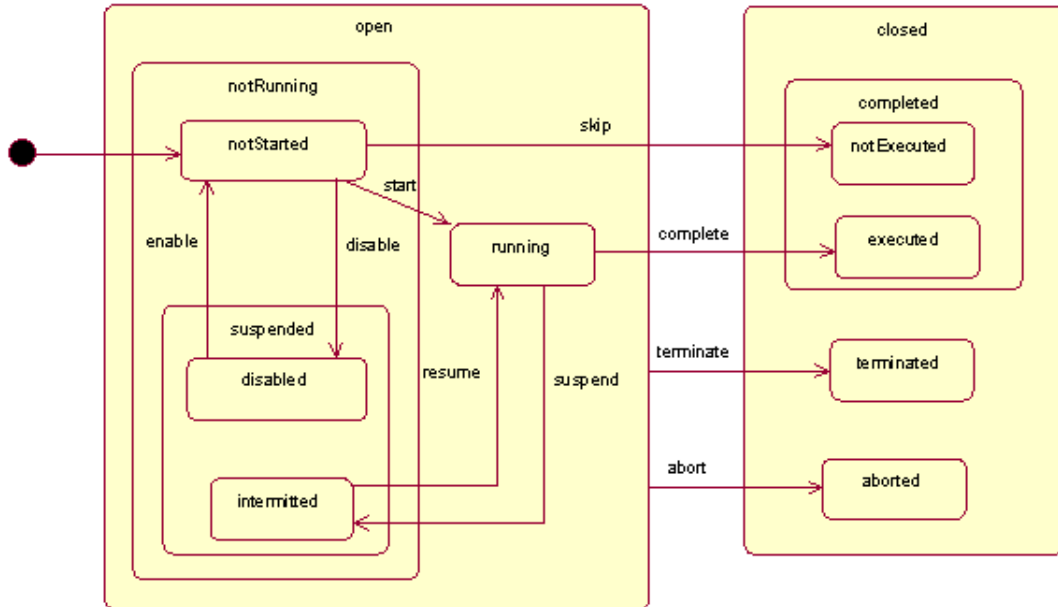
Select this option to delete all events of the last selected type associated with all processes.

### Exit

Select this option to terminate the Workflow History Viewer session.

## Workflow Instance States

The following diagram shows the states that an execution object (activity or process) can be in, and the transitions from state to state.



Rounded boxes represent states; arrows represent transitions. The actual states are always the innermost ones. The others are super-states, indicating a collection of substates. For example, a query for closed processes returns the processes that have completed successfully and also the processes that have terminated or were aborted.

The initial state for all execution objects is the not started state. The following is the normal sequence of states for an execution object is:

1. Not Started
2. Running
3. Executed

The final state can be reached with the following two transitions:

- Start
- Complete

Some transitions apply to more than one state. This is indicated by an arrow that begins in a super-state. For example, terminate transitions from any open state to the terminated state. An additional transition, reset, is not shown in the diagram. The reset transition brings any object back to the not started state.

Both `Open.NotRunning.Suspended.Disabled` and `Open.NotRunning.Suspended.Intermitted` are labeled suspended in the GUI; however, the `Open.NotRunning.Suspended.Disabled` state is currently not used, so there is no chance of confusion.

The following is the model for connectors:

- Enabled
- Disabled

## Domain-based Workflows

During initialization, loading administration data into the Windchill database creates an access control policy rule that grants all users read, modify, and create permissions for objects of type `wt.workflow.engine.WfExecutionObject` in the `/System` domain. The `WfProcess` class extends the `WfExecutionObject` class, so this rule applies to workflow process instances. All Windchill users can view process information by using Local Search. This is the default out-of-the-box behavior.

In order to provide better control over access to workflow process instances, you can set a property to enable domain-based workflows. The workflow processes are created and run within the same administrative domain as the primary business objects associated with the processes. You can specify access control policies for the domains associated with the workflows, granting different rights to `WfProcess` objects for each of the domains.

The property in the `wt.properties` file that toggles domain-based functionality is `wt.workflow.engine.domainBasedWorkflow`. The default setting is false. If this property is set to true, the domain-based functionality takes effect following method server restart. `WfProcesses` do not immediately change to their new domain; however, upon the next workflow state change (for example, from not started to running), the `WfProcess` changes domains, as appropriate.

The user can use Local Search to find workflow processes in domains for which they have read access to `WfProcess` objects. If access control policy rules do not grant a user access to the primary business object, the user can be granted ad hoc access to the object. These rights are explicit to the primary business objects and are not applied to the parent folder. Workflow process objects are always created in the context of the business object and are not accessible to users that are not members of that context.

## Out-of-the-Box Workflow Templates

This section provides information about the out-of-the-box workflow templates and process images that can be associated with them.

The out-of-the-box workflow templates in Windchill allow you to automate procedures in which tasks and documents are passed among participants.

PTC Global Services consultants can help you determine how the out-of-the-box workflows correspond to your business processes. Although you can modify workflows if necessary, you will get better results if you map the existing workflows to your processes, rather than create new workflows based on your processes.



**Caution:** Changes you make to workflows may not be compatible with future Windchill release and service packs, and may require additional support from PTC Global Services consultants.

The following list provides all out-of-the-box workflow templates by solution that are loaded into the site context during installation:

- Windchill PDMLink:
  - Change Activity
  - Change Notice
  - Change Request
  - Problem Report
  - Promotion Request Approval Process
  - Promotion Request Review Process
  - Variance Workflow
- Windchill ProjectLink:
  - Approval Process
  - Notify Process
  - Release Process
  - Review Process
  - Two Level Approval Process

**Note:** The Workflow Administrator is only accessible in Windchill PDMLink. That is, it cannot be accessed from the Project or Program tab.

- All Windchill solutions:
  - Submit
  - Review

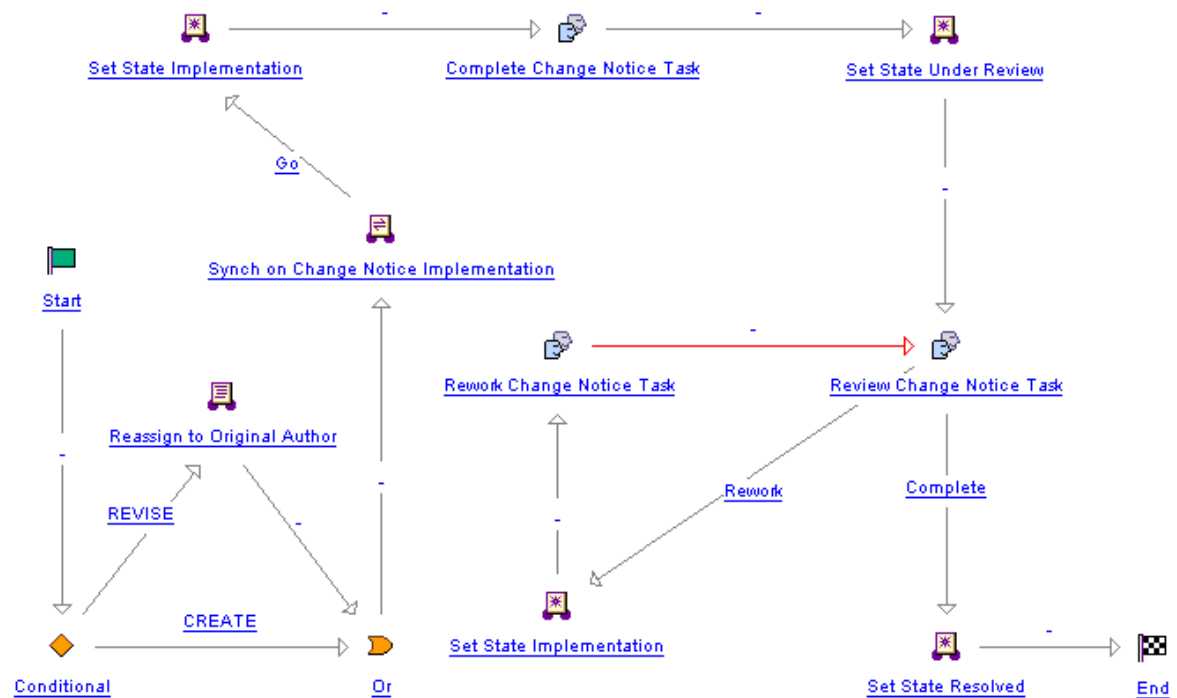
## Change Management Workflows

**Note:** When the Windchill 8.0 out-of-the-box Change Management workflows are run in Windchill 9.0, the user will see the activity variables in the task details page that are defined in the 8.0 workflow templates, instead of the activity variables that are defined in the 9.0 workflow templates. Additionally, when running the Windchill 8.0 Change Management workflows, the user might see two Comment fields on the task details page; one for a defined activity variable and one for the Routing History table.

The following section provides out-of-the-box workflow templates specific to Change Management. Each is listed below along with a summary of the steps in each workflow.



## Change Activity Workflow



### 1. Complete change notice task

Review the assigned change notice task, specify edits to the affected data, and submit the completed task.

### 2. Review change notice task

Review the data modifications resulting from the preceding step and approve or reject the modifications.

### 3. Rework change notice task

According to the reviewer's instructions step 2, perform the required modifications and resubmit them to the Change Administrator III for audit.

## 16-54



Review the change request and provide comments as needed.

## 5. Modify Roles

Update the change request team roles as needed.

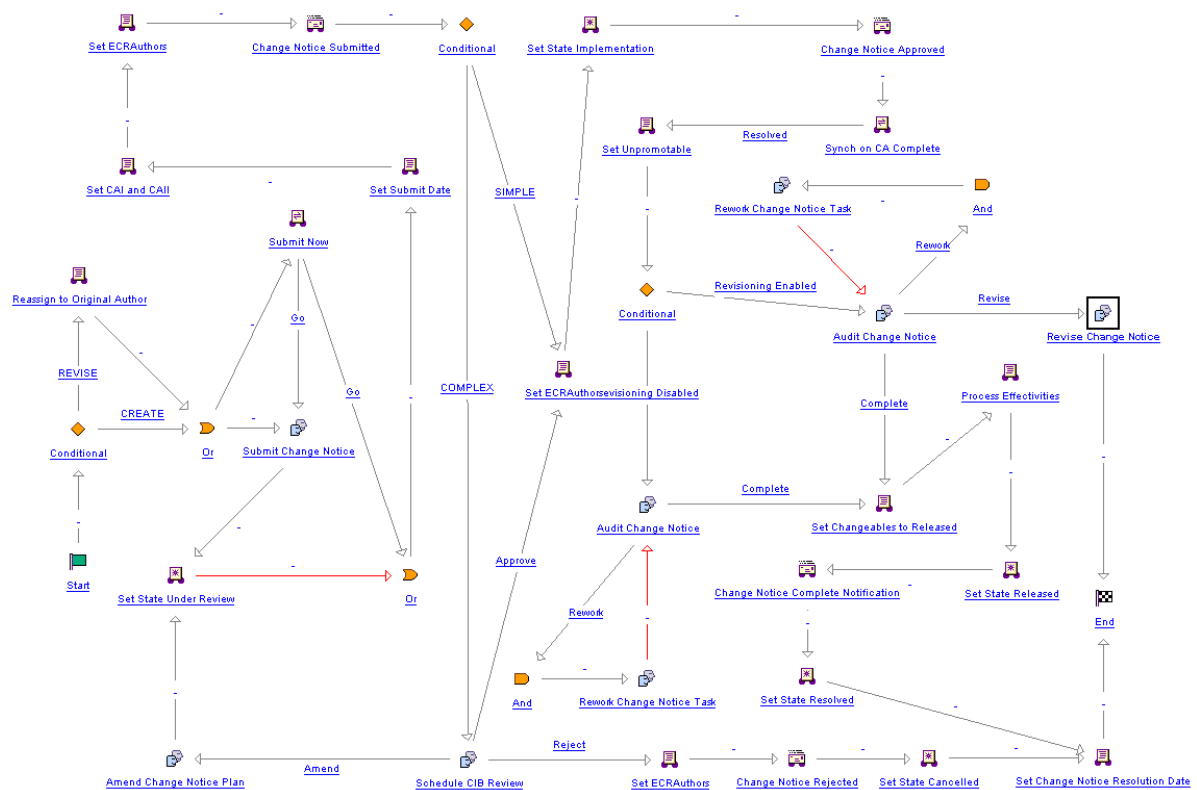
## 6. Create new change notice

Create a new change notice as necessary for the corresponding change request.

## 7. Schedule CRB (Change Review Board) review

For Full Track change requests, schedule and facilitate CRB review and record the CRB decision.

## Change Notice Workflow



## 1. Submit change notice

Submit change notice for review by the Change Implementation Board (CIB).

## 2. Schedule CIB review

Schedule and facilitate the Change Implementation Board review and record the CIB decision.

3. Amend change notice

If necessary, amend the change notice plan based on the CIB's review and resubmit to the CIB.

4. Audit change notice

Review the change notice data modifications and approve or reject the change notice implementation.

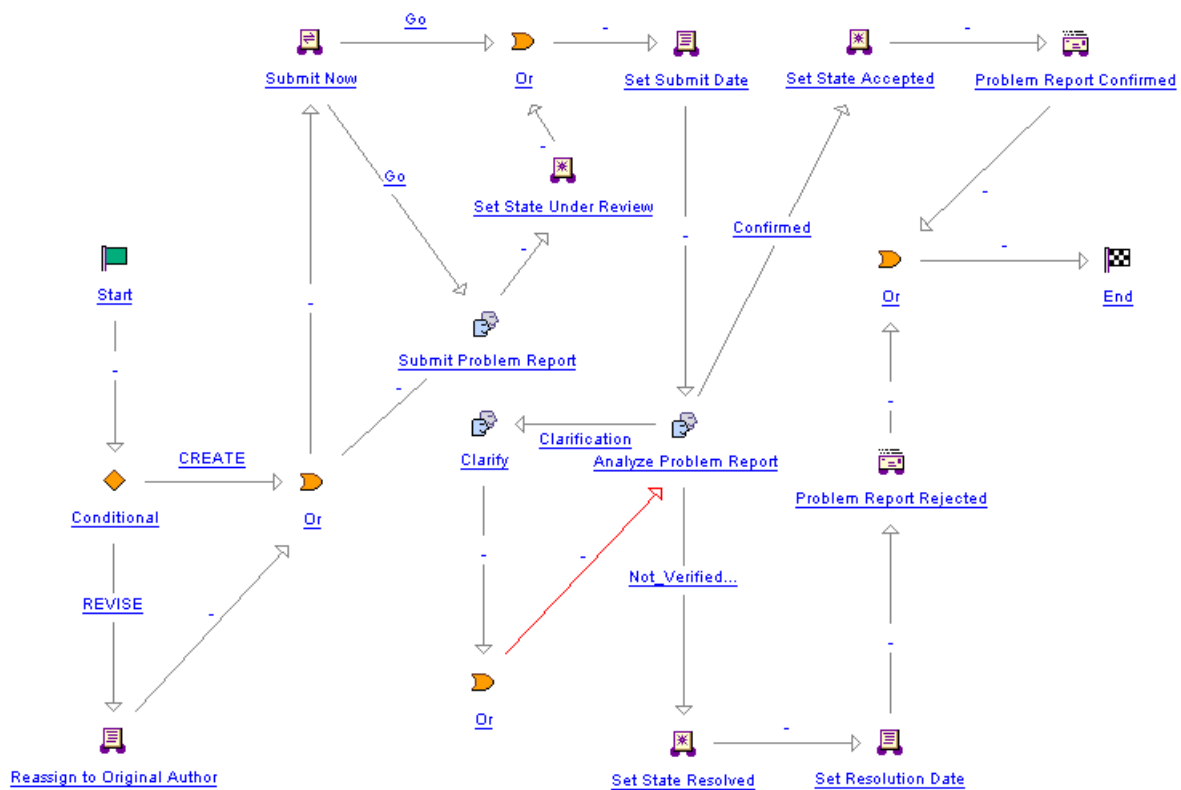
5. Rework change notice task

Rework change notice task and resubmit to Change Administrator II for audit.

6. Revise change notice

If necessary (and revisioning is enabled), revise the change notice based on the CIB's review.

## Problem Report Workflow



1. Submit problem report

Submit the problem report for review by Change Administrator I.

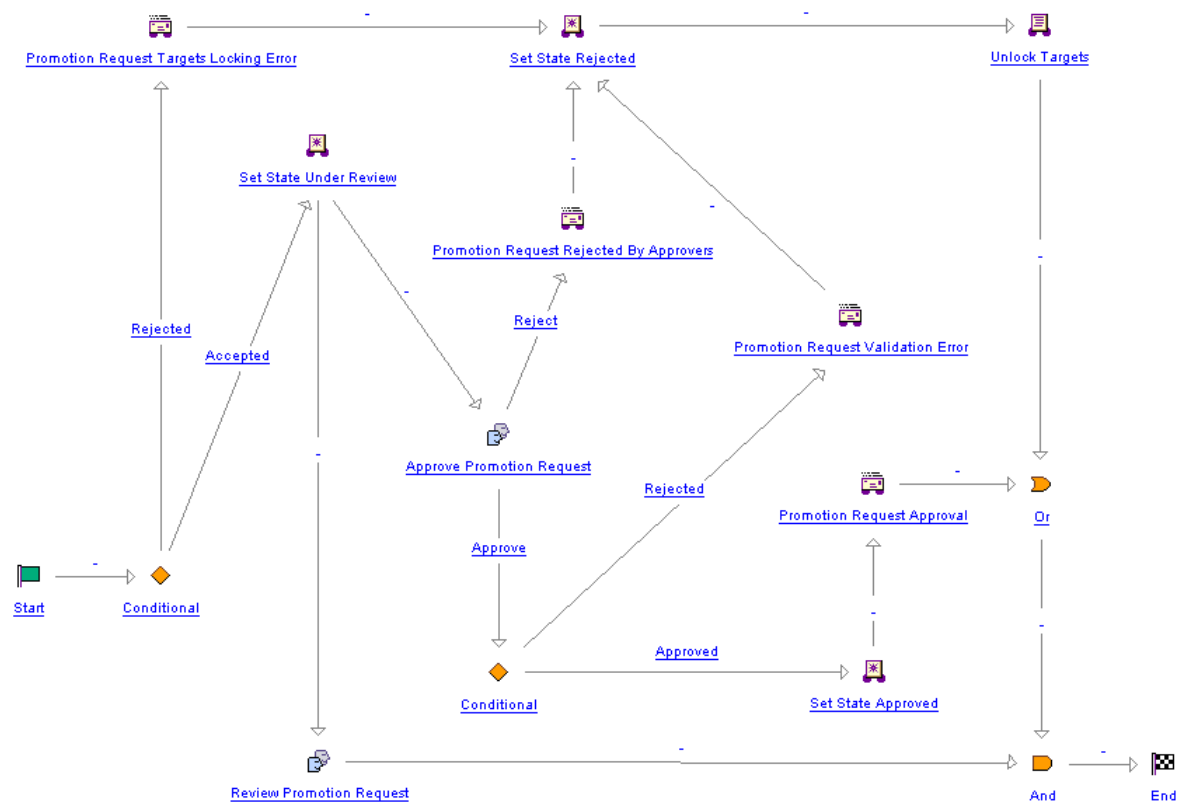
2. Analyze problem report

Analyze the problem report, associate the relevant analysis information, and then confirm or reject (as not verified) the problem report.

3. Clarify problem report

The problem report is sent back to the author for the addition of further information.

## Promotion Request Approval Process Workflow

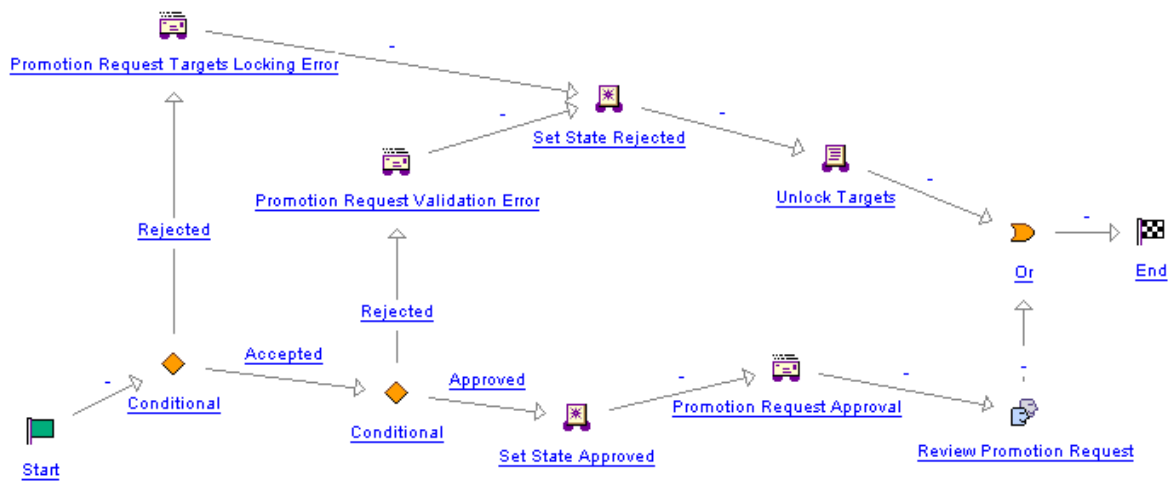


1. Review Promotion Request

Invite Reviewers from the Promotion Reviewers system group, as well as participants added by the requesting author, to provide comment on these items.

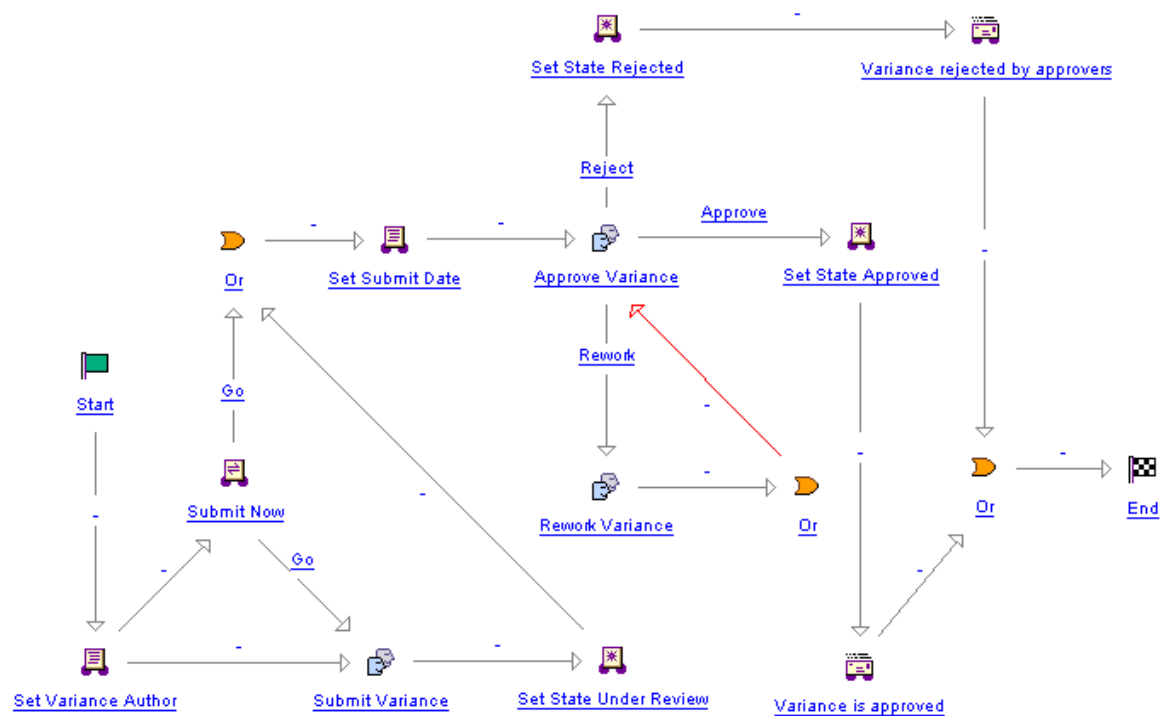
2. Approve promotion request

## Promotion Request Review Process Workflow



The Review Promotion Request workflow routes the promotion request to a group of people to review and comment on the items. All items on the promotion request are automatically promoted to the requested life cycle.

## Variance Workflow



1. Submit variance  
Submit the variance for approval.
2. Approve variance  
Validate the variance, review the identified variance objects and then approve, reject, or request rework by the variance author.
3. Rework variance  
Rework the variance and resubmit for approval.

## Out-of-the-Box Process Images

Process images are available out-of-the-box for routing and change management processes and can be associated with workflow templates via the Workflow Administrator. Process images can also be edited to correspond with changes you make to the workflow templates.

Process images are graphical overviews of the workflow process providing an at-a-glance summary of the process flow. They can also included textual descriptions of the process flow.

Once the process images have been associated with the workflow templates, they can be viewed from the Assignments table in the context of a specific routing assignment or change request. They provide the end-user with a quick look at the process flow and the behavior of the task in which they are participating.

For detailed procedures for editing and associating process images to the workflow templates, see the Workflow Administrator online help.

The following process images are available out-of-the-box in Windchill ProjectLink:

- Approval Process
- Release Process
- Review Process
- Two Level Approval Process

The following process images are available out-of-the-box in Windchill PDMLink:

- Change Activity Workflow
- Change Notice Workflow
- Change Request Workflow
- Problem Report Workflow
- Promotion Request Approval Process
- Promotion Request Review Process
- Variance Workflow

## **Workflow Template Execution Flags**

The workflow engine uses several flags to control behavior during the execution of the workflow. These flags default to wt.property values. Execution flags fall into three categories:

- Process - These flags affect the entire workflow process.
- Activity - These flags affect individual workflow activities.
- Both process and activity - These flags are meaningful for both workflow processes and workflow activities.



## Process Flags

These flags are meaningful only to workflow processes.

Flag Name	Associated wt.property	Short Description
Create process - record	wt.workflow.engine.recordProcessCreation	Determines if the workflow process creation events are recorded as part of the workflow history. You can change the value of this property for a specific process by altering its execution characteristics.
Create process - emit	wt.workflow.engine.emitProcessCreation	Determines if workflow process creation events should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific process by altering its execution characteristics.
Change process state - record	wt.workflow.engine.recordProcessStateChange	Determines if the workflow change of state events should be recorded as part of the workflow history. You can change the value of this property for a specific execution object by altering its execution characteristics.
Change process state - emit	wt.workflow.engine.emitProcessStateChange	Determines if the workflow state change events that occur with processes should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific process by altering its execution characteristics.
Change process data - record	wt.workflow.engine.recordProcessDataChange	Determines if the workflow change of data events should be recorded as part of the workflow history. You can change the value of this property for a specific execution object by altering its execution characteristics.

<b>Flag Name</b>	<b>Associated wt.property</b>	<b>Short Description</b>
Change process data - emit	wt.workflow.engine.emitProcessDataChange	Determines if the workflow state change events that occur with processes should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific process by altering its execution characteristics.
Execution error - record	workflow.engine.recordException	Determines if the workflow execution exceptions should be recorded as part of the workflow history. You can change the value of this property for a specific execution object by altering its execution characteristics.
Execution error - emit	workflow.engine.emitException	Determines if workflow execution exception should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific execution object by altering its execution characteristics.
Notify on exception	wt.workflow.engine.notifyOnException	This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific process by altering its execution characteristics.
Abort process on exception	wt.workflow.engine.abortProcessOnException	Determines whether to abort the workflow when an exception occurs. If true, the process instance is aborted if a contained activity aborts.
Notify on overdue	wt.workflow.engine.notifyOnOverdue	Determines whether a notification should be sent to the principals in the overdue notification list when the process or activity is overdue. You can change the value of this property for a specific process by altering its execution characteristics.

<b>Flag Name</b>	<b>Associated wt.property</b>	<b>Short Description</b>
Notify on complete	wt.workflow.engine.notifyOnComplete	Determines whether a notification should be sent to the process responsible when the execution object completes. You can change the value of this property for a specific process by altering its execution characteristics.
Notify on past deadline	wt.workflow.engine.notifyOnPastDeadline	Determines whether a notification should be sent to the process responsible after some time has elapsed since the execution object completed. The time is defined in an attribute if the object is past the deadline.
Notify on approaching deadline	wt.workflow.engine.notifyOnApproachingDeadline	Determines whether a notification should be sent to the process responsible when the time to deadline has been reached. You can change the value of this property for a specific process by altering its execution characteristics.
Asynchronous execution	wt.workflow.engine.asynchronousExecution	Determines if the execution of an activity is asynchronous, that is, not terminated when the host process or block completes. If an asynchronous activity occurs in a loop, one may have several instances of the same activity template executing concurrently.

Flag Name	Associated wt.property	Short Description
Has dedicated queue	wt.workflow.engine.dedicatedQueueMode	<p>Determines whether the workflow processes have dedicated queues for event propagation and user work (robot expressions, synchronization, and so forth). A value of true causes the processes to have dedicated queues. Users may want a dedicated queue for a process when they want to prevent its execution from being impacted by the execution of other workflows.</p> <p>The default value is false, which means that these queues are normally shared (as are the other workflow queues).</p> <p>This attribute is used when the dedicatedQueueMode is one of propagation, userWork, or both.</p> <p>You can control this property in an individual process base by setting the appropriate execution flag in the process template.</p>
Select Latest Template Iteration	wt.workflow.engine.useLatestTemplateIteration	<p>Determines whether the latest iteration of a specific version of an object is used the first time the subprocess is encountered in the workflow. If the subprocess template is iterated and then the process loops back, the initial version (not the iterated version) continues to be used since the subprocess referenced that version.</p> <p>The default value is true, which means that the latest iteration is used the first time the subprocess is encountered.</p>

## Activity Flags

These flags are meaningful only to workflow activities.

Flag Name	Associated wt.property	Short Description
Change activity state - record	wt.workflow.engine.recordActivityStateChange	Determines if the workflow state change events that occur with activities are recorded as part of the workflow history. You can change the value of this property for a specific process by altering its execution characteristics.
Change activity state - emit	wt.workflow.engine.emitActivityStateChange	Determines if workflow state change events that occur with activities should be emitted as a keyed event. This event can then be listened to and used for synchronization and other purposes. You can change the value of this property for a specific activity by altering its execution characteristics.
Change activity data - record	wt.workflow.engine.recordActivityDataChange	Determines if the workflow change of data events that occur with activities are recorded as part of the workflow history. You can change the value of this property for a specific process by altering its execution characteristics.
Change activity data - emit	wt.workflow.engine.emitActivityDataChange	Determines if workflow change of data events that occur with activities should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific activity by altering its execution characteristics.
Change assignment - record	wt.workflow.engine.recordAssigneeChange	Determines if workflow change of assignment events should be recorded as a as part of the workflow history. The value of this property for a specific activity can be changed by altering its execution characteristics.

<b>Flag Name</b>	<b>Associated wt.property</b>	<b>Short Description</b>
Change assignment - emit	wt.workflow.engine.emitAssigneeChange	Determines if workflow change of assignment events should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific activity by altering its execution characteristics.
Ignore unresolved Role	wt.workflow.engine.ignoreUnresolvedRoles	Determines whether unresolved roles should be ignored rather than automatically assigned to the responsible role. If set to true and all roles in an activity are unresolved, no work item is generated without an indication that something went wrong. This is the global installation default behavior.

## Both Process and Activity Flags

These flags are meaningful to both workflow processes and activities.

Flag Name	Associated wt.property	Short Description
Execution error - record	wt.workflow.engine.recordException	Determines if the workflow execution exceptions should be recorded as part of the workflow history. You can change the value of this property for a specific execution object by altering its execution characteristics.
Execution error - emit	wt.workflow.engine.emitException	Determines if workflow execution exception should be emitted as a keyed event. This event can be listened to and used for synchronization and other purposes. You can change the value of this property for a specific execution object by altering its execution characteristics.
Notify on abort	wt.workflow.engine.notifyOnAbort	Determines whether a notification should be sent to the process or activity responsible when the process or activity aborts. You can change the value of this property for a specific process by altering its execution characteristics.
Abort on exception	wt.workflow.engine.abortOnException	Determines whether to abort the activity or process if an exception occurs during its execution.
Reassign on overdue	wt.workflow.engine.reassignOnOverdue	Automatically reassigns the process or activity when the process or activity is overdue. The new assignee is read from a variable named reassignPrincipal. You can change the value of this property for a specific activity by altering its execution characteristics.
Complete on overdue	wt.workflow.engine.completeOnOverdue	Determines whether an activity or process is automatically completed once it is overdue.

Flag Name	Associated wt.property	Short Description
Skip on overdue	wt.workflow.engine.skipOnOverdue	Automatically skips the process or activity when the process or activity is overdue. Note that an execution object can be skipped only if it has not been started. Also this transition needs to be as the event of an outgoing link for it to be propagated. You can change the value of this property for a specific activity or process by altering its execution characteristics.

## Modifying Execution Flags

To modify the execution flags, use the SetConfiguration tool provided.

### Running SetConfiguration

To run SetConfiguration, issue the following command:

```
java wt.clients.workflow.definer.SetConfiguration
Y:\Windchill>java wt.clients.workflow.definer.SetConfiguration
Introspection Runtime Mode = true
Windchill DataStore: Oracle

Workflow process templates
>> 1. Submit - (10003aaa)
      2. Review - (10003aaa)
      3. Problem Report Workflow - (10003aaa)
      4. Change Request Workflow - (10003aaa)
      5. Change Notice Workflow - (10003aaa)
      6. Change Activity Workflow - (10003aaa)
      7. Promotion Request Approval Process - (10003aaa)
      8. Promotion Request Review Process - (10003aaa)
      9. Approval Process - (1001faaa)
     10. Notify Process - (1005faaa)
     11. Release Process - (1005faaa)
     12. Review Process - (1005faaa)
```



13. Two Level Approval Process - (1001faaa)

14. Change Notice Workflow - (10003aaa)

Activity templates of selected process:

>> 1. Submit - (10003aaa)

Set template configuration

-----

1. Select process template
2. Show/edit process template configuration
3. Select activity template
4. Show/edit activity template configuration
5. Refresh templates
6. Exit

>>> Choose an option:

This tool shows a list of workflow templates, with the active template selected with the >> symbol. The next list is a list of the activities of the selected process templates. The active activity is shown with the >> symbol. In our example above, the Submit process template is active and the Submit activity is the active activity within the Submit process. To change the active process template, choose option 1. To change the active activity, select option 3.

To modify an exception flag, choose option 2. To modify an activity flag choose option 4. A list of flags is shown. For example if you choose 4:

```
Y:\Windchill>java wt.clients.workflow.definer.SetConfiguration
```

```
Introspection Runtime Mode = true
```

```
Windchill DataStore: Oracle
```

```
Workflow process templates
```

- ```
>> 1. Submit - (10003aaa)
    2. Review - (10003aaa)
    3. Problem Report Workflow - (10003aaa)
    4. Change Request Workflow - (10003aaa)
    5. Change Notice Workflow - (10003aaa)
    6. Change Activity Workflow - (10003aaa)
    7. Promotion Request Approval Process - (10003aaa)
    8. Promotion Request Review Process - (10003aaa)
    9. Approval Process - (1001faaa)
   10. Notify Process - (1005faaa)
```

- 11. Release Process - (1005faaa)
- 12. Review Process - (1005faaa)
- 13. Two Level Approval Process - (1001faaa)
- 14. Change Notice Workflow - (10003aaa)

Activity templates of selected process:

>> 1. Submit - (10003aaa)

Set template configuration

-----

- 1. Select process template
- 2. Show/edit process template configuration
- 3. Select activity template
- 4. Show/edit activity template configuration
- 5. Refresh templates
- 6. Exit

>>> Choose an option:

Selecting an option toggles the value to either true or false.

## Saving Your Work

To save the configuration, select either option 30 or 31. Make sure that the relevant workflow process template is checked out. Also, make sure that all existing instances of the relevant process template are terminated. If one of these conditions is not met, an exception is thrown.

New workflow processes will use the new behavior defined by these execution flags.

## Using Task Form Templates in a Workflow

A task form template contains general formatting information, content and variables such as form fields, labels, and text fields which are place holders to display the attributes of a task. These attributes are displayed to a task assignee when a task activity is executed through a workflow process. The template can be rendered in the following ways:

- In an HTML format when the task form template has an associated JSP template file (this is the default format).

You can create the required JSP file by identifying the task variables through the Workflow Administrator and then using any standard Java editing tool to create the JSP file.

- In a PDF format when the task form template has an associated XDP template file.

You can create the required XDP file by using the Adobe Designer software with input in the form of an XML file generated from the Workflow Administrator. The Adobe Designer software is purchased separately from Windchill.

For detailed procedures for rendering a workflow task as either a PDF or in HTML format, see the Workflow Administrator online help.

## Creating Task Form Templates with Adobe Forms Software

As part of the process of modifying or creating a workflow template, you can modify or create task form templates.

The Adobe Designer software can be installed and configured for use with Windchill, allowing forms to be utilized in workflow processes. (For installation and configuration instructions, see the *Windchill Installation and Configuration Guide - Advanced*.) Custom forms can be created in the Adobe Designer software, uploaded into Windchill, and associated to a workflow process as a Task Form Template. The form is distributed through the workflow process as a task and can be rendered as a PDF or the default HTML format. During the completion of a workflow task, a user fills in a form with the required information and the workflow system updates the Windchill system with the acquired information.

If the Adobe forms software is not installed and configured for use with Windchill, the workflow tasks are rendered in the HTML default format, using the JSP template type.

The Adobe Designer software creates templates in the XDP template type and contains various form fields like labels, text fields, and radio buttons that act as place holders to display the workflow task attributes like variables, process name, etc.

The attributes of the workflow tasks are stored in an XML file. The XML file is displayed when a user clicks the **Generate XML** button on the **Activity** tab of an **Assigned Activity** process node. The Adobe Designer software is used to create a form template (XDP type) and a connection between the form template and the previously saved XML file is made to define each field association.

An application context manager can create forms with the Adobe Designer software and upload them into Windchill as a Task Form Template. The forms are managed from the **Template** sub-navigation link within the respective context; Site, Organization, Project, or Product. For more information about templates, see the Templates online help.

For information on creating customized task form template files (both JSP and XDP) and associating with a workflow process template, see the *Windchill Customizer's Guide*.

**Note:** PTC provides the Default task form template which is used with the workflow activities that are defined in the out-of-the-box workflow templates. If you are using the out-of-the-box workflow templates, do not delete this task form template.

## Electronic Signatures

The Windchill products allow you to require electronic signatures on workflow activities to authenticate the activity.

### Setting Up for Electronic Signatures

If the eSignatures are in an LDAP directory that has already been established for Windchill, there is no additional setup required. If the LDAP directory is not used with Windchill, see the *Windchill Installation and Configuration Guide - Advanced* for information about creating a JNDI adapter entry and repository description. You must identify the directory in the `wt.org.electronicIdentity.authenticationService` property.

From a windchill shell, execute the following commands:

- To display the current value of the property:

```
xconfmanager -d  
wt.org.electronicIdentity.authenticationService
```

- Specify the existing and new value (append new value to the existing property value). You can specify one or more JNDI adapter Service Names. Use a comma to separate the adapter names. See the xconfmanager guidelines for specifying multiple property and property value combinations:

```
xconfmanager -s  
wt.org.electronicIdentity.authenticationService=<JNDIadapter  
service name(s)> -t <Windchill>/codebase/wt.properties -p
```

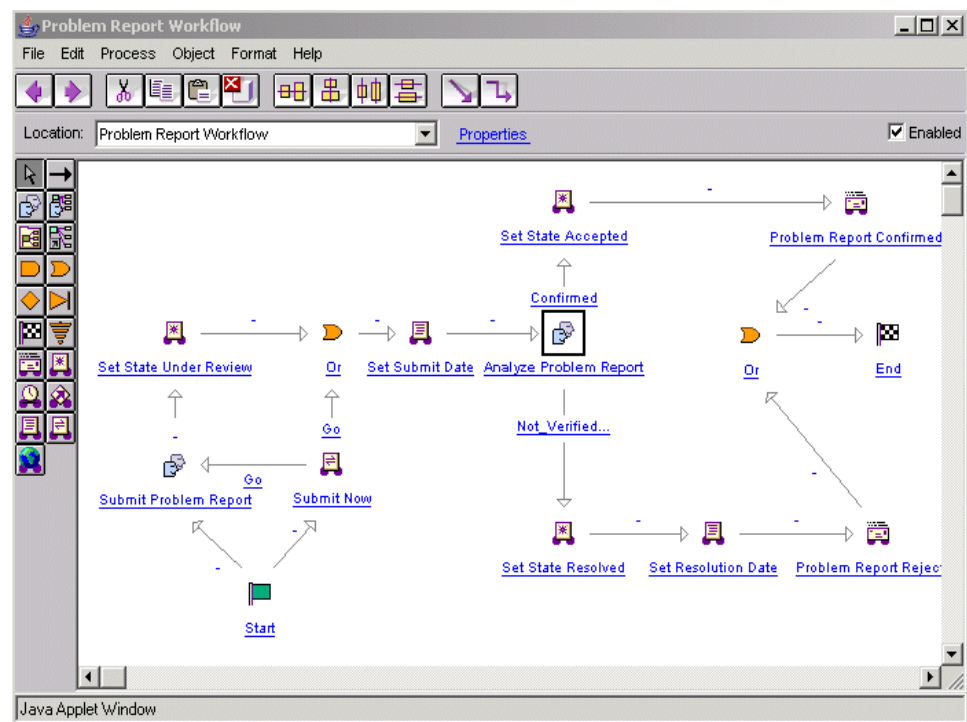
Where `<Windchill>` is the location where Windchill is installed.

- The parameter `wt.org.electronicIdentification.class` can be set to determine the authentication required. The parameter can be set as follows:
  - `wt.org.electronicIdentity.engines.LDAPPasswordSignatureEngine` – this signature engine only requires a password to be supplied. It verifies that the password belongs to the currently logged in user.
  - `wt.org.electronicIdentity.engines.LDAPFDACompliantSignatureEngine` – this signature engine requires both a user name and a password field to be entered, in compliance with FDA rule 21 CFR part 11 section 11.2. It verifies that the user name and password belong to the currently logged in user.

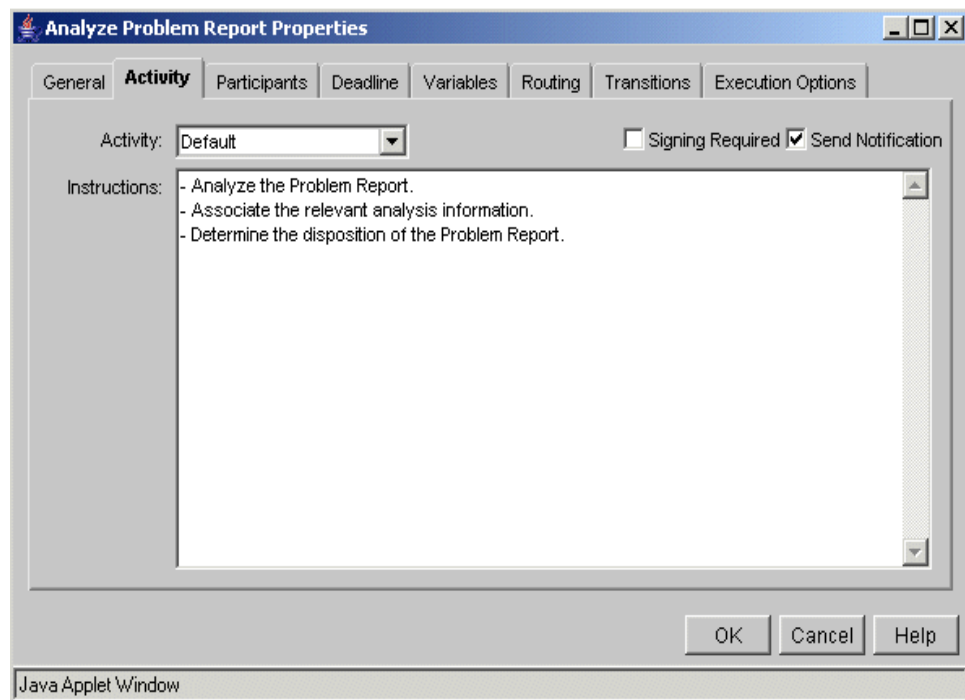
## Requiring Electronic Signatures in a Workflow

To require electronic signatures, perform the following steps:

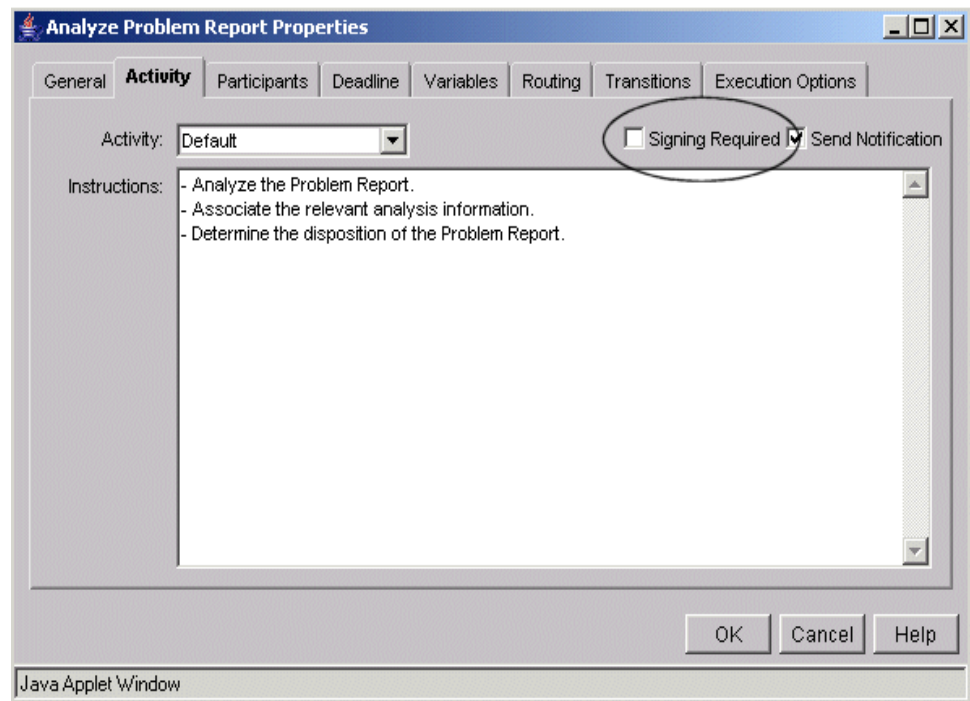
1. Use Workflow Administrator to make an electronic signature required for a particular activity. When you create or edit a process template, the process window appears.



2. If you double-click the activity or right-click the activity and select **Properties**, a properties window appears. In this window, select the **Activity** tab.



3. Select the **Signing Required** check box to require an electronic signature at this point in the process.



**Note:** If you select **Signing Required**, you are prompted for authentication in order to start the activity.

## Best Practices

The following sections contain best practices for Workflow templates and processes.

### Access Control

Access control is generally defined, using the Policy Administrator, on the domain to which the object belongs. However, it is possible to specifically add additional access policies to a given Windchill object (such as a specific WTPart or WTDocument). For example, you can use ad hoc access policies inside Windchill workflow, or you can define object-level access policies in Windchill ProjectLink.

If you are likely to have an extremely high number of running workflow activities referencing the same object, make sure that you grant ad hoc access only as needed. Use Policy (static) access control where appropriate. For more information, see the [Access Control](#) chapter.

## Using a Single Workflow in a Life Cycle Having Multiple States

Duplicate tasks get started when a set state robot is used along with a loop link to the task related to the new state.

In order to avoid this situation, create an initial dummy life cycle phase that is associated with the workflow and that represents a state that is never returned to in a loop or in a set state robot.

**Note:** When you assign a workflow template to a life cycle template, you see a list of valid workflows. The list of valid workflow templates includes the ones defined in the given application context, plus those defined in the ancestor organization and site contexts. Workflow templates defined in a sub-context override and filter out the workflow templates defined in parent contexts having the same name.

## Workflow Process Support in Windchill ProjectLink

The following table shows the workflow process support in Windchill ProjectLink.

| Feature                                   | Support Level | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use of Workflow Administrator's interface | Partial       | Site and organization administrators can see all templates in the context from which they launched the Workflow Administrator and those inherited from the ancestor context. In Windchill ProjectLink, you can launch the Workflow Administrator from the <b>Utilities</b> page on the Organization and the Site tabs. The Workflow Administrator is not available on the <b>Project</b> or <b>Program</b> tab to administrators of project or program contexts. From the Organization context, you will see the templates from the site and the organization context from which the applet was launched. From the organization context, the site templates can only be viewed or saved. |
| Use of workflow process editor interface  | Full          | The full capabilities of the process editor are exposed. There is no bootstrap capability with Windchill ProjectLink; so a high bandwidth connection to the server is necessary to stream the workflow process editor applet to the client.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Default task choices                      | Partial       | Life cycle and default task types are supported. Setup participant task types and change tasks are not supported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| E-mail task delivery                      | Full          | All tasks can be delivered by e-mail, in addition to being added to the user's assignment table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |



| Feature                              | Support Level | Notes                                                                                                                                                                                           |
|--------------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ad hoc task support                  | None          | There is no support for ad hoc workflow tasks.                                                                                                                                                  |
| Custom task form used and definition | None          | Windchill ProjectLink task forms are rendered as JSP pages. Custom task forms can be created using Adobe Forms software. See <a href="#">Using Task Form Templates in a Workflow</a> .          |
| Electronic signature support         | Full          | Windchill ProjectLink requires additional authentication, as specified by the signature engine, for tasks generated from activities that require signing.                                       |
| Activity-based access control        | Partial       | Activity-based access control may be used for the primary business object only. Access right may only be added to the rights currently defined for the object. Access rights cannot be removed. |
| Variable support                     | Partial       | All standard variables are supported. Variables that represent Windchill classes can be defined and used, but will not be rendered in the task for user interface.                              |
| Routing                              | Full          | Exclusive and nonexclusive manual routing is supported. Automatic routing is supported via expressions. Routing selections are displayed only for required roles.                               |
| Robots                               | Full          | Notification, set state, timer, execution, and expression robots are all supported.                                                                                                             |
| Connectors                           | Full          | Start, stop, AND, OR, and threshold connectors are all supported.                                                                                                                               |
| Process references                   | Full          | Only processes defined for the organization associated with the logged-in user may be referenced.                                                                                               |
| Blocks                               | Full          |                                                                                                                                                                                                 |
| Transition conditions                | Full          |                                                                                                                                                                                                 |
| Roles                                | Partial       | Roles that are referenced must be satisfied in the project or program instance associated with the process instance.                                                                            |
| Voting                               | Full          |                                                                                                                                                                                                 |
| Variable substitution                | Full          | Variable support is limited. Variables representing Windchill classes can be defined and used, but will not be displayed in the task pages.                                                     |

| <b>Feature</b>                         | <b>Support Level</b> | <b>Notes</b>                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Import/export                          | Partial              | Process templates can be exported to XML format. There is no organization association with the imported or exported template so the workflow import/export folder should be cleared between uses.                                                                                                                                                                                                                   |
| Process initiation                     | Partial              | A workflow process may be instantiated indirectly only by creating a life cycle managed object or by initiating a routing action. A workflow may also be started programmatically. There is no user interface to explicitly start a workflow process in Windchill ProjectLink.                                                                                                                                      |
| Participation identification           | Partial              | Participation in a downstream task can be accomplished through the use of user and group variables that are satisfied by an assignee in a task form. You can add or remove assignees from a running process by updating the route participants. The change applies to unstarted started activities, it will not affect tasks that are currently running. There is no support for the setup participants capability. |
| Content holder access                  | None                 | There is no workflow content holder exposed in a workflow task.                                                                                                                                                                                                                                                                                                                                                     |
| Ad hoc tasks                           | None                 | Ad hoc tasks cannot be created from a parent task.                                                                                                                                                                                                                                                                                                                                                                  |
| Access to primary business object      | Partial              | A link to the subject is provided.                                                                                                                                                                                                                                                                                                                                                                                  |
| Adding comments                        | Partial              | A link to the subject's discussion forum is available in the task form. Comments can also be added to a string variable.                                                                                                                                                                                                                                                                                            |
| Task actions                           | Partial              | A user can open and reassign task items, but there is no capability to delegate them or to access a user calendar. When tasks are reassigned, the assignee pool is the project or program team.                                                                                                                                                                                                                     |
| Assignments customization              | None                 | The Assignments tables support multiple predefined views. There is no capability to further define custom views.                                                                                                                                                                                                                                                                                                    |
| Project or Program assignments view    | Full                 | There is a specific view of all assignments in a project or program.                                                                                                                                                                                                                                                                                                                                                |
| Access to the Workflow Process Manager | Partial              | There is no access to the process manager for a general workflow user.                                                                                                                                                                                                                                                                                                                                              |

| <b>Feature</b>                                        | <b>Support Level</b> | <b>Notes</b>                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Access to workflow participation or execution history | Partial              | Windchill ProjectLink exposes no explicit access to workflow participation or history information for non-administrators. However, individual task forms can accumulate information regarding participation and disposition. Comment history can be captured in a discussion forum associated with the primary business object. |
| Local search support for process information          | None                 | There is no support for searching running or completed processes.                                                                                                                                                                                                                                                               |
| Workflow process management                           | Partial              |                                                                                                                                                                                                                                                                                                                                 |



# 17

## Views and View Associations

This chapter provides information about views and how to manage them using the View Administrator.

| Topic                                     | Page |
|-------------------------------------------|------|
| Overview .....                            | 17-2 |
| Views and View Associations .....         | 17-2 |
| Managing Views and View Associations..... | 17-3 |

## Overview

The View Administrator utility allows the site administrator to create and modify the product structure views for use in Windchill PDMLink. The view is then available in Windchill PDMLink for the end user during the Create Part action.

Within Windchill, a part is assigned to a view when it is created. A part may be a *view-dependent* object, if several versions of the part are required to address the needs of the various organizations working on it. For example, the Engineering and Manufacturing departments may want to work on different versions of a part, with each version representing a view specific to that department's needs.

### Accessing the View Administrator

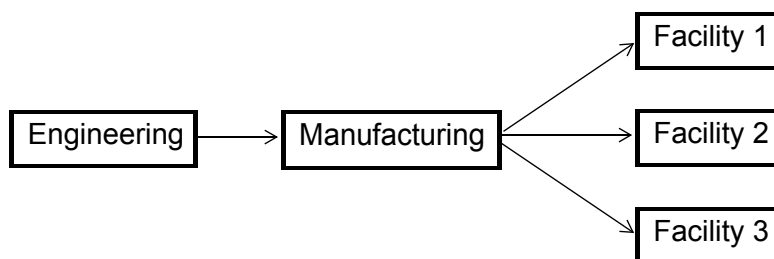
The View Administrator utility can only be accessed by a site administrator. If other users enter the URL manually, an error message will display and the View Administrator utility will not launch.

From the Site tab in Windchill PDMLink, click the **Utilities** link. In the **Business Administration** list, click the **View Administrator** link.

## Views and View Associations

Before a part can be assigned to a view, you must set up views and view associations for your Windchill system. Each view you configure must have a unique name. A view can have many child views, but only one parent view.

Only the first view in the structure has no parent view. You can define only one such *root view* for your site. The following diagram provides an example of a typical view setup:



View-dependent versions of a part can be derived from parent views. As shown in the diagram above, the Engineering view is the first view in the structure and is the parent of the Manufacturing view. Also, the Manufacturing view is the parent of the Facility 1, Facility 2, and Facility 3 views. Therefore, a Manufacturing version of the part can be built from the Engineering version of the part. Facility 1, Facility 2, and Facility 3 views can be derived from Manufacturing.

When a user creates a view-dependent version of a part, the new version is assigned an initial revision letter, which is prefixed with the revision letter of the part version from the parent view. For example, if the Engineering version of a part is Revision B, the Manufacturing version of the part is labeled Revision B.A. Each view-dependent version of a part goes through its own life cycle process.

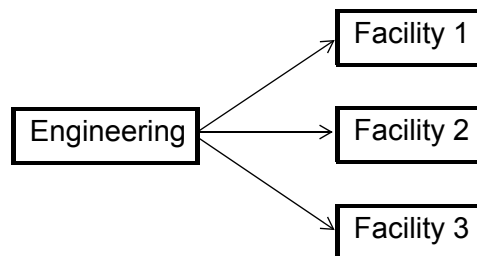
## Managing Views and View Associations

Views can be managed by using the View Administrator utility. The online help provides detailed procedures for creating, inserting, renaming, and moving views.

Access the online help by launching the View Administrator utility from the **Site** tab.

**Note:** After parts have been created in a view, you cannot delete the view. Attempting to delete a view that has associated parts returns an error message.

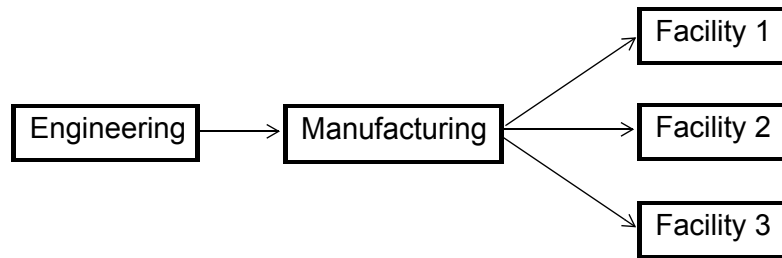
When you delete a view, all child views of the deleted view become child views of the parent of the deleted view. For example, if you delete the Manufacturing view from the previous example shown in the [Views and View Associations](#) section, the Facility 1, Facility 2, and Facility 3 views become child views of Engineering as shown in the following diagram:



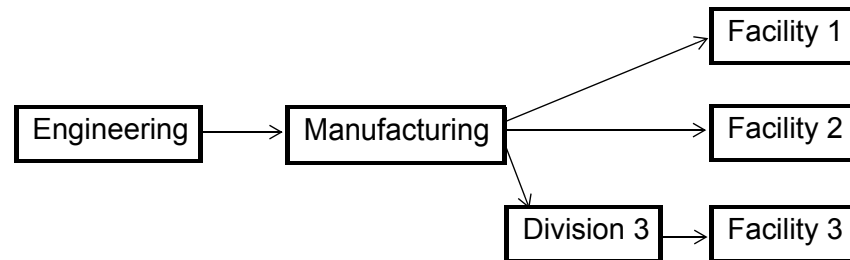
When inserting a view, if you provide values for both the parent name and the child name, the inserted view becomes a child of parent view and parent of the child view.

When inserting a view, a new view is inserted as a child, directly under the selected view in the hierarchy. The children of the existing parent view are automatically made children of the inserted view.

For example, assume the following views exist (as described earlier in this section):



If you specify **Manufacturing** as the parent view and **Facility 3** as its child, and then insert a **Division 3** view. The Division 3 view would be a child of Manufacturing and new parent of Facility 3, appearing in between the two as shown in the following diagram:





# 18

## Indexing

This chapter provides information on indexing within Windchill solutions.

| Topic                                     | Page |
|-------------------------------------------|------|
| Overview .....                            | 18-2 |
| About Indexing Rules.....                 | 18-2 |
| Creating and Managing Indexing Rules..... | 18-3 |
| About Indexing Policy.....                | 18-4 |
| About Indexing Lists .....                | 18-4 |
| Defining a Collection .....               | 18-5 |

## Overview

*Indexing* is the process of extracting text strings of attribute names and attribute values from Windchill objects and sending them to a search engine that builds indices optimized for searching. This enables users to efficiently search for data stored in a Windchill database, without having to know anything about the internal object model.

Windchill solutions provide the option of installing Windchill Index Search to help with indexing. For additional information about Windchill Index Search and its indexing capabilities, see the *Windchill Installation and Configuration Guide - Advanced*.

## About Indexing Rules

Creating an indexing rule from within the Policy Administrator requires you to specify the rule antecedent and the rule consequent. The rule antecedent comprises the following parts:

- The *domain*.
- The *object type*-- determines which rules within an indexing policy apply to a specific object.
- The *life cycle state*, identifies the life cycle phase that an object must be in for a rule to apply.
- The *collections* into which objects are to be entered, when the objects belong to the domain, are of the type, and are in the life cycle state specified by the rule.

For example, you can define a rule specifying that a general document object is to be placed in a Released collection when the object's state becomes Released. Together, the domain indexing rules form the indexing policy for a domain.

The rule consequent is a list of one or more collections.

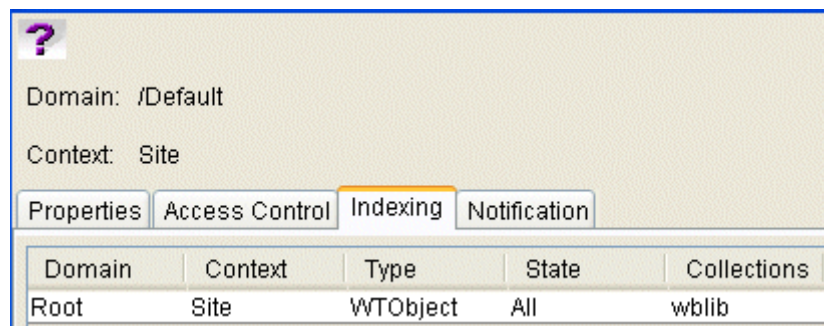
A *collection* represents a group of related objects that can be searched. It includes indices optimized for searching, as well as references to the actual object locations. Every indexable object carries a list of collections into which it is indexed. The first such list is assigned when the object is created. When the object is deleted, it must be withdrawn from every collection in which it is indexed. Between creation and deletion, the collections in which the object is indexed can change, based on the object's life cycle state and which domain it belongs to. When you create indexing rules, you customize the indexing policy for a domain by specifying which collections an object should move into (or be removed from) when the object moves into a specified life cycle state. From this policy, indexing lists are generated and associated with an object type. To improve performance, indexing lists are cached after they are created.

An *indexing rule* identifies a life cycle state for a particular object type and the collections into which the object should be entered, based on the state that it is in. There can be only one state and one object type specified within a single rule. However, each rule can identify multiple collections.

An *object type* specifies a category of objects that share the same attributes and functions. For example, WTDocument is an object type, and instances of that type may be found in some of the domains you have created. Since Windchill domains are hierarchical, indexing rules defined for a domain are inherited by descendent domains. For example, indexing rules defined for the WTDocument object type in all states within the Design domain apply to instances of the type within that domain or any descendent domains. Because Windchill types are also hierarchical, an object inherits rules defined for its ancestor types. Therefore, more than one rule may apply to a given object. For example, a rule that applies to the type AnnotationSet also applies to the type StructuredAnnotationSet. Additionally, there can be indexing rules specific to StructuredAnnotationSet.

## Creating and Managing Indexing Rules

Use the Policy Administrator to create and manage indexing rules. Indexing rules are created and managed for a domain that is within a specific context as described in the [Administering Domains and Policies](#) section of the Administering Containers chapter. Open the Policy Administrator from the context of the domain where you want the indexing rules to apply. Select the domain and click **Update**. On the **Administrative Domain** window, click the **Indexing** tab to bring it forward.



Click the help icon located in the upper left hand corner of the window for specific instructions on retrieving, creating, updating, deleting, and reporting on indexing rules.

## About Indexing Policy

This section describes the Windchill implementation of indexing policies.

### Considerations for Establishing Indexing Rules

As you create an indexing policy, you may find it helpful to answer the following questions:

- Which life cycle states are associated with the most changes in business objects? You could decide to create rules that affect objects in the In Work state. Or, you might decide to index objects in the Under Review or the Released state.
- Do certain domains contain objects that are more dynamic than others or objects that users are more likely to search for?
- How do different types of objects undergo change in the system?
- How might rules differ for object types that are versioned (for example, document types) and object types that are not versioned (for example, change objects)?

## About Indexing Lists

Indexing lists are generated for each object type, state, and domain. Objects are associated with the indexing list of the domain, type, and state to which they belong. For example, all WTPart objects in a given state and domain are associated with the same indexing list. The indexing list for WTPart objects is different from the list associated with other types of objects (for example, WTDocument objects) that are in the same state and belong to the same domain.

An indexing list for an object is obtained by combining all rules that apply to that object based on its type, state, and the domain to which it belongs.

A rule is applicable to a given type when the object type referred to in the indexing rule is the type itself or one of its ancestor types. For example, if IncidentReport is a soft type of the type WTObject, then a rule that applies to WTObject also applies to IncidentReport. In addition, rules are inherited from ancestor domains.

As described above, this type and domain hierarchy means that more than one collection may have to be updated when a specific event occurs. Consider the following example:

|         | Domain        | Context           | Type           | State | Collections          |
|---------|---------------|-------------------|----------------|-------|----------------------|
| Rule 1: | /             | Site              | IncidentReport | ALL   | Current, Assignments |
| Rule 2: | /Publications | Demo Organization | WTOBJECT       | ALL   | Assignments          |
| Rule 2: | /Publications | Demo Organization | IncidentReport | ALL   | CustomerX            |

The combination of these rules produces the following index list entry for IncidentReport in /Publications:

| Domain        | Context           | Type           | State | Collection(s)                   |
|---------------|-------------------|----------------|-------|---------------------------------|
| /Publications | Demo Organization | IncidentReport | ALL   | Current, Assignments, CustomerX |

As this list entry specifies, the Current, Assignments, and CustomerX collections must be updated whenever an incident report is created or modified that is associated with /Publications domain, which is in the Demo Organization context, regardless of the life cycle state it enters.

When you have defined the indexing rules for your domains, all of the objects with the same domain, type, and state combination for which you have created a rule share an indexing list.

This association between the indexing list and the object is preserved.

## Defining a Collection

*Indexing* is the process of extracting text strings of attribute names and attribute values from Windchill objects and sending them to a search engine that builds index *collections* optimized for searching. This enables users to efficiently search for data stored in a Windchill database without having to know anything about the internal object model.

Windchill collections are defined in the wt.properties file. Each collection has properties that define the collection. For more information, see the *Windchill Installation and Configuration Guide - Advanced*.



# 19

## Notifications

This chapter provides information on Windchill notifications, including notification rules, notification lists, and subscriptions.

| Topic                                              | Page |
|----------------------------------------------------|------|
| Overview .....                                     | 19-2 |
| About Notification Rules.....                      | 19-2 |
| Creating and Managing Notification Rules.....      | 19-3 |
| Notification Lists.....                            | 19-4 |
| Object-based and Context-based Subscriptions ..... | 19-5 |
| About Notification.....                            | 19-6 |

## Overview

A notification policy determines who is notified when events of interest happen in the system.

**Note:** Notifications are also generated as part of workflow processing. Workflow notifications can be more task specific. For more information, see the chapter entitled [Workflow](#).

## About Notification Rules

When you create notification rules from within the Policy Administrator, you specify who is to be informed when a given system event occurs within the context of a specific object. You can construct a rule for a domain, an object type, and an event posted by a manager type (ownership, locking, versioning, life cycle, and so on). The set of notification rules for a domain constitutes the notification policy for that domain.

Actual notification is accomplished by sending an e-mail message to the users on the notification list. This message identifies the event and the object.

**Note:** A user must have Read permission to an object to receive notification of an event applied to that object.

Creating a notification rule requires you to specify the rule antecedent and the rule consequent. The rule antecedent comprises the following parts:

- The domain
- The object type, which determines which rules within a notification policy apply to a specific object
- The system event type (for example, Checkout).

The rule consequent is a list of one or more principals.

A *notification rule* for a domain identifies a system event of interest for a particular object type and determines which users, groups, and organizations should be notified when one of those events occurs, for example, checkout. There can be only one event type and one object type specified within a single rule. However, each rule can identify multiple principals.

An *object type* specifies a category of objects that share the same attributes and functions. For example, WTDocument is an object type, and instances of that type may be found in some of the domains you have created. Since Windchill domains are hierarchical, notification rules defined for a domain are inherited by descendent domains. For example, notification rules defined for the WTDocument object type in all states within the Design domain apply to instances of the type within that domain or any descendent domains.

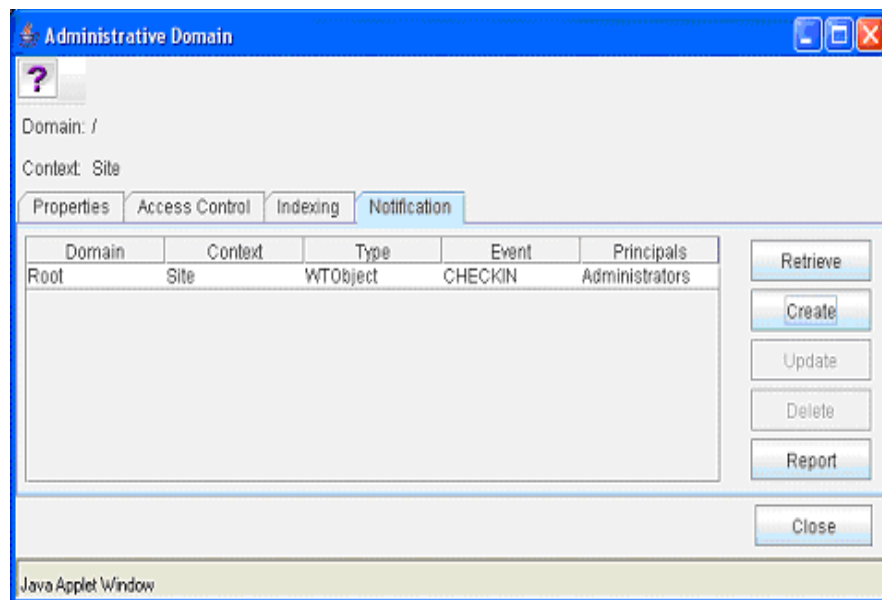


Because Windchill types are also hierarchical, an object inherits rules defined for its ancestor types. Therefore, more than one rule may apply to a given object. For example, a rule that applies to the type `AnnotationSet` also applies to the type `StructuredAnnotationSet`. Additionally, there can be notification rules specific to `StructuredAnnotationSet`.

A *principal* is either an individual user, a group, or an organization. Typically, you should define notification rules for groups. Dealing with groups helps reduce administrative overhead by making it possible to apply rules to multiple users at the same time, enabling mass mailings for notification. Sometimes, however, you need to create rules specific to an individual user or to an entire organization.

## Creating and Managing Notification Rules

Use the Policy Administrator to create and manage notification rules. Notification rules are created and managed for a domain within a specific context as described in the [Administering Domains and Policies](#) section of the Administering Containers chapter. Open the Policy Administrator from the context of the domain where you want the notification rules to apply. Select the domain and click **Update**. In the **Administrative Domain** window, click the **Notification** tab to bring it forward.



Click the help icon located in the upper right hand corner of the window for specific instructions on retrieving, creating, updating, deleting, and reporting on notification rules.

## Notification Lists

Notification lists are generated from the notification rules for a domain and its ancestor domains. These lists are the basic mechanism for initiating user notification when an event occurs within the context of a specific object within the domain. For performance reasons, once lists are constructed, they are kept in a cache.

Notification lists are generated for every combination of type, event, and domain. A notification list for an object that is the target of an event is obtained by combining all rules that apply to the event, the domain to which the object belongs, and the type of object. For example, the same notification list applies to the Create event for all WTDocument objects within a given domain. In addition, this notification list can be different from the list associated with WTPart objects, even if the objects belong to the same domain.

To make this definition precise, it is necessary to describe how rules are combined and when a rule applies to a type.

A rule is applicable to a given type when the object type referred to in the notification rule is the type itself or one of its ancestor types. For example, a rule that applies to the WLObject type also applies to the IncidentReport type if IncidentReport is a soft type of WLObject.

This type of hierarchy, in addition to the domain hierarchy, means that more than one set of principals may need to be notified when a specific event is applied to an object.

For example, consider the combination of the following rules:

|         | Domain        | Context           | Type           | Event  | Principal(s)         |
|---------|---------------|-------------------|----------------|--------|----------------------|
| Rule 1: | /             | Site              | WLObject       | Create | Marketing, Engineers |
| Rule 2: | /Publications | Demo Organization | WLObject       | ALL    | Amanda.Smith         |
| Rule 3: | /Publications | Demo Organization | IncidentReport | Create | Support              |

The combination of these rules produces the following notification list entry for IncidentReports created in the /Publications domain:

| Domain        | Context           | Type           | Event  | Principal(s)                                |
|---------------|-------------------|----------------|--------|---------------------------------------------|
| /Publications | Demo Organization | IncidentReport | Create | Marketing, Engineers, Support, Amanda.Smith |

As this list entry specifies, all members of the Marketing, Engineers, and Support groups, and also user Amanda Smith, must be notified whenever an incident report is created in the /Publications domain. This result is based on the following application of the notification rules:

1. Members of the Support group are to be notified when an incident report is created in the /Publications domain.
2. Members of the Marketing and Engineers groups are to be notified when an object of type WObject is created in the Root domain. IncidentReport is a soft type of WObject, and the Root domain is an ancestor of /Publications, so this rule also applies to the incident report created in the /Publications domain.
3. User Amanda.Smith is to be notified of all events that occur to an object of type WObject in the /Publications domain, which includes any event that occurs to an object of the soft type IncidentReport in the /Publications domain.

When you have defined the notification rules for your domains, a single notification list is shared by all events with the same event type, target object type, and target object domain. When an event occurs to an object, the notification list associated with the object is retrieved, and the appropriate principals are notified of the event.

## Object-based and Context-based Subscriptions

Users can have context-based subscriptions and object-based subscriptions. For example, users can subscribe to a project, program, product, or library, or they can subscribe to a folder. A user can also subscribe to a specific object like a document or part.

For more information about user subscriptions, see the Windchill user's guides for the Windchill solutions that are installed.

## About Notification

A property exists in `wt.properties` called `wt.notify.subscriptionConfigFiles`. This property contains all the supported subscription configuration files and their location. The default file can be found at `<Windchill>/codebase/wt/notify/subscriptionConfig.xml`.

The XML specified in the file contains the following sections:

- Type to Events list

This section lists all of the events the Notification Policy Manager must subscribe to and listen for.

This section applies to both notification rules and user subscriptions.

- Category to Events list

This section lists the events associated with each category. As a reminder, one or more events are assigned to a category. Categories are a way to group related events for a subscription in order to simplify the subscription user interface.

This section applies only to user subscriptions.

Additional information exists for each category:

- Resource name

The path of the resource bundle.

- Subscription type

Specifies if the associated information is for an object-type subscription or context-type subscription.

The value can be either Object or Context and the default is Object.

If the subscription is not for the type specified, then the given category is not displayed. For example, if user subscribes to a specific document (the subscription is an object-based subscription) and the category, `NEW_OBJECT`, has subscription type = Context, then `NEW_OBJECT` will not be displayed as 1 of the categories for document.

- Exclude type

Exclude the given category if the type is that specified or a parent/child relationship exists with the type specified.

- Override type

Only those types specified under the `override_type` section will display the given category.

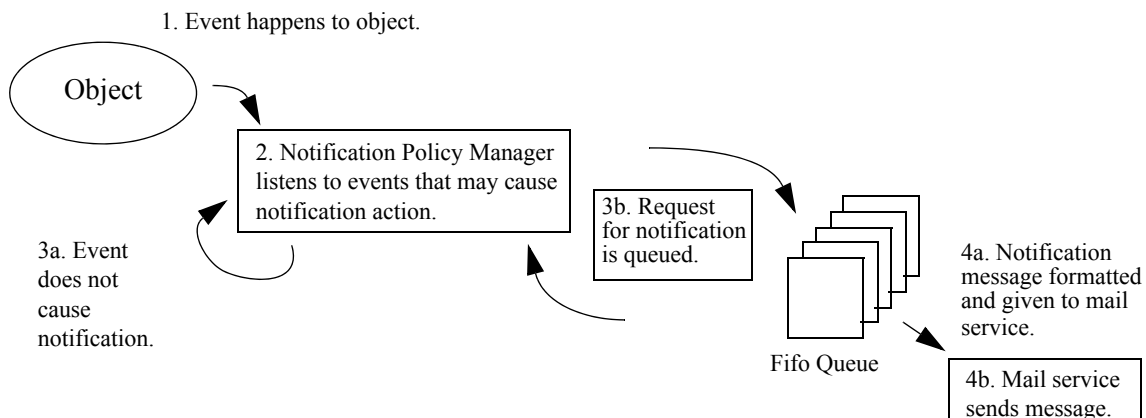
- Migrated Event to Category list

This section is used if your system is being upgraded from a pre Windchill 9.0 system. During the upgrade of your system, all subscribed subscription events are converted to subscription categories.

This section applies only to user subscriptions.

**Note:** Changing of this file is not supported.

The following figure represents an overview of the notification process:



- An event happens to an object (step 1). If it is an event for which the Notification Policy Manager is listening, the event and the object are posted to the manager (step 2).
- The Notification Policy Manager checks to determine whether the event triggers a notification action. In many cases it does not, and can be ignored (step 3a). (For example, if the object does not belong to a type subject to notification rules, or there is no list for the domain/type/event.)

In some cases, there is a notification list for the domain/type/event. When a list exists, the manager does not try to send the notification immediately. Rather, it queues the notification request for deferred processing in a FIFO queue (step 3b).

- Later, the queued requests are asynchronously executed. These requests translate into calls, so that the notification message is formatted and then mailed using the mail service (steps 4a and 4b).

For more information about background processing queues and their maintenance, see the *Windchill System Administrator's Guide*.



# 20

## Visualization Services

This chapter has an overview of the Windchill Visualization Service (WVS), providing information on WVS functionality and architecture. Included is a troubleshooting section that you can use as well as additional administrator information about watermarks, the Copy Forward functionality, and Publish Rules. Also included is a description of important WVS properties.

| <b>Topic</b>                                       | <b>Page</b> |
|----------------------------------------------------|-------------|
| Overview .....                                     | 20-2        |
| Architecture .....                                 | 20-3        |
| Troubleshooting.....                               | 20-6        |
| Visualization Collaboration.....                   | 20-12       |
| Creating and Enabling Server-side Watermarks ..... | 20-13       |
| Exporting Watermarks for ProductView.....          | 20-18       |
| Copy Forward Functionality .....                   | 20-19       |
| Using Publish Rules .....                          | 20-23       |
| Windchill Visualization Service Properties .....   | 20-45       |

# Overview

This section is an overview of WVS functionality and architecture, providing a context for the troubleshooting guidelines later in this chapter.

## File Types

ProductView can display many file types. However, it is important that you are familiar with the following four file types:

- OL files

An OL file is a binary file, which is created by publishing a CAD part. It contains the 3D and 2D CAD information. A single CAD part may create many OL files.

- PLT files

A PLT file contains 2D-vector information, and is created when drawing output is requested during publication of a CAD part. A CAD part can produce many PLT files.

- PVS files

A PVS file is a binary file that contains product structure information for ProductView.

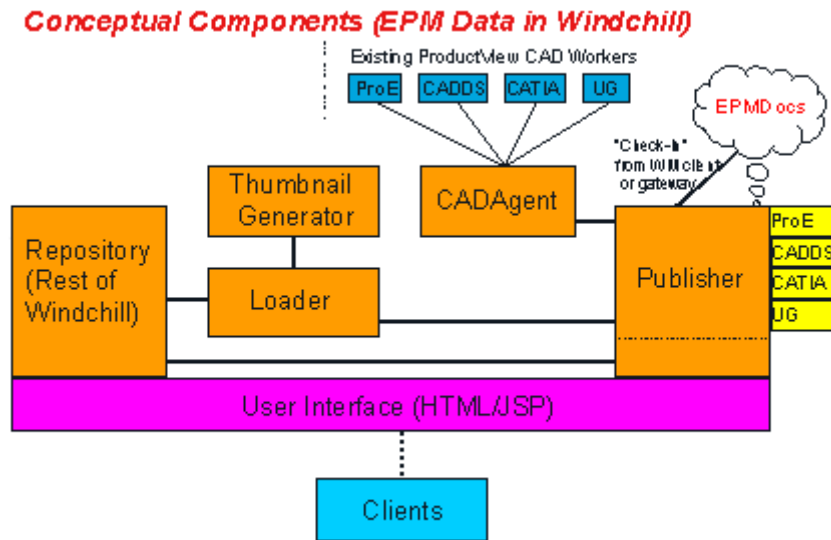
- PVZ files

A PVZ file is a ZIP file containing PVS, OL, and other ProductView files. The PVZ acts as a snapshot of a product structure. As it is a single file, it provides a faster way to access a large amount of information, as only one download allows you to view a complete product. The PVZ also provides a single file that can be easily exchanged (for example, through e-mail).



# Architecture

The following figure is a graphical representation of the Windchill Visualization Structure architecture.



WVS allows Windchill users to generate viewable files, store those files in the Windchill database, and view data in ProductView. ProductView displays many document formats directly from the file, requiring no preparation. However, CAD data must be published before it can be viewed in ProductView. WVS also allows you to generate non-CAD data, such as dynamic documents and general documents.

The loader is responsible for preparing data for storage in Windchill before it is converted. The loader can be used in two ways:

- As a Windchill service, which looks for ticket files in a directory.  
*A ticket file is a text file that defines the location of the preconverted data and specifies the way in which it will be catalogued when stored in Windchill. For details on using the CAD Agent for publishing, see the *Windchill System Administrator's Guide*.*
- As an operation called directly by the publisher.  
Calls from the publisher are performed programmatically. The data is handled in the same way as if it were loaded through a ticket.

The loader can optionally call the thumbnail generator to create a JPG image and 3D thumbnail file of the 3D geometry. If required, the thumbnail generator can be configured by its recipe file to only create a JPG image. To use this functionality, the ProductView Thumbnail Generator must be installed. For information on installing the generator, see the *Windchill Installation and Configuration Guide - Advanced*.

Data stored in Windchill by the loader is stored on a Representation, which is associated with a Representable (currently a WTPart, WTDocument, EPMDocument, or DynamicDocument).

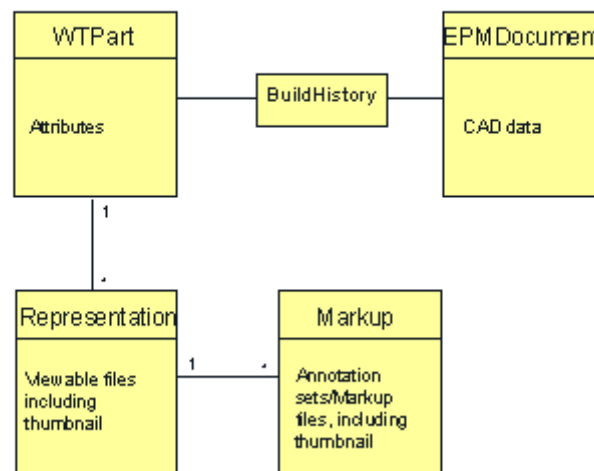
The loader can also create an PVZ file, which is stored as role PRODUCT\_VIEW\_EDZ on the Representation. All the data files are loaded as secondary content on the Representation. The PVS (product structure) file is processed, so the references to the viewable files point to Windchill content through URLs. The PVS file is then stored as role PRODUCT\_VIEW\_ED on the Representation.

If a thumbnail image or 3D thumbnail file has been created, it is stored on the Representation as content role THUMBNAIL/THUMBNAIL3D. If this is the default Representation, the Representable also receives a copy of the thumbnail content. The actual content in the database is not duplicated, if it is shared with the Representation.


From any visualization link, you can view the data stored on a Representation in Windchill within ProductView. ProductView can author Windchill information in the form of annotations and groups saved as a markup in the database. Markups are associated to a particular Representation.

The following figure shows the conceptual Windchill visualization data model:

### Persistent Server Objects



If no WTParts are created in Windchill through EPM operations, the Representation is associated to the EPMDocument. Where the CAD data has been stored in Windchill by an EPM client, (for example, the Workgroup Manager for Pro/ENGINEER) or is referenced by an EPM gateway (for example, the Pro/INTRALINK Gateway), WVS can publish and store the data. The publishing of data can be accomplished by:

- Using the Create Representation wizard.
- Having a user click the publish icon .
- Checking in data from the EPM client or gateway.
- Changing the life cycle, life cycle state, team, or folder within Windchill.
- Executing a scheduled job.

A publish job is always created and passed to a Windchill processing queue. The publish job is *self-logging*; that is, all end-user messages are contained within the job itself, and they can be viewed through the WVS publish monitor.

When the publisher receives a publish job, it traverses structures within Windchill and extracts necessary CAD files. The traversal and file selection is based on the type of data being processed. The publisher uses the CAD agent to schedule the conversion of the CAD data.

The CAD agent has CAD workers configured to it. A *CAD worker* is a program written using the API of a particular CAD system, and it produces files that ProductView can read from native CAD files and assemblies. The CAD agent manages the CAD worker resource.

The following are characteristics of the CAD agent:

- Aware of which CAD workers are configured.
- Can start and stop CAD workers as necessary.
- Allow viewing of CAD worker log files.
- Allow CAD worker to be tested during configuration.
- Manages the passing of data to and from the CAD worker.

The CAD worker typically runs on a remote system. When the CAD agent receives the published data, it returns it to the publisher. The publisher then stores the resulting data in Windchill, by programmatically invoking the loader.

# Troubleshooting

This section provides information you can use to analyze and resolve issues that may arise with the WVS components. For CAD Agent troubleshooting and additional publishing troubleshooting information, see the *Windchill System Administrator's Guide*.

## Windchill Visualization Service Loader

The method server starts the WVS loader service. The relevant entry in the wt.properties file is in the following form:

```
wt.services.service.nn=com.ptc.wvs.server.loader.GraphicsServer  
LoaderService/com.ptc.wvs.server.loader.StandardGraphicsServerL  
oaderService
```

The service first reads the wvs.properties file to ensure that the wvs.enabled property is set to true. If this property is not set to true, the service is not started.

The loader may output additional debugging information to the method server start window and log if the edrload.verbose property is set to true.

Every 5 seconds, the loader polls the directory defined by the following property:

```
edrload.directory=$(wt.temp)\\wcinput
```

If this directory does not already exist, the loader creates it. When polling the directory, the loader looks only for INI files. All other files are ignored. If the contents of an INI file (located in the directory) are terminated with <!>, it is renamed with a .txt extension. (For example, ticket.ini would be renamed ticket.txt.)

If content is not terminated with <!>, the loader waits 5 more seconds to make sure the file is not currently being written to. If, after 5 seconds, the file content still does not terminate with <!>, the file is deleted. The loader requires write access to the file so that it can rename or delete it.

In order to start processing them, the loader next parses the file and validates the contents.

The file should contain entries of the form Keyword=value (for example, Partnumber=123456). The following table lists valid keywords:

| Keyword         | Value or Description                                                                      |
|-----------------|-------------------------------------------------------------------------------------------|
| Directory       | Specifies the fully qualified directory location of the converted data.                   |
| Documentnumber  | Specifies the number of an existing WTDocument to which you associate the representation. |
| Documentversion | Specifies the version of the WTDocument to which you associate the representation.        |

| <b>Keyword</b> | <b>Value or Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Encoding       | Specifies the character set encoding of the PVS file (if the PVS file has no J tag present). The default is to use the J tag-specified encoding or the encoding of the Windchill server.                                                                                                                                                                                                                                                                         |
| Edzcreate      | Specifies if an PVZ file is to be created. Can be set to true or false.<br><br>The default is false.                                                                                                                                                                                                                                                                                                                                                             |
| Ignoreonmerge  | Adds a flag to children of the PVS files root node to indicate that those children should be ignored when using this representation in a WTPart structure. For example, if the root WTPart includes a representation of the complete assembly, but you want to view the data from the individual WTParts when viewing a structure, use ignoreonmerge.                                                                                                            |
| Includemarkups | Specifies if markups in the input data should be stored with the representation in Windchill. Can be set to true or false.<br><br>The default is true.                                                                                                                                                                                                                                                                                                           |
| Iteratepart    | Specifies if an existing part is to be iterated. Can be set to true or false.<br><br>The default is false.                                                                                                                                                                                                                                                                                                                                                       |
| Partfolder     | Specifies the folder in which the part is created.                                                                                                                                                                                                                                                                                                                                                                                                               |
| Partlifecycle  | Specifies the life cycle associated with the new part.                                                                                                                                                                                                                                                                                                                                                                                                           |
| Partcontainer  | Specifies the context (for example, Project, Product, or Library) that in which a new part is created. The folder and life cycle values are determined by the context and need not be specified.<br><br>You can specify the context as a name or parentname/name. For example, if an organization PTC contains a project proj1, you can specify partcontainer as proj1 or PTC/proj1 (to distinguish it from other projects called proj1 in other organizations). |
| Partname       | Specifies the part name.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Partnumber     | Specifies the part number of an existing part (the part number is created if it does not exist).                                                                                                                                                                                                                                                                                                                                                                 |

| Keyword          | Value or Description                                                                                                                                                                      |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Partoid          | Specifies the Windchill ID of an existing part.                                                                                                                                           |
| Partteam         | Specifies the team associated with the new part.                                                                                                                                          |
| Partrevision     | Specifies the part revision.                                                                                                                                                              |
| Repdefault       | Specifies whether the representation is to be the default. Can be set to true or false.<br><br>The default is false.                                                                      |
| Repdesc          | Describes the representation to be created.                                                                                                                                               |
| Repname          | Specifies the name of the representation to be created.                                                                                                                                   |
| Representableoid | Specifies the Windchill ID of an existing representable.                                                                                                                                  |
| Thumbnailcreate  | Specifies whether a thumbnail is to be created. Can be set to true or false.<br><br>The default is false.                                                                                 |
| Ticketencoding   | Specifies the character set encoding of the ticket file. If specified, this must be the first line in the ticket file. If not specified, the encoding of the Windchill server is assumed. |

**Note:** The file must end in <!.>. Keywords are case-insensitive, as are true and false values.

The initial checks of the file ensure that the directory specified by the Directory keyword exists and that the loader can write to it.

Additional checks include the following actions, some of which depend on keyword values:

- If a Partoid is specified in the file, it is checked to ensure that it references a valid WTPart.
- If a Partoid is not specified, the Partfolder, Partlifecycle, and Partteam values are checked to ensure that they exist.
- If the Partnumber/Partname does not exist, a WTPart is created. If it does exist, and Iteratepart is set to true, the part is iterated. The result is a WTPart in the database to which a new representation will be added, with the specified Repname and Repdescription.
- The specified directory is scanned to locate the PVS file. Only one PVS file is allowed. All other files are uploaded to Windchill, associated as secondary content of the representation.

- If a Representableoid is specified in the file, it is checked to ensure that it references a valid representable.
- If the Thumbnailcreate keyword is set to true, a thumbnail image is created and uploaded as content of the representation, provided the thumbnail generator has been installed. If it is the default representation, the thumbnail is copied (shared) to the Representable.
- If the Edzcreate keyword is set to true, an PVZ file containing all the files in the directory is saved as content of role PRODUCT\_VIEW\_EDZ on the representation, provided edrload.edzenabled=true is also set in wvs.properties.
- The PVS file is modified to reference the secondary content in Windchill, rather than the local files, before being stored as the role PRODUCT\_VIEW\_ED on the Representation.
- The loader removes the ticket.txt file from the directory it is polling. This occurs whether the loading task succeeds or fails. If an error occurs, it is reported only in the method server log. Data referenced by the ticket is not removed. The removal of the ticket.txt file signifies that the loader has completed its task.

**Note:** For a large assembly, the loader task can be time-consuming, especially if thumbnail generation is performed. For more information about options that are available for thumbnail generation, see the *Windchill Installation and Configuration Guide - Advanced*.

## Publishing CAD Documents

This section provides troubleshooting information for publishing operations.

### Timeouts

When the CAD agent sends a request to the CAD worker, it has no way of determining the status of the job. Therefore, the CAD agent waits for a specified period of time. In the wvs.properties file, the following properties define timeout values for publishing:

```
publish.cadtimeout.component=600
publish.cadtimeout.assembly=3600
publish.cadtimeout.drawing=600
```

These properties specify the number of seconds that the CAD agent waits when the publisher is processing a single component, assembly, or drawing, respectively. These values should be adjusted to the needs of your site, so that they will process the largest data sets. If the values are too small, errors are displayed, and no viewable CAD data is created.

Alternatively, many of the CAD workers can be configured with long and short timeout values that are sent back to the CADAgent. If these have been configured, the last timeout value sent to the CADAgent is used. See the CAD worker documentation for details of setting CAD worker timeouts in its recipe file.

You should also tune the CAD agent settings for Auto Idle Stop and Auto Busy Stop to help control system resources. (These values are specified when you use the CAD Agent Wizard to configure a CAD worker.) For example, for CADDSS5, when processing of drawings is enabled, Auto Idle Stop should be set to about 900 seconds. For Pro/ENGINEER, setting Auto Busy Stop ensures that system memory is released on a regular basis.


When you set values that automate the stopping of CAD workers, you should enable Auto Start and correctly configure it so that the worker can be restarted.

## Automated Publishing

There are several ways in which the publishing of viewable CAD data can be automated. An event is emitted when a CAD document is checked in from a Workgroup Manager, the Pro/INTRALINK Gateway, or the Optegra Gateway. If the following property is set to true in the wvs.properties file, this event results in submission of a publish job:

```
publish.service.readytopublish.enabled=true
```

This job is processed exactly as if the user (who checked the file in) had made a

publish request using the publish action icon  on the property page. The config spec that is used for the publishing process, is provided by the application that emits the event, for example, a WGM. If that config spec is "null," WVS uses either the latest config spec or an as-stored config spec depending on the value of the property "publish.configspec.default.useasstoredifavailable."

## Publish Scheduler

With the Publish Scheduler, requests can be submitted to the Windchill Schedule queue for processing. Examples are provided below, but typically you will set automated publishing schedules specific to your site, to automate the publishing of certain types of data on a regular basis.

You can access the Publish Scheduler from the **Utilities** page on the **Site**, **Organization**, **Project**, and **Product** tabs.

The first time that a scheduled job is submitted, a new Windchill schedule queue called WVSScheduleQueue is created. You can use the Windchill Queue Manager to confirm that this queue exists and is active. The schedule queue executes a publish request at the date and time specified, with the specified frequency.



Settings in the `wvs.properties` file define the jobs that are available for selection from the **Schedule Publish Job** page. There are a number of example jobs already configured. An appropriate schedule job can be created if you have specific requirements.

There are two parts to the process of creating a Schedule Publish Jobs:

- configuring the `wvs.properties` file
- writing the java code to select the objects to be published

The following property defines the list of available entries (where `<n>` is an increasing integer, starting with 1).

```
Schedulejobs<n>=<schedulename>
```

The *schedulename* is then used to find additional properties of the following forms:

```
<schedulename>.description=pull-down description  
<schedulename>.class=<ClassContainingMethod>  
<schedulename>.method=<nameOfMethod>  
<schedulename>.enableOnContainers=<true/false>
```

The `<schedulename>.description` property defines the text that appears in the drop-down menu on the **Schedule Publish Job** page. The class and method are used to identify the specific method that will be invoked by the schedule job. The value of `enableOnContainers` determines if this schedule job is displayed in the list of jobs when the scheduler UI is invoked in a specific context. This indicates that the schedule job contains code to filter the objects to be published based on the context.

The following are the signatures of the publish job method:

```
public static QuerySpec <nameOfMethod>()  
  
or  
  
public static QueryResult <nameOfMethod>()
```

If a `QuerySpec` is returned, then `PersistenceHelper.manager.find()` is used to make the query and return a `QueryResult`. This `QueryResult` or the one returned directly from the `schedule jobs` method contains the EPMDocuments/WTParts/WTDocuments/Representations that are sent for publishing.

A default configuration spec is used. If the `QueryResult` contains a `Representation`, then that is sent for republishing. If the `QueryResult` contains a `WTDocument`, all publishable files (that is, those with worker XXX=mapping defined in `wvs.properties`) are sent for publishing.

To obtain the current container context for the schedule job, use the following method call in the jobs method:

```
WTContainerRef cr = com.ptc.wvs.server.schedule.ScheduleJobs.getCurrentContainer();
```

The following example schedule job method will publish all EPMDocuments in the current context:

```
public static QuerySpec allEPMDocuments()
{
    QuerySpec qs = null;
    try {
        qs = new QuerySpec(EPMDocument.class);
        WTContainerRef cr =
com.ptc.wvs.server.schedule.ScheduleJobs.getCurrentContainer();
        if( cr != null ) {
            ContainerSpec cs = new ContainerSpec();
            cs.addSearchContainer(cr);
            qs.setAdvancedQueryEnabled(true);
            qs.appendWhere(WTContainerHelper.getWhereContainerIn(cs,
EPMDocument.class),new int[]{0});
        }
    } catch (Exception e ) { e.printStackTrace(); }
    return qs;
}
```

To obtain debug information for a schedule job, set the following property to true in the wvs.properties file:

```
publish.publishqueuehelper.verbose=true
```

## Visualization Collaboration

Install the collaboration agent from the Visualization - Windchill Support CD, which adds the required entry into wvs.properties (via site.xconf) to specify the command to execute the collaboration agent. If required (for example, to change the assigned port numbers), you can change the value of the "collaboration.server" property.

The options for the collaboration server define the minimum (-p) and maximum (-m) port numbers that are used by that server. Each collaboration session has its own running collaboration server, so this range should be large enough to accommodate the expected number of concurrent collaboration sessions. The default is 40. If the ProductView clients are operating through a firewall to the Windchill server, these ports must be open. The idle timeout can also be specified, in seconds, by the -i option. The default is to shut down a collaboration server if no one is connected for 10 minutes.

Only the user who starts a collaboration session can stop it or can host data into the session. All other users can only join the session. If the user who starts a session logs out and logs back in, he or she must go to the collaboration listing and make the session hosted into his or her current Windchill session before being able to load data into it.

## Creating and Enabling Server-side Watermarks

When you want users to be able to access common watermarks (such as a company logo) across the enterprise, you can create server-side watermarks for your Windchill solution.

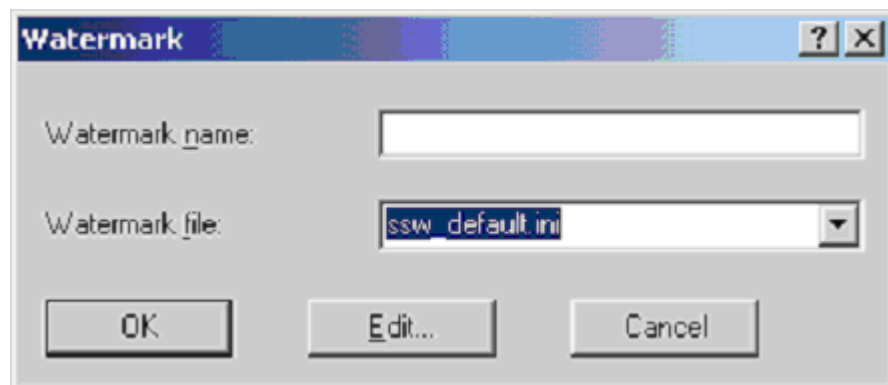
Watermarks can be viewed along with objects in the ProductView Standard Edition.

The following sections provide the details on how to create and enable server-side watermarks.

### Creating Server-side Watermarks

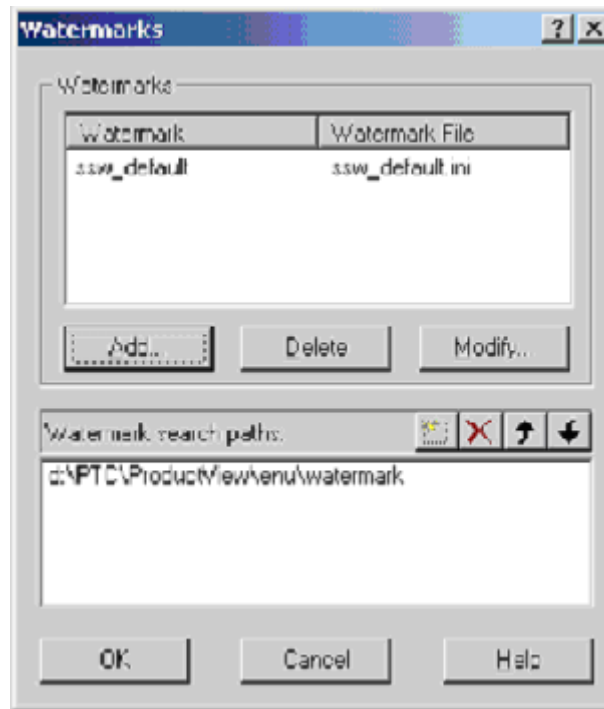
To set up a server-side watermark, follow these steps:

1. Back up and remove all default files from the `<ProductView>/<lang>/watermark` folder.
2. Open the ProductView Watermark Editor to create the watermark(s) you need for the various types of objects, such as 3D models, 2D drawings, documents, and images.
3. Save the watermark in the `<ProductView_installation>/<lang>/<watermark_folder>`. You can name the watermark folder `ssw_default`, which is used as an example throughout this procedure. The watermark file will be saved as a .ini file (`ssw_default.ini`).

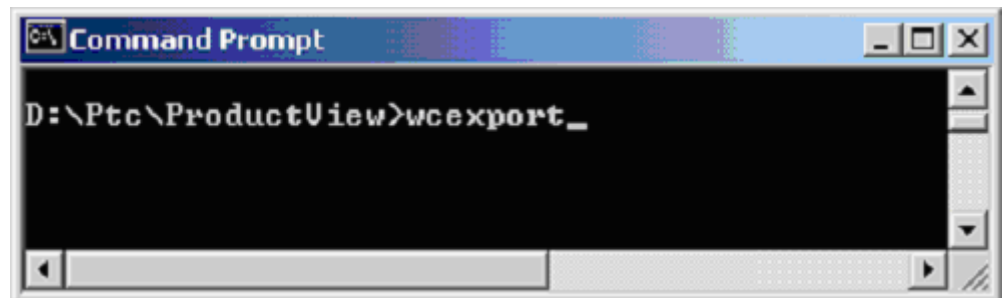


4. Open ProductView and choose **Edit > Preferences**.
5. In the **Preferences** dialog box, select **Watermarks** to display **Watermarks Preferences**.

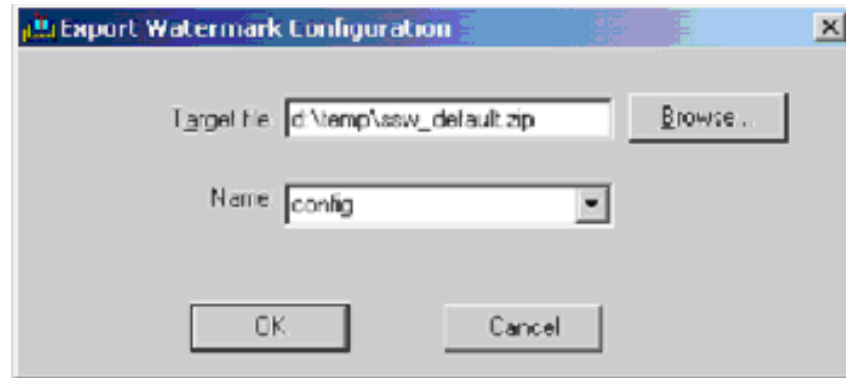
6. Click **Watermarks** to display the **Watermarks** dialog box shown below.



7. In the **Watermarks** dialog box, click **Add** and select the watermark (in this case, `ssw_default.ini`) from the **Watermark File** drop-down list.
8. Click **OK**. The **Watermarks** dialog box now displays the watermark you added.
9. Click **OK** again to enable the watermark when viewing and printing. View the watermark in ProductView Standard Edition to ensure that it looks correct.
10. Close Product View and, using the command line window, type `<Pview_install>/wcexport`, where `<Pview_install>` is the ProductView installation directory.



11. In the **Export Watermark Configuration** dialog box, in the **Target File** field, enter the path and name of the target zip file you want to create. For example, type d:\temp\ssw\_default.zip.



12. Use the **Name** drop-down list in the **Export Watermark Configuration** dialog box to specify areas such as Collaboration, Collection, Document, EPM document, or a part for the server-side watermark. Select **config** in the **Name** field to enable server-side watermarks for all objects and areas.
  - When publishing a shared object in Windchill ProjectLink, and viewing in the context of a share, WVS uses the context of the project instead of the product. As a result, when viewing the representation created in the share, the watermark displays the context of project.
  - In Windchill ProjectLink, when you publish an EMPDocument in a project with the latest configuration specification, the representation that is created will display detail information such as "EPM: Project: WVS Project1", when listed in the Representations table.
13. Click **OK** to create the server-side watermark zip file (in this case, ssw\_default.zip). This zip file will contain bitmap (BMP) files for images that you used in the watermark, along with two .ini files: config.ini and <watermark\_filename>.ini. In this example, ssw\_default is the watermark file name.
14. To disable users from editing watermarks, you can append the config.ini file within ssw\_default.zip with [Locked] as follows:

```
[Locked]

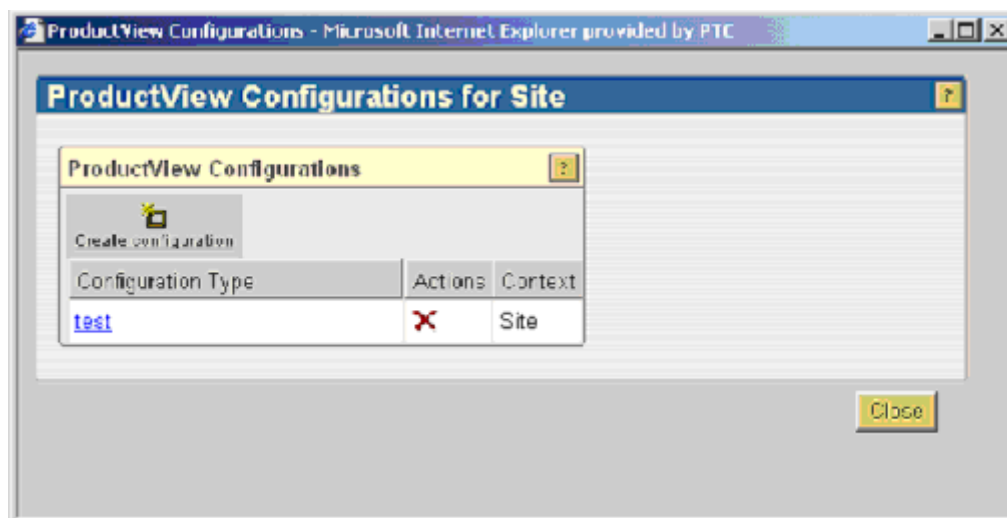
"WatermarksLocked"=dword:01
```

## Enabling Server-side Watermarks on the Server

To enable server-side watermarks on a Windchill server, follow these steps:

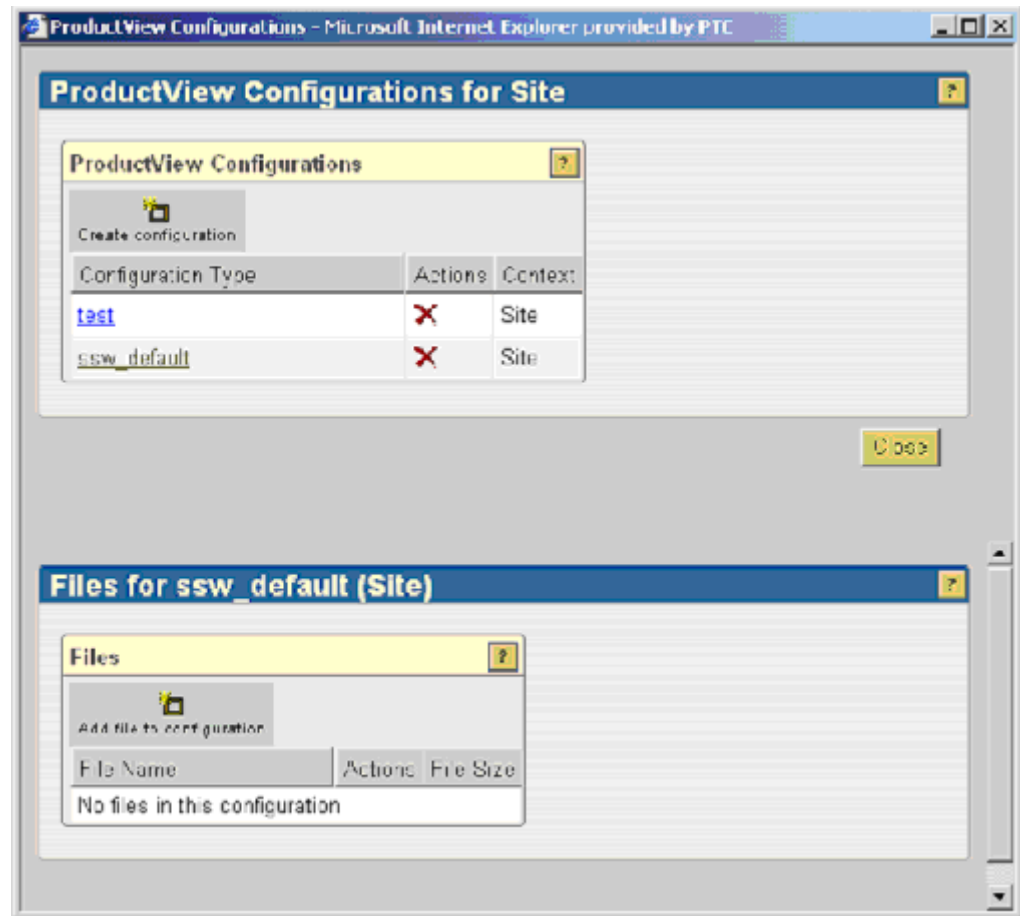
1. Log in as an administrator (for example, wadmin/wcadmin).
2. Using the xconfmanager utility, set the value "productview.configfromserver=true" in the wvs.properties file and propagate this change to wvs.properties.
3. As with any change to wvs.properties, you must restart the method server and servlet engine.
4. To create a watermark configuration, go to the **Server Controlled Configuration of ProductView**, accessed from the admin link in the ProductView Portal (or equivalent on bundled server configurations).
5. Select **Site > Utilities > Visualization Configuration Administrator** and click the Create configuration icon to add the watermark zip file. Create a configuration, for example, ssw\_default.

**Note:** This process adds the configuration to the current container context, such as a product, project, organization, or site. For example, a configuration added at the product level will be used in preference to a configuration added at the organizational level, which in turn will be used in preference to a configuration added at the site level.



When you click the Create configuration icon, you are prompted to enter an optional configuration type. For information about the available configuration types and their use, see the productview.modifyconfig property.

If you do not enter a configuration type and click **Create**, a default type is created. Click the **Default Type** link to display the **Files** table.



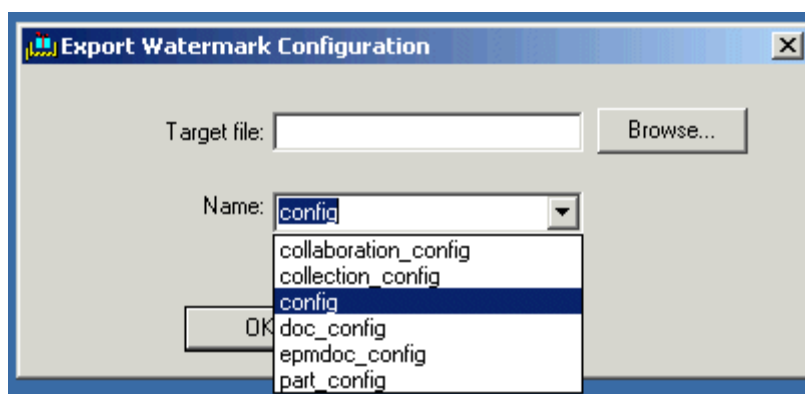
6. Click the Add file to configuration icon. Browse to select the watermark zip file (ssw\_default.zip).

You can view the server-side watermarks in both the ProductView Standard and in ProductView Lite Editions. Refer to the ProductView Help for details on viewing watermarks.

## Exporting Watermarks for ProductView

ProductView supports watermarking of 3D, drawings, images, and documents. Watermarks are defined in INI files created and edited using the ProductView watermark editor. The administrator that manages watermarks manually transfers the INI files from the watermarks directory into the Windchill server.

To export the watermark configuration, go to the location that ProductView Standard Edition is installed. Run the wcexport executable file, found in the productview installation directory. The **Export Watermark Configuration** window appears:



Specify the file name you want to export and click **Browse** to locate the directory. The wcexport tool creates a ZIP file including the main config.ini file, any watermark INI files referenced by the registry definitions, and any images referenced by any watermark file.

To add the ZIP file to the ProductView configuration for Windchill, click **Administration** on the Visualization navigation bar; then click **Server Controlled Configuration of ProductView**. To add the ZIP file for Windchill PDMLink or Windchill ProjectLink, click **Visualization Configuration Administration** on the **Utilities** pages of the appropriate tabs.

**Note:** For Windchill PDMLink or Windchill ProjectLink, the context of the **Utilities** page defines where the watermark configuration is stored. For example, if you are accessing Visualization Configuration from the Organization tab, you are defining the watermark configuration for the organization.

For more information on ProductView/Visualization configuration, see the Windchill online help for the **ProductView/Visualization Configurations** table.



# Copy Forward Functionality

You can copy forward Representations and Markups.

## Copying Representations Forward

This functionality controls the copying forward of Representations to new iterations of Representables (for example, Windchill Parts, CAD Documents, and Windchill Document objects). The fundamental principal is that Copy Forward applies only to out-of-the-box, non-derived Representations. However, in the case of a Copy or Revise of a Representable, all Representations (derived and non-derived) will be copied forward out-of-the-box. This behavior also applies to Representables being checked out to a new context. (You can adjust this out-of-the-box behavior using the Copy Forward properties described later in this section.)

For example, where a Representation has been created from pre-converted viewables on the local file system and associated to a Windchill Part that has no describing CAD Document, when that Windchill Part iterates, all of the Representations will copy across to the new iteration, as they cannot be derived from anything.

For a Representation that is associated directly to a CAD Document (because it has no described Windchill Part) or to a Windchill Part with an active link to a describing CAD Document, where the Representation was derived from the CAD Document content: when the CAD Document iterates, the Representation should not be copied forward. This is because it should have a new Representation created from the updated content and metadata of the new CAD Document iteration. This behavior is referred to as "Restricted Mode" Copy Forward, and is enforced at the current releases, out-of-the-box.

## Copying Markups Forward

Markups (ProductView Annotations, Markups, and Groups) can optionally be configured to copy forward when their parent Representation is copied forward or a new Representation of the same name is created on a new iteration of the Representable. The decision to copy Markups forward is based on the allowCopyForward flag, which is set on each individual Markup object and manually changed using the Annotations and Markups list. By default, the value of this flag is false, but if it is set to true, Markups associated directly to a Windchill Document (no Representation for natively viewed content) will be copied forward to the new iteration.

## Copy Forward Properties

The Copy Forward properties in wvs.properties are described next, grouped into the following categories:

- Enabling the Copy Forward functionality
- Configuring the Representation Copy Forward functionality
- Configuring the Markup Copy Forward functionality

### Enabling the Copy Forward Functionality

Use the following properties to enable the Copy Forward functionality:

| Property                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>publish.copyrepresentationsforward=true</code><br><code>publish.copymarkupsforward=true</code>                     | Allow Representations to be copied forward when parts iterate, as long as it does not compromise the validity of data published from EPM data. When Representations are copied forward, Markups and Annotations can be copied forward as well. If the copied forward Representation has Markups, it will not be replaced during publishing.                                                       |
| <code>publish.documents.copymarkupsforward=true</code><br><code>publish.documents.copyrepresentationsforward=true</code> | Set copy forward of Representations and Markups specifically for WTDocuments. All Representations of WTDocuments are candidates for copy forward, even if they are published from the document content files, as publishing only occurs when document files are upload. If markups on the WTDocuments Representations are copied forward, the Representations will not be replaced by publishing. |

## Configuring the Representation Copy Forward Functionality

Use the following properties to configure the Copy Forward functionality:

| Property                                                                                                                         | Description                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>publish.copyrepresentationsforward.restrict=true</code>                                                                    | Restricted Representation copy forward mode will not copy a published Representation forward when a WTPart iterates, and the old and new iterations both point to the same EPMDocument.                                                      |
| <code>publish.documents.copyrepresentationsforward.restrict=true</code>                                                          | Restricted Representation copy forward mode will not copy a published Representation forward when a WTDocument iterates, and the old and new iterations both point to the same WTDocument.                                                   |
| <code>publish.copyforwardallrepresentationsoncontainerchange=true</code>                                                         | When a new Representable iteration is created in a different context (such as during a sandbox checkout), this setting determines whether all Representations should be copied forward, or only those that would normally be copied forward. |
| <code>publish.copyforwardallrepresentationsoncopy=true</code><br><code>publish.copyforwardallrepresentationsonrevise=true</code> | When a Representable is copied/revise, determines whether to copy forward all the Representations, or only those that would normally be copied forward.                                                                                      |

## Configuring the Markup Copy Forward Functionality

Use the following properties to configure the Markup Copy Forward functionality:

| Property                                                                                                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>publish.copymarkupsrestricttosametype=false</code>                                                                                                                   | Determines whether the Markup copy should be restricted to copying to same types (from a Representation to a Representation, or a Viewable to a Viewable) only.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>edrload.copymarkupsfrompreviousiteration=</code><br><code>edrload.copymarkupsfrompreviousversion=</code><br><code>annotation,markup,group,pair_group,sequence</code> | Determines whether to copy markups from a previous iteration when a new Representation is created. If the previous iteration is in the same version, the <code>copymarkupsfrompreviousiteration</code> list is used, to decide the list of markup types to copy, for example, <code>annotation</code> , <code>Markup</code> , <code>group</code> , <code>pair_group</code> , or <code>sequence</code> .<br><br>If the previous iteration is in a different version, the <code>edrload.copymarkupsfrompreviousversion</code> list is used to determine the list of markup types to copy. |
| <code>edrload.copymarkupsmatchdefaultfirst=false</code>                                                                                                                    | When copying Markups from a previous iteration, the Representation to copy from is first matched based on the name and then on the default status. Setting this property to true will match on default status in preference to name.                                                                                                                                                                                                                                                                                                                                                    |
| <code>markup.allowcopyforward.default=false</code>                                                                                                                         | When a new Markup is created, determines whether its <code>allowCopyForward</code> flag should be true or false. If true, the Markup can be copied forward, depending on other settings.                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>publish.viewable.copymarkupsforward=true</code>                                                                                                                      | Allows Markups associated directly to a Viewable (for example, <code>WTPart</code> or <code>WTDocument</code> ) to be copied forward when the Viewable iterates.                                                                                                                                                                                                                                                                                                                                                                                                                        |

# Using Publish Rules

This section provides an overview of publish rules, along with instructions on enabling, evaluating, and creating publish rules along with post-publishing.

## About Publish Rules

Publish Rules provide two basic features:

- The ability to create a first-class business object to expose the generated output from the publish job, which is typically only accessible on the Representation generated by a publish job.

For example, if a PDF file is published and added to a Representation, a new WTDocument can be created to contain the PDF file as its primary content, thereby increasing the enterprise manageability of that PDF file. In addition, the new WTDocument is associated with the Representable that holds the Representation, which allows historical tracking of the WTDocument's primary content. This association is called a *PublishedContentLink*. The Publish Rules allow you to tailor the post-publish behavior to your needs.

- Increased control over what is published. You can specify rules for different scenarios, such as whether publishing is triggered by check-in, a schedule, or a manual creation from Windchill. You can also specify custom behaviors based on the type or number of the object that triggered the publish. Some authoring applications allow multiple output options, and the rules can help define when these particular output types are generated.

The terms *Dynamic Documents* and *EPMDocuments* are both used in this section. Dynamic Documents are specific to Arbortext and are a soft type of EPMDocument. For information on creating soft type and soft attributes for use with Arbortext Editor and the WVS publisher, see [Managing Types and Attributes for Dynamic Documents](#).

**Note:** Publish Rules are specified using an XML file. It is assumed that the reader of this document has a basic understanding of XML.

## Enabling Publish Rules

The site administrator must establish that Publish Rules Processing is to be used. This is done by including the names of all Authoring Applications that will use Publish Rules Processing in the WVS Property `publish.usesPublishRules`; for example, use the `xconfmanager` utility to set the following:

```
publish.usesPublishRules=ARBORTEXT PROE
```

This property setting indicates that EPMDocuments authored by Arbortext or Pro/ENGINEER make use of Publish Rules Processing; all other Authoring Applications are not affected by Publish Rules in this example.

The process for enabling publish rules is outlined below:

1. An administrator must load a Publish Rules XML file into a WVS Configuration Template. The Administrator can create WVS Configuration Templates in Product/Project/Library, Organization, or Site contexts.
2. When determining whether to publish an EPMDocument, WVS checks whether the authoring application for the EPMDocument is listed in WVS Property `publish.usesPublishRules`.
3. If WVS finds the authoring application in the `publish.usesPublishRule` property entry, the EPMDocument's Product/Project/Library, Organization, and Site contexts are searched hierarchically for a WVS Configuration Template containing a Publish Rules XML file.
  - The first Publish Rules file found during that search will be used during Publish Rules Evaluation.
  - If a Publish Rules file is not found during this search, processing will continue as if the Authoring Application was not configured for Publish Rules Processing.

## Evaluating Publish Rules

The following sections describe the events and logic for evaluating Publish Rules.

### Events that Trigger Publish Rules Evaluation

Publish Rules Evaluation is triggered by the following events:

- Checkin of the EPMDocument.
- Scheduled publishing of the EPMDocument.
- User selection of the “Create Representation” action on the EPMDocument's Representations List (or its associated WTPart's Representations List).

### Evaluating Publish Rules Logic

This section describes the logic of Publish Rules evaluation and includes fragments of Publish Rules files. Unless otherwise stated, the evaluation process uses case-sensitive string comparisons.

The Publish Rules file is a well-formed XML document, with `<rules>` being its root element.

Keep in mind that this section attempts to provide a high-level overview of the evaluation process. Many of the details you will need to successfully construct your own Publish Rule files are in the sections that follow. See also [Examples of Publish Rules XML Files](#).

**Note:** Parameters and values in the publish rules file are case sensitive.

### Step 1: Matching <authoring-application>

The <rules> element should have one <authoring-application> child for each Authoring Application specified in the WVS Property publish.usesPublishRules.

For example, to specify Arbortext and Pro/ENGINEER as authoring applications, these commands would be used:

```
<rules>
  <authoring-application name="ARBORTEXT">
    *
    *
    *
  </authoring-application>

  <authoring-application name="PROE">
    *
    *
    *
  </authoring-application>
</rules>
```

In general, the concept is:

```
<rules>
  <authoring-application name="MY_AUTH_APP">
    *
    *
    *
  </authoring-application>
</rules>
```

Publish Rules Evaluation searches for the <authoring-application> element with a 'name' attribute matching the EPMDocument's Authoring Application. If a match is not found, no Publish Jobs are generated; otherwise, evaluation proceeds to Step 2.

## Step 2: Matching <epm-number>

Evaluation continues by searching the children of the <authoring-application> element matched in Step 1 for an <epm-number> element whose 'number' attribute matches the EPMDocument's Number.

**Note:** <authoring-application> elements can have children other than <epm-number> elements; those elements are ignored in this step even if they appear in the Publish Rules XML file before <epm-number> elements.

```
<authoring-application name="MY_AUTH_APP">
  <epm-number number="1111">
    *
    *
    *
  </epm-number>

  <epm-number number="2222">
    *
    *
    *
  </epm-number>
</authoring-application>
```

If a match is found, the matched <epm-number> element becomes the root for searching for <publish> elements, which is described in Step 7; otherwise, evaluation proceeds to Step 3.

## Step 3: Matching <epm-iba> with 'value' attribute present

Evaluation continues by searching the children of the <authoring-authoring> element matched in Step 1 for an <epm-iba> element corresponding to an IBA Name/Value pair within the EPMDocument.

**Note:** The 'value' attribute of <epm-iba> is an optional attribute; Step 3 only considers those <epm-iba> elements whose 'value' attribute is present. We will discuss what happens when no 'value' attribute is present in Step 4.

```
<authoring-application name="MY_AUTH_APP">
  <epm-iba iba="IBA_NAME_1" value="IBA_VALUE_1">
    *
    *
    *
  </epm-iba>

  <epm-iba iba="IBA_NAME_2" value="IBA_VALUE_2">
    *
    *
    *
  </epm-iba>
</authoring-application>
```

If a match is found, the matched <epm-iba> element becomes the root for searching for <publish> elements, which are described in Step 7; otherwise, evaluation proceeds to Step 4.



#### Step 4: Matching <epm-iba> without 'value' attribute present

Evaluation continues by searching the children of the <authoring-application> element matched in Step 1 for a <epm-iba> element whose 'iba' attribute matches to an IBA Name within the EPMDocument.

**Note:** The 'value' attribute of <epm-iba> is an optional attribute; Step 4 only considers those <epm-iba> elements whose 'value' attribute is not present. The <epm-iba> elements that meet this criterion are considered in the order they appear in the Publish Rules XML file.

```
<authoring-application name="MY_AUTH_APP">
  <epm-iba iba="IBA_NAME_1">
    *
    *
    *
  </epm-iba>

  <epm-iba iba="IBA_NAME_2">
    *
    *
    *
  </epm-iba>
</authoring-application>
```

If a match is found, the matched <epm-iba> element becomes the root for searching for <publish> elements, which are described in Step 7; otherwise, evaluation proceeds to Step 5.

#### Step 5: Matching <epm-type>

Evaluation continues by searching the children of the <authoring-application> element matched in Step 1 for an <epm-type> element with a 'type' attribute matching the EPMDocument's Object Type. The value of the 'type' attribute is the EPMDocument's *Logical Identifier*. Logical Identifiers are defined in the Windchill Type and Attribute Manager.

**Note:** For more information about the Windchill Type and Attribute Manager, see the [Overview of the Type and Attribute Manager Utility](#) section in the Type and Attribute Manager chapter.

```
<authoring-application name="MY_AUTH_APP">
  <epm-type type="MyLogicalId">
    *
    *
    *
  </epm-type>

  <epm-type type="AnotherLogicalId">
    *
    *
    *
  </epm-type>
</authoring-application>
```

If a match is found, the matched <epm-type> element becomes the root for searching for <publish> elements, which are described in Step 7; otherwise, evaluation proceeds to Step 6.

### Step 6: Fall-through matching

If evaluation has not found a search root in any of the previous steps, the <authoring-application> element becomes the root for searching for <publish> elements. Publish elements are described in Step 7.

An example of fall-through publishing is provided next:

```
<authoring-application name="MY_AUTH_APP">
  <epm-number number="1111">
    *
    *
    *
  </epm-number>

  <epm-type type="MyLogicalId">
    *
    *
    *
  </epm-type>

  <!-- Begin: Fall-through search root -->
  *
  *
  *
  <!-- End: Fall-through search root -->
</authoring-application>
```

In this example, if the EPMDocument's number is not 1111 or does not have the logical identifier "MyLogicalId", the search root becomes the <authoring-application> (as noted between the Begin and End comment lines above).

**Note:** Using the fall-through matching is not a recommended practice, as you lose control of the ability to bypass publishing on any EPMDocuments for a particular authoring application. Fall-through matching is useful when testing Publish Rules, allowing you to focus only on matching the <authoring-application>.

## Step 7: Matching <publish>

Steps 2 through 6 determine a root for searching for <publish> elements. This root can be an <epm-number>, <epm-iba>, <epm-type>, or <authoring-application> element.

A Publish Job will be created for each <publish> element child of the root whose 'on' attribute matches the event that caused Publish Rules Evaluation to be invoked. If no <publish> elements are matched, no Publish Jobs will be created.

```
< ... root for publishing ... >

<publish on="checkin" param-set="SET1"/>

<publish on="checkin"/>

<publish on="schedule" output="VALID_WORKER_OUTPUT"
param-set="SET1"/>

</ ... end of root for publishing ... >
```

For each match, the attributes of the matched <publish> element are processed. The possible attributes are as follows:

**on** - The possible 'on' values correlate to the triggers described at the very beginning of this section. They are "checkin", "schedule", and "create-representation". (A fourth trigger called "manual-post" is described later in this document.) In the example above, there are three "on" values: two for checkin and one for schedule. If the trigger for Publish Rules evaluation was due to a checkin of an EPMDocument, two publish jobs would be created. If the trigger was due to a scheduled job of an EPMDocument, one publish job would be created. If the trigger was due to the manual create representation wizard from the user interface, no publish jobs would be created.

**output** - If an 'output' attribute is present (as shown in the third <publish> line above), it will be part of the information in the Publish Job that can be used by the Worker. The use of this attribute is not valid for all Workers.

**param-set** - If a 'param-set' attribute is present, the Publish Rules XML file is searched for a <param-set> element with a 'name' attribute matching the value of the 'param-set' attribute. Step 8 describes the processing of a matched <param-set> element. In the example above, the param-set value is "SET1".

## Step 8: Processing <param-set>

<param-set> elements can appear anywhere between the <rules> tags within the Publish Rules XML file. It is often useful to reference the same <param-set> element from several <publish> elements. The children of a <param-set> element contain information that will be part of the Publish Job. The information is contained in the following elements:

**post-publish** – These elements are used to pass necessary information to the PostPublishDelegate. For more information, see [Post-Publishing](#).

**iba** – These elements are also used to pass optional information to the PostPublishDelegate. For more information, see [Post-Publishing](#).

In order for IBAs to be set during processing, you first must define the IBA for the type in the Type and Attribute Manager (see the [Type and Attribute Manager](#) chapter). Since the Publish Rules reside in an XML file that contains strings, you need to be aware of how you represent non-string values to pass to the post publishing delegate.

The following is a list of supported IBA types and an example value for each IBA:

```
<iba name="com.ptc.MyTypeBoolean">Yes</iba>
<iba name="com.ptc.MyTypeDateTime">2007-07-13 11:12:32</iba>
<iba name="com.ptc.MyTypeInteger">36</iba>
<iba name="com.ptc.MyTypeRealNumber">0.12321345</iba>
<iba name="com.ptc.MyTypeRealUnitsArea">25 m**2</iba>
<iba name="com.ptc.MyTypeString">string test</iba>
<iba name="com.ptc.MyTypeURL">http://www.ptc.com (PTC)</iba>
```

**worker** – This element is used to pass extra information to the Worker if the Worker supports it. The worker is determined by the authoring application of the EPMDocument that is being used during the evaluation.

These elements have an identical structure; each has a ‘name’ attribute to identify the parameter and the text content of the element is the parameter’s value.

```
<param-set name="SET1">
  <post-publish name="name1">VALUE1</post-publish>
  <post-publish name="name2">VALUE2</post-publish>
  <iba name="iba_name1">IBA_VALUE1</iba >
  <iba name="iba_name2">IBA_VALUE2</iba >
  <worker name="worker_info_name">WORKER_INFO_VALUE</ worker >
</param-set>
```

A <param-set> element can have any number of <worker>, <post-publish>, and <iba> elements; however, unlike most of the other evaluation steps, the order of appearance is important. When Step 7 matches a <publish> element, the evaluation process constructs an internal table for each element, for example, a table for <worker>, a table for <post-publish>, and a table for <iba>. The children of the <param-set> element are retrieved in the order they appear in the file.

As each child is processed, the tables are populated with name/value pairs. Parameter names must be unique within each table, but are not required to be unique across tables. (For example, you can have the same <post-publish> name and <iba> name, but you can't have two <post-publish> names be the same.)

When a child is processed, if its parameter identifier was previously encountered, its table entry will be updated with a new parameter value, thus replacing previously encountered values for the same name.

A <param-set> element can also have <include> element children. The <include> element adds the parameters associated with a specified <param-set> to the tables. This is akin to calling a subroutine.

```
<param-set name="SET1">
  <include param-set="COMMON"/>
  <post-publish name="name">From SET1</post-publish>
</param-set>

<param-set name="COMMON">
  <post-publish name="name">From COMMON</post-publish>
</param-set>
```

In the above fragment, if the SET1 <param-set> was referenced in the Publish Rules file, the <post-publish> parameter 'name' will have the value 'From SET1' because the <include> element in SET1 appears before the <post-publish> element. If the elements were reversed, the value would be 'From COMMON'.

The <include> element effectively allows <param-set> elements to be chained together. After the entire chain is processed, processing proceeds to Step 9.

## Step 9: Substitution of text values

After the <param-set> elements have been processed and the parameter tables of Step 8 have been populated, the parameter values in the tables are scanned for Substitution Keys. A *Substitution Key* is a predefined sequence of characters (always beginning with an opening brace and ending with a closing brace) which, when encountered, is replaced with information shown in the table below.

Substitution Key	Data Substituted
{AUTHORING_APP}	Source EPMDocument's Authoring Application
{EPM_NAME}	Source EPMDocument's Name
{EPM_NUMBER}	Source EPMDocument's Number
{EPM_TYPE}	Source EPMDocument's Type
{OUTPUT_TYPE}	Value of 'output' attribute of <publish> element
{PARAM_SET_NAME}	Value of 'param-set' attribute of <publish> element

For example, if the following element were encountered in a <param-set>:

```
<post-publish name="name">{EPM_NUMBER} authored by  
{AUTHORING_APP}</post-publish>
```

The Name of the target Object would be something like the following. (In this example, the number of the EPMDocument is 000047.)

```
000047 authored by MY_AUTH_APP
```

Once all substitutions have been made, the parameter tables are associated with the Publish Job and evaluation returns to Step 7 to search for additional <publish> elements.

## Step 10: Evaluation Complete

After the Publish Rules are evaluated, you will have zero to many publish jobs for a single EPMDocument. Additionally, you may have defined post-publishing and worker-specific parameters for some or all of the publish jobs that were created.

A successful publish job results in a Representation. Then, based on the Publish Rules, you can define post-publishing behavior.

## Post-Publishing

Post-publishing is the process of executing delegated code after a representation has been successfully created and stored. The code that executes is called a *PostPublishDelegate*.

There are currently two *PostPublishDelegate* classes that exist: *DefaultPostPublishDelegate* and *EPMPostPublishDelegate*.

### DefaultPostPublishDelegate

The *DefaultPostPublishDelegate* is specific to Arbortext authored DynamicDocuments (EPMDocuments).

Once the publishing of a *DynamicDocument* is complete and a representation is stored, the *DefaultPostPublishDelegate* copies the content of the representation to a *WTDocument*. This is done to give enterprise control and visibility of the published content separately from the *DynamicDocument* structure, while maintaining a record of the association through a *PublishedContentLink*. Additionally, if the *WTDocument* already exists, the delegate code will iterate the *WTDocument*, thus providing a means to track the history through the iteration history of the *WTDocument*.

The table below lists the valid <post-publish> parameter names for the *DefaultPostPublishDelegate* and describes how each parameter is used.

### Required Parameters

The following parameters are required for the *DefaultPostPublishDelegate* class.

Parameter Name	Definition
delegate	Must be com.ptc.wvs.server.publish.DefaultPostPublishDelegate.
name	The Name of the Object that the PostPublishDelegate is to create/iterate. The Name must be unique among all Objects of its Type in order to guarantee that the delegate finds the right Object to iterate.

## Optional Parameters

The following parameters are optional for the DefaultPostPublishDelegate class.

Parameter Name	Definition
number	The number of the object that the PostPublishDelegate is to create/iterate. If specified, the Number must be unique among all Windchill Objects of this Type. If not specified, the Number is automatically generated.
type	The type of object that the PostPublishDelegate is to create. If specified, the Type can be specified by its Logical Identifier and must be a soft-type of wt.doc.WTDocument. If not specified, the DefaultPostPublishDelegate creates a wt.doc.WTDocument.
title	Title of the WTDocument.
description	Description of the WTDocument.
container-oid	The Object Identifier String in which the WTDocument is to be contained. If this value is not specified, the WTDocument is placed in the same context as the EPMDocument being published.
folder	The name of the folder (relative to the context) for the WTDocument. For example: /Default/My/Folder
content-description	A description for the content of the WTDocument.
lifecycle-template	The life cycle template to associate with the WTDocument.
lifecycle-state	The life cycle state to associate with the WTDocument.
published-content-link	The create <b>Published Content Link</b> indicator. If true, yes, or create, a <b>Published Content Link</b> is created. If not specified or any other value, a <b>Published Content Link</b> is not created.



## EPMPublishDelegate

The EPMPublishDelegate will generically handle any CAD authoring application.

Once publishing of an EPMDocument is complete and a representation is stored, the EPMPublishDelegate creates a ProductView archive file (PVZ) of the representation's content and stores the archive to a new EPMDocument. This provides the ability to insert the new EPMDocument into an Arbortext DynamicDocument structure and view the lightweight CAD data in the Arbortext Editor.

Additionally, any markups and annotations can be used in the Editor and subsequently published to provide tightly controlled, lightweight figures/illustrations in technical publications that are tied to real modeled CAD data. Similar to the DefaultPostPublishDelegate, a PublishedContentLink is maintained between the CAD data and the lightweight data objects for use with Arbortext Editor.

The table below lists the valid <post-publish> parameter names for the EPMPublishDelegate and describes how each parameter is used.

### Required Parameters

The following parameters are required for the EPMPublishDelegate class.

Parameter Name	Definition
delegate	Must be com.ptc.wvs.server.publish.EPMPostPublishDelegate.
name	The Name of the Object that the PostPublishDelegate is to create/iterate. The Name must be unique among all Objects of its Type in order to guarantee that the delegate finds the right Object to iterate.
cad-authoring-application	The Authoring Application to associate with the EPMDocument to be created.

## Optional Parameters

The following parameters are optional for the EPMPublishDelegate class.

Parameter Name	Definition
number	The number of the object that the PostPublishDelegate is to create/iterate. If specified, the Number must be unique among all Windchill Objects of this Type. If not specified, the Number is automatically generated.
type	The type of object that the PostPublishDelegate is to create. If specified, the Type can be specified by its Logical Identifier and must be a soft-type of wt.doc.WTDocument. If not specified, the EPMPublishDelegate creates a wt.epm.EPMDocument.
description	Description of the EPMDocument.
container-oid	The Object Identifier String in which the EPMDocument is to be contained. If this value is not specified, the EPMDocument is placed in the same context as the source EPMDocument.
folder	The name of the folder (relative to the context) for the EPMDocument. Example: /Default/MyFolder
content-description	A description for the content of the EPMDocument.
lifecycle-template	The life cycle template to associate with the EPMDocument.
lifecycle-state	The life cycle state to associate with the EPMDocument.
cad-type	If not specified, the EPMPublishDelegate uses the value PUB_CADVIEWABLE.  <b>Note:</b> This parameter is for advanced users only.

Parameter Name	Definition
published-content-link	The create Published Content Link indicator. If true, yes, or create, a Published Content Link is created. If not specified or any other value, a Published Content Link is not created.

## Manual Post Publishing

It is also possible to execute only the post-publishing portion of the Publish Rules. This extra flexibility allows you to separately control publishing and the execution of a PostPublishDelegate.

In [Step 7: Matching <publish>](#), it was mentioned that there is a fourth value for the ‘on’ attribute called “manual-post”. Creating a <publish> element that uses “manual-post” will specify what EPMDocuments can have manual post-publishing of their representations invoked. For example:

```
< ... root for publishing ... >
    <publish on="checkin"/>
    <publish on="manual-post" param-set="MY_PARAM_SET"/>
</ ... end of root for publishing ... >
```

When the Representations Listing is displayed for an EPMDocument (or associated WTPart), the Publish Rules are evaluated for a match on the EPMDocument and a manual-post publish element. If the evaluation yields Publish Job(s), the Post Publish action is enabled in the Representation Listing. If the User selects the action, the PostPublishDelegate for each Publish Job is executed.

**Note:** A Worker is not involved in this interaction.

## Examples of Publish Rules XML Files

This section contains complete Publish Rules XML File examples.

### Simple Arbortext Publishing

This example publishes all Arbortext authored EPMDocuments when they are checked in, when a user creates a representation via the Create Representation Wizard, or when the EPMDocument is included in scheduled publishing. A PDF file will be created by the Worker and stored as part of the Representation. The DefaultPostPublishDelegate is invoked when the Worker completes the creation or iteration of a WTDocument whose content will be the PDF created by the Worker.

## Highlights

- The use of Substitution Keys to define the Name of the WTDocument that will be created/iterated.
- Several <publish> elements make use of the same <param-set>.

```
<rules xmlns="http://www.ptc.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.ptc.com PublishRulesSchema.xsd">
  <authoring-application name="ARBORTEXT">
    <param-set name="Share with WTDocument">
      <post-publish
name="delegate">com.ptc.wvs.server.publish.DefaultPostPublishDelegate</post-publish
>
      <post-publish name="name">{OUTPUT_TYPE} {AUTHORING_APP}
{EPM_NUMBER}</post-publish>
      <post-publish name="published-content-link">create</post-publish>
    </param-set>
    <publish on="checkin" output="PDF" param-set="Share with WTDocument"/>
    <publish on="create-representation" output="PDF" param-set="Share with
WTDocument"/>
    <publish on="schedule" output="PDF" param-set="Share with WTDocument"/>
  </authoring-application>
</rules>
```

## Creating Arbortext Documents from CAD Documents

This examples publishes PRO/ENGINEER and SolidWorks authored EPMDocuments.If the EPMDocument has an IBA named PUBLISH TO ARBORTEXT or if a user manually invokes Post Publishing on an existing Representation, the EPMPublishDelegate will be invoked to create/iterate an Arbortext authored Dynamic Document.

## Highlights

- Multiple Authoring Applications are making use of a single <param-set>.
- All PRO/ENGINEER and SolidWorks authored EPMDocuments will be published, but the EPMPublishDelegate will only be invoked for those that have the specified IBA (its value is not considered).
- The EPMPublishDelegate can be invoked by the user for any Representation previously published by these Authoring Applications.

```
<rules xmlns="http://www.ptc.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.ptc.com PublishRulesSchema.xsd">
  <param-set name="Share with Arbortext">
    <post-publish
name="delegate">com.ptc.wvs.server.publish.EPMPublishDelegate</post-publish>
    <post-publish name="type">DynamicDocument</post-publish>
    <post-publish name="cad-authoring-application">ARBORTEXT</post-publish>
    <post-publish name="cad-type">PUB_CADVIEWABLE</post-publish>
    <post-publish name="name">GDD from {AUTHORING_APP} {EPM_NUMBER}</post-publish>
    <post-publish name="published-content-link">create</post-publish>
  </param-set>

  <authoring-application name="PROE">
    <epm-iba iba="PUBLISH TO ARBORTEXT">
      <publish on="checkin" param-set="Share with Arbortext"/>
      <publish on="create-representation" param-set="Share with Arbortext"/>
      <publish on="schedule" param-set="Share with Arbortext"/>
    </epm-iba>

    <publish on="checkin"/>
    <publish on="create-representation"/>
    <publish on="schedule"/>
    <publish on="manual-post" param-set="Share with Arbortext"/>
  </authoring-application>

  <authoring-application name="SOLIDWORKS">
```

```

<epm-iba iba="PUBLISH TO ARBORTEXT">
    <publish on="checkin" param-set="Share with Arbortext"/>
    <publish on="create-representation" param-set="Share with Arbortext"/>
    <publish on="schedule" param-set="Share with Arbortext"/>
</epm-iba>

<publish on="checkin"/>
<publish on="create-representation"/>
<publish on="schedule"/>
<publish on="manual-post" param-set="Share with Arbortext"/>
</authoring-application>
</rules>

```

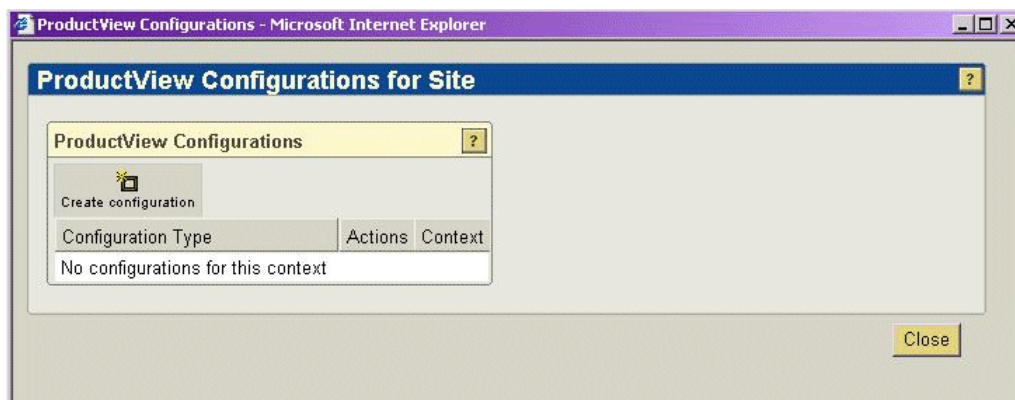
## Creating Publish Rules

Once you have read and understood the previous sections, you are ready to create your own Publish Rules. To do this, you can use the sample file in the *<Windchill>/codebase/com/ptc/wvs/server/xml* directory. Copy this file to another location before editing.

## Loading the Publish Rules XML File

Follow these steps to load the Publish Rules XML file. You must be logged on as a site administrator to access the windows shown in this example.

1. Navigate to the **Site > Utilities** page.
2. Select **Visualization Configuration Administrator**. This displays the **ProductView Configurations for Site** dialog box.



This example assumes there are no current configurations defined.

3. Click **Create Configuration** to create a new configuration at the Site level. This displays the **Create New Configuration for Site** dialog box.

The screenshot shows a web browser window titled "ProductView Configurations - Microsoft Internet Explorer". The main content area has a blue header bar with the text "Create New Configuration for Site" and a question mark icon. Below the header, there is a text box with the following text: "A type can optionally be specified, which can be used to provide a different configuration when a user has modify access to the item being viewed. See property 'productview.modifyconfigtype'." Below this text is a label "Configuration type:" followed by an empty text input field. At the bottom right of the dialog, there are two buttons: "Create" and "Cancel".

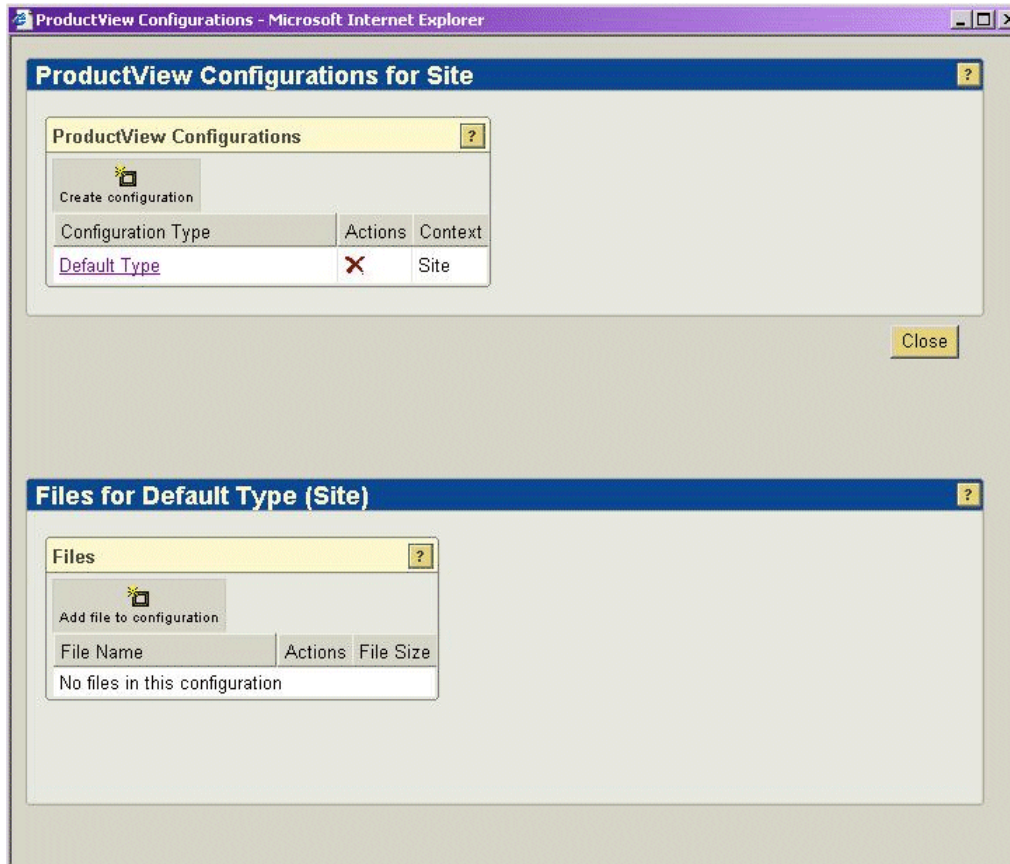
4. Type the name for the configuration you are creating, and click **Create**. In this example, the configuration will be called **Default Type**. The **ProductView Configurations for Site** dialog box displays again with the new configuration listed in the table.

The screenshot shows the same web browser window, but the dialog box now displays a table of configurations. The dialog box has a blue header bar with the text "ProductView Configurations for Site" and a question mark icon. Below the header, there is a section titled "ProductView Configurations" with a question mark icon. Under this section, there is a "Create configuration" button. Below the button is a table with the following data:

Configuration Type	Actions	Context
<a href="#">Default Type</a>		Site

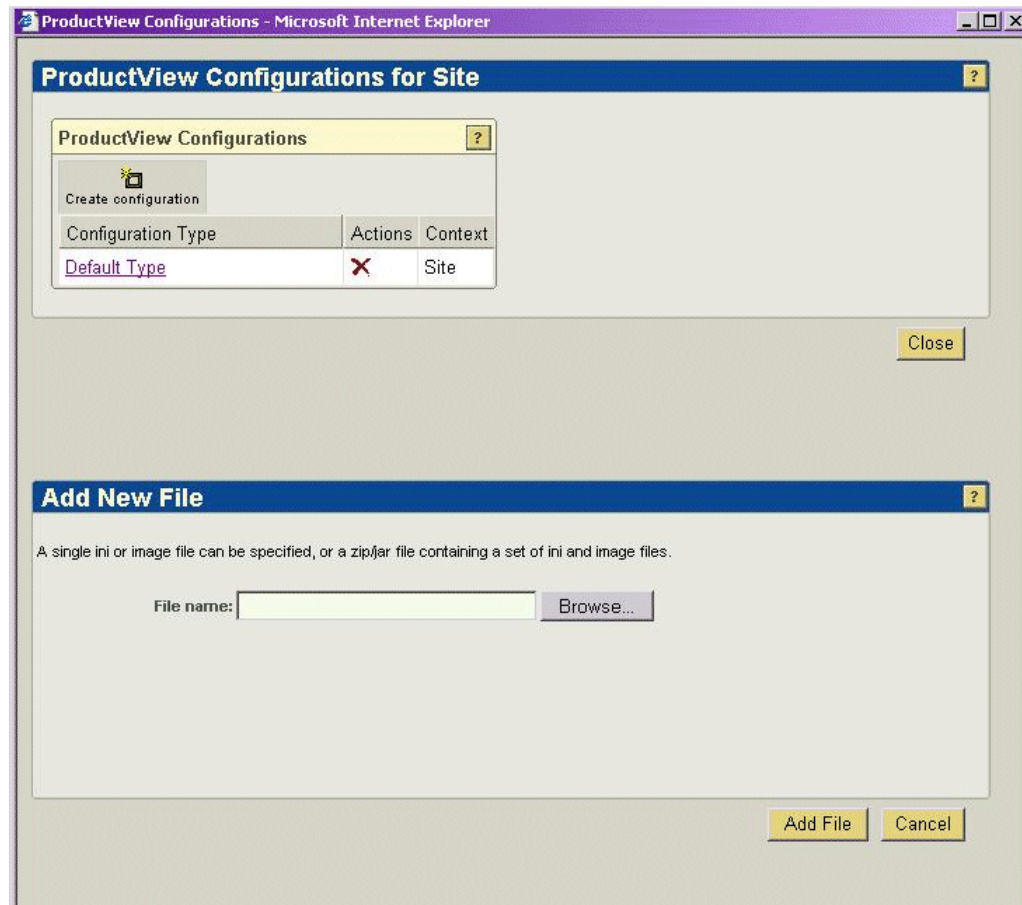
At the bottom right of the dialog, there is a "Close" button.

5. Under **Configuration Type**, select **Default Type**. This displays the **Files for Default Type** list in the **ProductView Configurations** dialog box. When you first create a configuration, the file list will be empty. The next step is to add a file to the configuration.

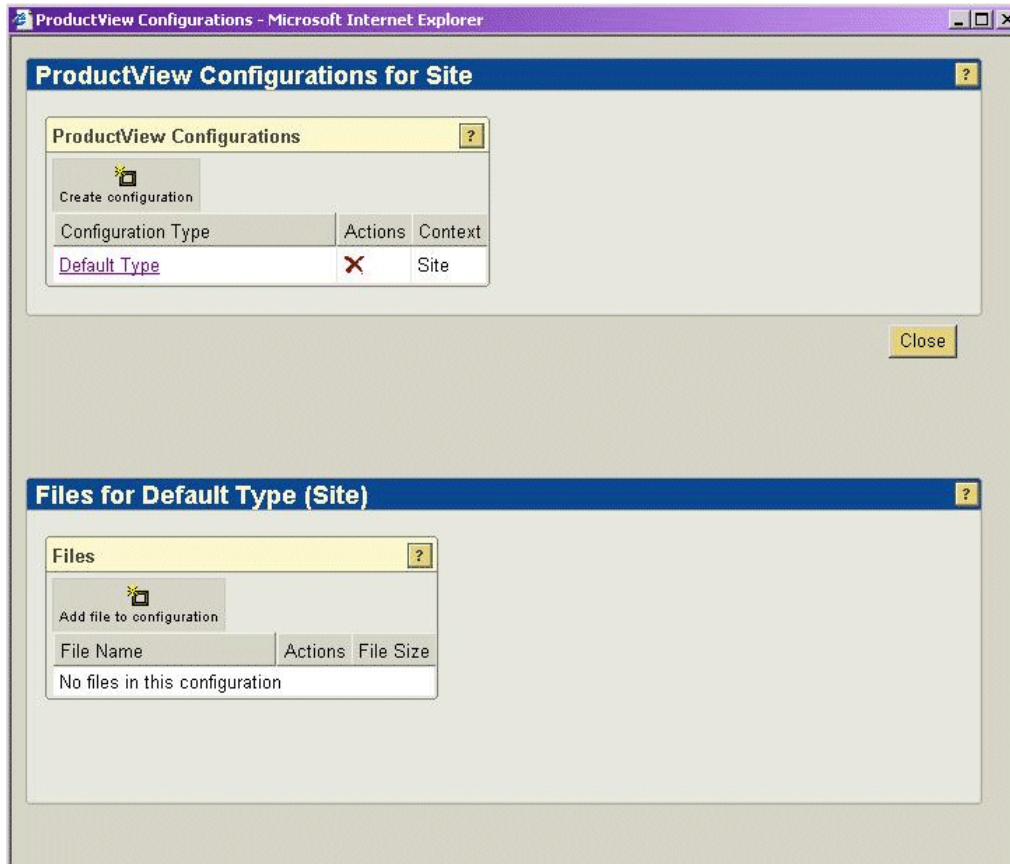




6. In the **Files for Default Type** box, click **Add File to Configuration**. The **Add New File** dialog box displays.



- Click **Browse** and select the publish rules XML file you want to load. Click **Add File** to load the XML file to the configuration.



- If you wish to change the Publish Rules, repeat the steps in the wizard above to load the updated Publish Rules file. When you see the old file listed in the **Files for Default Type** list shown above, delete it. Then, click **Add file to configuration** and select the updated file.

**Note:** Changes will be realized immediately after you load the new file.

For more advanced configuration, you can override the site template by adding a template at the organization level. Or, you can add a template to the library, project, program, or product within the organization to apply just to those specific areas.

## Windchill Visualization Service Properties

Windchill uses standard Java property files to dynamically configure many optional or site-dependent settings. 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.

You can edit these files by using the `xconfmanager` utility, which allows you to add properties and values, delete properties, and save your changes to the properties files, for implementation when you restart the Windchill system. For more information, see [About the xconfmanager Utility](#) in the Administration Overview chapter.

Windchill Visualization Service (WVS) uses the properties described in the following table. They are set in the `wvs.properties` file.

Windchill Visualization Service Property	Description
<code>cadagent.inifile</code>	<b>Default Value:</b> <code>\$(wt.home)\\codebase\\agent.ini</code> <b>Synopsis:</b> Configuration file for CADAgent. <b>Description:</b> Specifies the configuration file used by the CADAgent. This file configures the CAD Workers that are available for use by the WVS Publisher.
<code>cadagent.logs</code>	<b>Default Value:</b> <code>\$(wt.logs.dir)\\cadagent</code> <b>Synopsis:</b> Directory for CADAgent log files. <b>Description:</b> Specifies the directory where CADAgent log files are written.
<code>cadagent.pvfiletypes</code>	<b>Default Value:</b> OL ED PVS PVP PVM PLT DXF HPGL PGL TXT AST CCZ CC GIF JPG PDF PVT GRP EMK ETB PVA CGM TGA DWF EXTENTS DWG <b>Synopsis:</b> File types that the CadAgent will retrieve. <b>Description:</b> Add to this space-delimited list, any file extensions that the CAD workers may create which need to be stored in Windchill. This list is not case sensitive.
<code>cadagent.filetypes.XXX</code>	<b>Default Value:</b> NONE <b>Synopsis:</b> File types that the CadAgent will retrieve for a specific worker type XXX. <b>Description:</b> Add to this space-delimited list, any file extensions that the specific worker type XXX may create which need to be stored in Windchill. This list is not case sensitive. For example, the types for a THUMBNAIL worker are defined as JPG PVT EXTENTS GIF.

Windchill Visualization Service Property	Description
cadagent.monitor.workerverboseargs	<p><b>Default Value:</b> @debug_options.txt</p> <p><b>Synopsis:</b> Arguments passed via the worker monitor to the worker when started in debug mode.</p> <p><b>Description:</b> Arguments passed via the worker monitor to the worker when started in debug mode from the CADAgent admin page.</p>
cadagent.testfiledirectory	<p><b>Default Value:</b> \$(wt.home)\loadFiles\wvs\testfiles.</p> <p><b>Synopsis:</b> The directory where the test files are stored.</p> <p><b>Description:</b> The directory where the test files are stored when loaded from the Visualization - Windchill Support CD. The test files are used by the CADAgent admin test worker function.</p>
collaboration.enabled	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables the Collaboration option from the WVS portal.</p> <p><b>Description:</b> Enables the Collaboration option from the WVS portal. This allows ProductView collaboration to be used in a Windchill context. The property is set when the collaboration agent is installed from the Visualization - Windchill Support CD.</p>
collaboration.server	<p><b>Default Value:</b> \$(wt.home)\bin\pview_collaboration.exe -p 5620 -m 5660 -i 600</p> <p><b>Synopsis:</b> Collaboration server start command.</p> <p><b>Description:</b> Specifies the command for starting the collaboration server. Typical options are; -p start_port_number, -m max_port_number, -i idle_timeout_in_seconds. The collaboration server executable referred to here should be placed in the correct location. The property is set when the collaboration agent is installed from the Visualization - Windchill Support CD. For more information on collaboration see the ProductView documentation.</p>
collaboration.tempdir	<p><b>Default Value:</b> \$(wt.temp)\collaboration</p> <p><b>Synopsis:</b> Directory used for Temporary files for collaboration.</p> <p><b>Description:</b> Specifies the directory used for Temporary files for collaboration.</p>
collaboration.verbose	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Enable verbose output for services related to ProductView collaboration</p> <p><b>Description:</b> Enable verbose output for services related to ProductView collaboration.</p>

Windchill Visualization Service Property	Description
distributedcadagent.enabled	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Set to true to enable the configuration of a Distributed CadAgent.</p> <p><b>Description:</b> Specifies if the CAD Agent Monitor wizard will present options and screens allowing a Distributed CadAgent to be configured. This only may be relevant when working with data populated via the Pro/INTRALINK gateway.</p>
edrload.copymarkupsfrompreviousiteration	<p><b>Default Value:</b> annotation,markup,group,pair_group,sequence</p> <p><b>Synopsis:</b> Copy markups from previous iteration when a new representation is created.</p> <p><b>Description:</b> If previous iteration is in the same version, specifies to copy markups from previous iteration when a new representation is created.</p>
edrload.copymarkupsfrompreviousversion	<p><b>Default Value:</b> annotation,markup,group,pair_group,sequence</p> <p><b>Synopsis:</b> Copy markups from previous iteration when a new representation is created.</p> <p><b>Description:</b> If previous iteration is in a different version, specifies to copy markups from previous iteration when a new representation is created.</p>
edrload.copymarkupsmatchdefaultfirst	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Setting this property to true matches the default status in preference to name.</p> <p><b>Description:</b> When copying markups from previous iteration, the representation to copy from is first matched on the name and then on the default status. Setting this property to true matches the default status in preference to name.</p>
edrload.copyreferencedanddescribingtorep	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Copy content from referenced and describing documents to the representation.</p> <p><b>Description:</b> Specifies whether to copy content from referenced and describing documents to the representation or reference the content from the document.</p>
edrload.copytransform	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Copy transform information from EPM./CAD System to Part structure.</p> <p><b>Description:</b> Copies transform information from EPM./CAD System to Part structure. For CAD systems whose structure is file driven, this will cause publish.matchcadnames to be treated as true.</p>

Windchill Visualization Service Property	Description
edrload.directory	<p><b>Default Value:</b> \$(wt.temp)\\wcinput</p> <p><b>Synopsis:</b> Directory for ticket files.</p> <p><b>Description:</b> Specifies the directory polled by loader for ticket files that are used to load preconverted visualization data into Windchill.</p>
edrload.docpropertygroup	<p><b>Default Value:</b> WindchillDocument</p> <p><b>Synopsis:</b> ProductView property group for WTDocument properties.</p> <p><b>Description:</b> Specifies the ProductView property group for WTDocument properties. A change to this value will require the corresponding change in the ProductView installation.</p>
edrload.edzenabled	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Enables creation of a PVZ file when visualization data is stored.</p> <p><b>Description:</b> Enables the creation of a PVZ file when visualization data is stored. This includes data stored by the WVS publisher. Based on a user preference, a user will view a PVZ (if available) or a PVS file. A PVZ file will download to the client all file data from a Representation initially, while with an PVS file, individual files are downloaded on demand.</p>
edrload.epmpropertygroup	<p><b>Default Value:</b> WindchillEPM</p> <p><b>Synopsis:</b> ProductView property group for EPMDocument properties.</p> <p><b>Description:</b> Specifies the ProductView property group for EPMDocument properties. A change to this value will require the corresponding change in the ProductView installation.</p>
edrload.includedescribing	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Includes Describing WTDocuments in PVS file.</p> <p><b>Description:</b> Includes Describing WTDocuments in PVS file for WTPart structure traversal. Significantly increases the structure traversal time.</p>
edrload.includeepmpropertiesinpartstructure	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Flag to specify if EPMDocument properties/property page link is to be included in part structure viewing/representations.</p> <p><b>Description:</b> Flag to specify if EPMDocument properties/property page link is to be included in part structure viewing/representations.</p>

Windchill Visualization Service Property	Description
edrload.includepartmastersdefault	<b>Default Value:</b> True <b>Synopsis:</b> Specifies whether to include part masters in the part structure. <b>Description:</b> Flag to specify if part masters are to be included in the part structure.
edrload.includeproperties	<b>Default Value:</b> True <b>Synopsis:</b> Includes Windchill properties in PVS file. <b>Description:</b> Includes Windchill properties in PVS file.
edrload.includepropertypagelink	<b>Default Value:</b> True <b>Synopsis:</b> Link to properties pages is to be added. <b>Description:</b> Flag specifies if link to properties pages is to be added.
edrload.includereferenced	<b>Default Value:</b> False <b>Synopsis:</b> Includes Referenced WTDocuments in PVS file. <b>Description:</b> Includes Referenced WTDocuments in PVS file for WTPart structure traversal. Will significantly increase the structure traversal time.
edrload.new.encoding	<b>Default Value:</b> <b>Synopsis:</b> Character set to use for new PVS files. <b>Description:</b> Specifies the character set to use for new PVS files (for example, a representation of a part structure). If not specified (default), the system default encoding is used.
edrload.overwritetransform	<b>Default Value:</b> True <b>Synopsis:</b> Overwrites transform information on Part structure. <b>Description:</b> Overwrites transform information on Part structure, when copying data from EPM/CAD System.
edrload.partpropertygroup	<b>Default Value:</b> WindchillPart <b>Synopsis:</b> ProductView property group for WTPart properties. <b>Description:</b> Specifies the ProductView property group for WTPart properties. A change to this value will require the corresponding change in the ProductView installation.
edrload.propertyexplicitlist	<b>Default Value:</b> <b>Synopsis:</b> Windchill properties that should be retrieved explicitly and added to the ProductView representation. <b>Description:</b> Windchill properties that should be retrieved explicitly and added to the ProductView representation.

Windchill Visualization Service Property	Description
edrload.propertyskiplist	<b>Default Value:</b> <b>Synopsis:</b> Windchill properties that should not be included in ProductView representations. <b>Description:</b> Windchill properties that should not be included in ProductView representations.
edrload.read.encoding	<b>Default Value:</b> \$(wvs.edfileencoding) <b>Synopsis:</b> Default character set to use for reading PVS or ETB files. <b>Description:</b> Specifies the default character set to use for reading PVS or ETB files. The default is to use the default charset of the server. You can also specify the charset in ticket file to override this value. If the PVS file specifies a value, that value is always used.
edrload.verbose	<b>Default Value:</b> False <b>Synopsis:</b> Enables debug mode for loader. <b>Description:</b> Enables debug mode for loader.
edrload.write.encoding	<b>Default Value:</b> <b>Synopsis:</b> Character set to use for storing PVS files in database. <b>Description:</b> Specifies the character set to use for storing PVS files in database. If not specified (default), the existing encoding or the system default is used.
markup.allowcopyforward.default	<b>Default Value:</b> False <b>Synopsis:</b> When a new markup is created, should it be set to copy forward or not. <b>Description:</b> When a new markup is created, should it be set to copy forward or not. The copy forward setting for individual markups can be set from the Annotations and Groups listing.
markup.tempdir	<b>Default Value:</b> \$(wt.temp) <b>Synopsis:</b> Temporary directory used when creating markups. <b>Description:</b> Specifies the temporary directory used when creating markups.
markup.verbose	<b>Default Value:</b> False <b>Synopsis:</b> Enables debug information when creating markups. <b>Description:</b> Enables debug information when creating markups.
productview.batchprintoptions	<b>Default Value:</b> batchprint='true' <b>Synopsis:</b> Options to send to ProductView at startup when batch printing. <b>Description:</b> Options to send to ProductView at startup when batch printing.



Windchill Visualization Service Property	Description
productview.collectionconfigbasedonfirstobject	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> When viewing objects from the visualization collection, the ProductView configuration selected is based on the context of the first object.</p> <p><b>Description:</b> When viewing objects from the visualization collection, the ProductView configuration selected is based on the context of the first object.</p>
productview.collectionconfigbasedonorganization	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> When viewing objects from the visualization collection, the ProductView configuration selected is based on the organization of the user.</p> <p><b>Description:</b> When viewing objects from the visualization collection, the ProductView configuration selected is based on the organization of the user.</p>
productview.configfromserver	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Get ProductView configuration information from server.</p> <p><b>Description:</b> Get ProductView configuration information from server (for example, this retrieves watermark information from the WVSTemplate for the context of the object being viewed).</p>
productview.modifyconfigtype	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> The type for the WVSTemplate to be used when a user has modify access to the object being viewed.</p> <p><b>Description:</b> When ProductView receives configuration files from the server, users who have modify access to the representable can use a different WVSTemplate object. The type is specified when the WVSTemplate is created and that string should match the value of this property. If no value is specified, all users, irrespective of access rights, use the same configuration, based on the context.</p>
productview.options	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Options to send to ProductView at startup.</p> <p><b>Description:</b> Options to send to ProductView at startup.</p>
productview.redirectoptions	<p><b>Default Value:</b> redirecturl='unload.jsp'</p> <p><b>Synopsis:</b> Web page that ProductView will redirect to upon exit.</p> <p><b>Description:</b> Specifies the Web page that ProductView will redirect to upon exit. The default page simply closes the small web browser window that hosts the ProductView plugin.</p>

Windchill Visualization Service Property	Description
publish.alwaysusespecifiedobject	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Publish specified object event if the specified config spec does not select it.</p> <p><b>Description:</b> When the object specified to be published would not be selected with the specified config spec, the specified object or the one selected by the config spec can be used for the publish job. For example, publishing an old iteration of an EPMDocument when using a Latest config spec.</p>
publish.cadconvert."xxx"	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Application Type to Java class lookup.</p> <p><b>Description:</b> Specifies a required entry that relates the Application Type to a Java class that will handle the CAD system specific publishing for any CAD system supported by the WVS Publisher.</p>
publish.cadtimeout.assembly	<p><b>Default Value:</b> 3600</p> <p><b>Synopsis:</b> Timeout for conversion of CAD assemblies.</p> <p><b>Description:</b> Specifies the timeout for CADAgent conversion of CAD assemblies in seconds.</p>
publish.cadtimeout.component	<p><b>Default Value:</b> 600</p> <p><b>Synopsis:</b> Timeout for conversion of CAD components.</p> <p><b>Description:</b> Specifies a timeout for CADAgent conversion of CAD components in seconds.</p>
publish.cadtimeout.drawings	<p><b>Default Value:</b> 600</p> <p><b>Synopsis:</b> Timeout for conversion of CAD drawings.</p> <p><b>Description:</b> Specifies the timeout for CADAgent conversion of CAD drawings in seconds.</p>
publish.configspec.default.useasstoredifavailable	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Use as stored config spec when publishing EPMDocuments</p> <p><b>Description:</b> When publishing an EPMDocument structure and no config spec has been specified, or the Create Representation wizard specifies default (which is the default option), the config spec used depends on the EPMDocument. Setting this option to true causes the as stored config spec to be use when default is specified.</p>
publish.copyforwardallrepresentationsoncontainerchange	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> All representations can be copied forward when a new representable iteration is created in a different context (for example, in an integral checkout).</p> <p><b>Description:</b> Indicates if all representations should be copied forward when a new representable iteration is created in a different context (for example, in an integral checkout).</p>

Windchill Visualization Service Property	Description
publish.copyforwardallrepresentationsoncopy	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> When a representable is copied, copy forward all the representations or only those that would normally be copied forward.</p> <p><b>Description:</b> When a representable is copied, copy forward all the representations or only those that would normally be copied forward.</p>
publish.copyforwardallrepresentationsonrevise	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> When a representable is revised, copy forward all the representations or only those that would normally be copied forward.</p> <p><b>Description:</b> When a representable is revised, copy forward all the representations or only those that would normally be copied forward.</p>
publish.copymarkupsforward	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allows the copy forward of Annotation and Groups when enabled.</p> <p><b>Description:</b> Applies in the case where a Representation is copied forward when a Part iterates. If this is enabled, the associated Annotation and Groups will also be copied.</p>
publish.copymarkupsrestricttosametype	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Allow the copy markups to be restricted to copying from a representation to a representation and a viewable to a viewable only.</p> <p><b>Description:</b> When copying markups, allows the copy to be restricted to copying from a representation to a representation and a viewable to a viewable only.</p>
publish.copyrepresentationsforward	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> When enabled, allows Representations to be copied forward (to the next iteration) when Parts iterate.</p> <p><b>Description:</b> Specifies if the copy forward of Representations takes place. This will take place if set to true and the copy does not compromise the validity of data published from EPM.</p>
publish.copyrepresentationsforward.restrict	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Restrict representation copy forward mode.</p> <p><b>Description:</b> Restricted representation copy forward mode does not copy a published representation forward when a WTPart iterates and old and new iteration point to the same EPMDocument.</p>
publish.createpublishaudit	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Create publish audit information for each publish job.</p> <p><b>Description:</b> Create publish audit information for each publish job.</p>

Windchill Visualization Service Property	Description
publish.createrepresentationcadtypes	<p><b>Default Value:</b> PROE CADDSS5 CATIA PRODESKTOP UG SOLIDWORKS CATIAV5 OTHER</p> <p><b>Synopsis:</b> List of CAD types that can be loaded or converted from the <b>Create Representation</b> window.</p> <p><b>Description:</b> Lists CAD types that can be loaded or converted from the <b>Create Representation</b> window on a WTPart without an EPMDocument. This is a space-separated list of keys from EPMAuthoringAppTypeRB.rinfo. If the type OTHER is included, the file to be processed is treated as if it is on a document: the document worker mapping determines the conversion that occurs.</p>
publish.deletepreconvertededz	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Delete temporary PVZ file.</p> <p><b>Description:</b> When publishing an EPMDocument that has a PVZ file of preconverted data (client side-generated viewables), deletes the PVZ file when the EPMDocument is published.</p>
publish.documents.copymarkupsforward	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Set copy forward of markups for WTDocuments.</p> <p><b>Description:</b> Sets copy forward of markups for WTDocuments. Individual markups are copied forward only if that markup has its copy forward flag set.</p>
publish.documents.copyrepresentationsforward	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Set copy forward of representations for WTDocuments.</p> <p><b>Description:</b> Sets copy forward of representations for WTDocuments. All representations of WTDocuments are candidates for copy forward, even if they are published from the document content files. Publishing occurs only when document files are uploaded. If markups on the WTDocuments representations are copied forward, the representations are not replaced by publishing.</p>
publish.forcerepublish	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Enables forcing of republishing existing Representations.</p> <p><b>Description:</b> Forces the republishing of existing Representations that are already valid, (for example, when CAD Worker settings have been changed and it is required to reconvert data.)</p>

Windchill Visualization Service Property	Description
publish.markonlydirectusesrepresentations outofdate	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Restrict marking out of date to only direct uses EPMDocuments.</p> <p><b>Description:</b> Restricts marking out of date to only direct uses EPMDocuments.</p>
publish.markoutofdatefiltermethod	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Method to filter which representations are marked out of date.</p> <p><b>Description:</b> By default, representations that are derived with a Latest Config Spec are candidates for being marked out of date. You can use a custom method to provide different criteria, for example, life cycle state. The method is defined by the following property in the form classname/methodname. The method should have the following signature: public static Boolean methodname(EPMDocument epmdoc, Representation rep). A return of Boolean TRUE indicates that the passed in representation of the passed in EPMDocument is a candidate for being marked out of date. A return of Boolean FALSE indicates that is should not be a candidate.</p>
publish.markoutofdateoncheckin	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Marks parent's representations out of date on checkin.</p> <p><b>Description:</b> When set to true, representations of parent objects created with a latest configuration spec will be marked out of date when the child is checked in.</p>
publish.markoutofdateonpdmcheckin	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Marks parent's representations out of date on PDM checkin.</p> <p><b>Description:</b> When set to true, representations of parent objects created with a latest configuration spec will be marked out of date when a PDM checkin is complete for a child.</p>
publish.markoutofdateonpublishcomplete	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Marks parent's representations out of date on publish.</p> <p><b>Description:</b> This is the default setting. To specify any other setting for marking out of date, such as on checkin or revise, you must set this value to False.</p>
publish.markoutofdateonrevise	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Marks parent's representations out of date on revision.</p> <p><b>Description:</b> When set to true, representations of parent objects created with a latest configuration spec will be marked out of date when the child is revised.</p>

Windchill Visualization Service Property	Description
publish.markoutofdaterepublishmethod	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Method to filter which representations that are marked out of date should be sent for republishing.</p> <p><b>Description:</b> When a representation is marked out of date for the first time, it can be republished. This causes many extra publishing jobs. You can define a method to use to filter which EPMDocuments should have the representation automatically republished, for only released data could be selected. The method is defined by the following property in the form classname/methodname. The method should have the following signature:</p> <pre>public static Boolean methodname(EPMDocument epmdoc, Representation rep).</pre> <p>A return of Boolean TRUE indicates that the passed in representation of the passed in EPMDocument is sent for republishing; a return of Boolean FALSE indicates that it should not.</p> <p>For example, the markRepublishAll method sends all representations for republishing.</p> <pre>publish.markoutofdaterepublishmethod=com.ptc.wvs.server.publish.PublishHelper/markRepublishAll</pre> <p>When representation are not sent for republishing, the schedule job republishOutOfDate, or a customer schedule job, can be used to republish representations marked as out of data at a time best suited to system resource usage.</p>
publish.markpositioningasassemblyrepresentationsoutofdate	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Provide a warning to the user when viewing representations marked as out of date.</p> <p><b>Description:</b> Representations can be marked as out of date which provides a warning to the user when viewing that representation. Generally, positioning assemblies do not need to be marked out of date, as typically they are using a Latest config spec to select the latest children at the time of viewing.</p>
publish.markreferencesrepresentationsoutofdate	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Provide a warning to the user when viewing representations marked as out of date.</p> <p><b>Description:</b> Representations can be marked as out of date which provides a warning to the user when viewing that representation. When an EPMDocument is published, it can find referencing EPMDocuments and mark their representations that are older than the EPMDocument being published and use a Latest Config Spec as being potentially out of date.</p>

Windchill Visualization Service Property	Description
publish.markusesrepresentationsoutofdate	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Provide a warning to the user when viewing representations marked as out of date.</p> <p><b>Description:</b> Provides a warning to the user when viewing representations marked as out of date. When an EPMDocument is published, it can find using EPMDocuments and mark their representations (that are older that the EPMDocument being published and use a Latest Config Spec) as being potentially out of date. If required, the marking of uses of representations be limited to representation that directly use the object being published.</p>
publish.matchcadnames	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allows matching of data from CAD system with Windchill when enabled.</p> <p><b>Description:</b> Enables matching of data from CAD system with Windchill for "file based" systems. Allows Windchill property pages to be referenced from ProductView and population of Windchill properties in ProductView. Also implied when transform information is being populated from EPM/CAD System to Part structure.</p>
publish.monitor.displaylimit	<p><b>Default Value:</b> 250</p> <p><b>Synopsis:</b> Display limit for number of jobs in the publish monitor.</p> <p><b>Description:</b> Specifies the display limit for jobs in the publish monitor. The value of the limit is the number of ready and the number of executing/completed jobs (for example, a limit of 250 displays up to 250 ready jobs and 250 executing/completed jobs).</p>
publish.positioningassembly.filename publish.positioningassembly.filtermethod publish.positioningassembly.ibaname	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Used to identify assemblies to be published as "positioning assemblies."</p> <p><b>Description:</b> Used to identify assemblies to be published as "positioning assemblies." See wvs.properties.xconf for details.</p>
publish.positioningassembly.publishchildren	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> When publishing a positioning assembly, publish children.</p> <p><b>Description:</b> When publishing a positioning assembly, publish children if the child does not have a representation for the positioning assembly to use.</p>

Windchill Visualization Service Property	Description
publish.publishqueue.priorities."type"."source"	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Defines the priority of queue that a particular type of publish job, from a particular source, will use.</p> <p><b>Description:</b> Defines the priority of queue that a particular type of publish job, from a particular source, will use. See wvs.properties.xconf for details.</p>
publish.publishqueuehelper.verbose	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Enables debug information for publisher queues.</p> <p><b>Description:</b> Enables debug information for publisher queues and schedule queue jobs.</p>
publish.publishqueuepollinterval	<p><b>Default Value:</b> 5</p> <p><b>Synopsis:</b> Polling interval for PublisherQueue to look for free queues.</p> <p><b>Description:</b> Specifies the polling interval (in seconds) for PublisherQueue to look for free queues.</p>
publish.republishondocumentchange	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Republish or update representation when metadata on a WTDocument is changed and the WTDocument does not iterate.</p> <p><b>Description:</b> When the metadata on a WTDocument is changed and the WTDocument does not iterate (for example, a life cycle state is changed) the associated representations are updated. In certain cases (for example, when an OFFICE file on the WTDocument has an association to that metadata), it may be desirable to republish the representation rather than just updating it.</p>
publish.republishonepmdocumentchange	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Republish or update representation when metadata on an EPMDocument is changed and the EPMDocument does not iterate.</p> <p><b>Description:</b> When the metadata on an EPMDocument is changed and the EPMDocument does not iterate (for example, a life cycle state is changed) the associated representations are updated. In certain cases (for example, when the CAD file on the EPMDocument has an association to that metadata), it may be desirable to republish the representation rather than just updating it.</p>
publish.retrieveallfiles	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Retrieve all dependent files.</p> <p><b>Description:</b> If true, retrieves all files; if false, retrieves only those files whose links are marked as required.</p>



Windchill Visualization Service Property	Description
publish.service.addprohibitiononpublishfailure."type". "source"	<b>Default Value:</b> N/A <b>Synopsis:</b> Automatically add a publishing prohibition when a type of publish job from a specific source fails. <b>Description:</b> Automatically add a publishing prohibition when a type of publish job from a specific source fails. See wvs.properties.xconf for details.
publish.service.documents.checkin.enabled	<b>Default Value:</b> True <b>Synopsis:</b> Send WTDocuments for publishing on checkin. <b>Description:</b> Specifies if WTDocuments are to be sent for publishing on checkin. Only publishable files (those with an associated worker) are considered. All publishable files on the checked in document are sent for publishing. In addition, any files uploaded to a non-checkout document are published (for example, a new document or one in the user's personal cabinet). This option is mutually exclusive with publish.service.document.upload.enabled.
publish.service.documents.options	<b>Default Value:</b> <b>Synopsis:</b> Pass options to document publishing. <b>Description:</b> Pass options for document publishing. Comma-separated list of name=value pairs. For example: encodefilename={true, false} - encode non-ascii filename before sending to worker encoding= - specify the character encoding for the PVS file, for example, SJIS or UTF-8
publish.service.documents.upload.enabled	<b>Default Value:</b> False <b>Synopsis:</b> Send WTDocuments for publishing on each file upload. <b>Description:</b> Specifies if WTDocuments are to be sent for publishing on each file upload. Only publishable files (those with an associated worker) are considered. This option is mutually exclusive with publish.service.document.checkin.enabled.
publish.service.documents.upload.markoutofdate	<b>Default Value:</b> True <b>Synopsis:</b> Mark document representation, published from a file on the document, as out of date when a new version of the file is uploaded. <b>Description:</b> Marks document representation, published from a file on the document, as out of date when a new version of the file is uploaded. The file is not sent for publishing (for example, for an update when publishing on checkin or if publish on checkin and upload are disabled).

Windchill Visualization Service Property	Description
publish.service.enabled	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Provides listener for publishing CAD data on checkin when enabled.</p> <p><b>Description:</b> Enables WVS listener that listens for EPM CheckIn complete events. Checked in objects will be sent for publishing.</p> <p><b>Note:</b> This property is not typically used anymore and should be left with the default setting of false. See publish.service.readytopublish.enabled.</p>
publish.service.filterdocumentpublishmethod	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Supply method to provide custom filtering of the WTDocument content to be published.</p> <p><b>Description:</b> Specifies a method to provide custom filtering of the WTDocument content to be published as a result of file upload or checkin (provided the content has a worker associated with it). If no method is supplied, all WTDocument content that meets the above criteria is sent for publishing. The property value is specified in the form classname/methodname with the following signature:  publish static Boolean methodname(WTDocument doc, ContentItem ci)  A return of Boolean TRUE indicates the WTDocument should be published, Boolean FALSE indicates that it should not be published.</p>
publish.service.filterepmdocumentpublishmethod	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Custom filtering of EPMDocuments published as a result of CheckIn Complete or Ready To Publish events.</p> <p><b>Description:</b> You can supply a method to provide custom filtering of the EPMDocuments that are published as a result of a CheckIn Complete or Ready To Publish event. If no method is supplied, all EPMDocuments in the event are sent for publishing. The property value is specified in the form classname/methodname, with the following signature:  public static Boolean methodname(EPMDocument epmdoc)  a return of Boolean TRUE indicates the EPMDocument is published, Boolean FALSE, the EPMDocument is not published.</p>

Windchill Visualization Service Property	Description
publish.service.filterpublishmethod	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Supply a method to provide custom filtering of objects that can be published.</p> <p><b>Description:</b> Allows you to supply a method to provide custom filtering of objects that can be published. This method checks all publishing, even from preconverted. If the publish is to convert data stored in Windchill, the flag publishFromDB is true; if the publish is for local data or data from the clipboard, publishFromDB is false. The property value is specified in the form classname/methodname, with the following signature: public static Boolean methodname(Persistable p, Boolean publishFromDB). A return of Boolean true indicates the object can be published; Boolean false indicates it cannot.</p>
publish.service.ignoredefaultepmevents	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Restrict EPMDocument publishing from CheckIn Complete or Ready To Publish events.</p> <p><b>Description:</b> Restricts EPMDocument publishing from CheckIn Complete or Ready To Publish events to cases where the client initiating the event, for example the work group manager, has specified options for the creation of the representation. If this option is set, default CheckIn Complete or Ready To Publish events are ignored.</p>
publish.service.ignoreeventconfigspec	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Ignore the config spec specified in Ready To Publish events.</p> <p><b>Description:</b> Ignore the config spec specified in Ready to Publish events. WVS will then use either a latest or as stored config spec for the publishing.</p>
publish.service.onlypublishpreconvertededz	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Restrict EPMDocument publishing to PVZ files.</p> <p><b>Description:</b> When using PVZ client side viewables, restricts publishing of EPMDocuments to those that have PVZ files. This limits all publishing to be only from a temporary PVZ file (for example, client side viewables).</p>
publish.service.onlypublishpreconvertededzfrom events	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Restrict EPMDocument publishing to PVZ files, for the check-in listener only.</p> <p><b>Description:</b> When using PVZ client side viewables, restricts publishing of EPMDocuments to those that have PVZ files. This limits publishing initiated by an event only.</p>

Windchill Visualization Service Property	Description
publish.service.readytopublish.enabled	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Listen for Ready To Publish event to initiate publishing of EPMDocuments.</p> <p><b>Description:</b> Listens for Ready To Publish event to initiate publishing of EPMDocuments. Workgroup managers should emit this event for EPMDocuments that require visualization.</p>
publish.service.verbose	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Enables debug mode for publish listener.</p> <p><b>Description:</b> Enables debug mode for publish listener.</p>
publish.tempdir	<p><b>Default Value:</b> \$(wt.temp)\\pubtemp</p> <p><b>Synopsis:</b> Temporary directory for WVS publisher.</p> <p><b>Description:</b> Stores WVS publisher's temporary files in a directory.</p>
publish.tempuploadir	<p><b>Default Value:</b> \$(publish.tempdir)</p> <p><b>Synopsis:</b> Temporary directory for uploaded files.</p> <p><b>Description:</b> Creates a temporary directory for uploading files when creating representations from local files. For a cluster environment, this directory must be a shared, common directory, accessible to all involved method servers.</p>
publish.viewable.copymarkupsforward	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allow markups associated directly to a viewable (for example, WTPart or WTDocument) to be copied forward when the viewable iterates.</p> <p><b>Description:</b> Allows markups associated directly to a viewable (for example, WTPart or WTDocument) to be copied forward when the viewable iterates. An individual markup is copied only if its copy forward flag is set.</p>
schedulejobsN	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Internal name for a schedule job definition.</p> <p><b>Description:</b> Specifies the internal name for a schedule job definition. All other entries for this job definition will use this name as the prefix on the property name.</p>
"schedulejobname".description	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Display name for the schedule job definition.</p> <p><b>Description:</b> Specifies the display name for the schedule job definition.</p>

Windchill Visualization Service Property	Description
"schedulejobname".enableOnContainers	<b>Default Value:</b> N/A <b>Synopsis:</b> Should this schedule job be available in the context of a specific context (for example, a Product) <b>Description:</b> The queries in the schedule job should have been written appropriately in this option is set to be set to true.
"schedulejobname".class	<b>Default Value:</b> N/A <b>Synopsis:</b> Java class for the schedule job definition. <b>Description:</b> Specifies the Java class for the schedule job definition.
"schedulejobname".method	<b>Default Value:</b> N/A <b>Synopsis:</b> Java method for the schedule job definition. <b>Description:</b> Specifies the Java method for the schedule job definition.
scheduler.user	<b>Default Value:</b> <b>Synopsis:</b> User name used by scheduler. <b>Description:</b> Specifies the user name used by scheduler. If no user name is specified, schedule queue jobs will be executed as the user who submitted them.
thumbnail.byworker.enabled	<b>Default Value:</b> True <b>Synopsis:</b> Flag to enable thumbnail generation by a CAD worker. <b>Description:</b> Flag to enable thumbnail generation by a CAD worker. The worker itself must be configured to generate thumbnails as well, or else this setting will have no effect.
thumbnail.cadagenttimeout	<b>Default Value:</b> 3600 <b>Synopsis:</b> Timeout in seconds. <b>Description:</b> Sets the timeout in seconds for thumbnail generation when using the cadagent and a remote thumbnail worker.
thumbnail.enabled	<b>Default Value:</b> True <b>Synopsis:</b> Flags to enable thumbnail image generation. <b>Description:</b> Enables the generation of thumbnail images that are displayed on listings and property pages. The default thumbnail generation method uses Java3D, hence this must be installed on the Windchill Server machine.
thumbnail.filelimit	<b>Default Value:</b> 800 <b>Synopsis:</b> Maximum number of files for thumbnail generation. <b>Description:</b> Sets a limit on the size of an assembly (by the number of referenced files) that will have a thumbnail image generated. This can be used to control server resources.

Windchill Visualization Service Property	Description
thumbnail.generator.backcolor	<p><b>Default Value:</b> 183,183,183</p> <p><b>Synopsis:</b> Sets the image background color for thumbnail generator.</p> <p><b>Description:</b> Specifies the image background color to be created by the thumbnail generator. The three numbers represent red, green, and blue in the range 0 to 255.</p>
thumbnail.generator.extents	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables extents generation for thumbnail generator.</p> <p><b>Description:</b> Specifies that the thumbnail generator should compute the bounding box for each OL file that is processed. The bounding box is stored in the PVS file, and is used by ProductView for proximity search and file loading optimization.</p>
thumbnail.generator.height	<p><b>Default Value:</b> 128</p> <p><b>Synopsis:</b> Sets the image height for thumbnail generator.</p> <p><b>Description:</b> Specifies the image height to be created by the thumbnail generator.</p>
thumbnail.generator.nativecmd	<p><b>Default Value:</b> set by Visualization - Windchill Support CD installer</p> <p><b>Synopsis:</b> Executes the native thumbnail generator.</p> <p><b>Description:</b> Specifies the command to execute the native thumbnail generator, which creates 2D thumbnail images and 3D thumbnail files. The value of this property is set by the Visualization - Windchill Support CD installer.</p>
thumbnail.generator.rx	<p><b>Default Value:</b> -60</p> <p><b>Synopsis:</b> Sets image X axis rotation for thumbnail generator.</p> <p><b>Description:</b> Specifies the image X axis rotation in degrees.</p>
thumbnail.generator.ry	<p><b>Default Value:</b> -25</p> <p><b>Synopsis:</b> Sets image Y axis rotation for thumbnail generator.</p> <p><b>Description:</b> Specifies the image Y axis rotation in degrees.</p>
thumbnail.generator.rz	<p><b>Default Value:</b> -10</p> <p><b>Synopsis:</b> Sets image Z axis rotation for thumbnail generator.</p> <p><b>Description:</b> Specifies the image Z axis rotation in degrees.</p>

Windchill Visualization Service Property	Description
thumbnail.generator.width	<p><b>Default Value:</b> 192</p> <p><b>Synopsis:</b> Sets the image width for thumbnail generator.</p> <p><b>Description:</b> Specifies the image width to be created by the thumbnail generator.</p>
thumbnail.generator.zoomfactor	<p><b>Default Value:</b> 1.2</p> <p><b>Synopsis:</b> Sets image zoom factor for Java thumbnail generator. This is not used if the native thumbnail generator is installed.</p> <p><b>Description:</b> Specifies the image zoom factor, allowing the object to be made larger or smaller in the thumbnail image.</p>
thumbnail.usecadagent	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Set to true if thumbnail generation is performed on a remote system.</p> <p><b>Description:</b> Executes thumbnail generation via the CadAgent by configuring a worker type THUMBNAIL and using the GenericWorker. In most cases this is not required, but if the Windchill server is not capable of running the thumbnail generator, this does allow a remote machine to be used to execute the actual thumbnail generation process.</p> <p>This will be less efficient than executing directly.</p>
viewer.<DataFormatName>	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Maps Windchill file type to ProductView viewer type.</p> <p><b>Description:</b> Maps Windchill file type to ProductView viewer type. Windchill file type (the DataFormat name) specified here should have any spaces removed. Additional entries of this form can be added for any file types (for example, viewer.GIFImage=image).</p>
viewer.<FileExtension>	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Maps file extension to ProductView viewer type.</p> <p><b>Description:</b> Maps file extension to ProductView viewer type. Note that the file extension includes the dot, and should be specified in upper case. There is an alternative technique for associating a file with a specific file viewer, using the DataFormat name. Additional entries of this form can be added for other file extensions (for example, viewer.HPGL=drawing).</p>

Windchill Visualization Service Property	Description
webpage.allowdeleterepresentation	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allows representation deletion.</p> <p><b>Description:</b> Provides the ability for a user to delete a representation from the user interface. If true, all users can delete representations from the user interface; if false, no users can delete representations from the user interface. If admin, only administrators (system, project, or product administrators) can delete representations from the user interface.</p>
webpage.allowmakedefaultrepresentation	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allows default representation selection.</p> <p><b>Description:</b> Provides the ability for a user to change the default representation from the user interface. If true, all users can change the default representation from the user interface; if false, no users can change the default representation from the user interface. If admin, only administrators (system, project, or product administrators) can change the default representation from the user interface.</p>
webpage.allowpublish.epmdocument webpage.allowpublish.wtpart webpage.allowpublish.wtdocument	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Allows publishing from the user interface.</p> <p><b>Description:</b> Provides the ability for a user to publish from the user interface, on a type basis. If true, all users can publish the specified type from the user interface; if false, no users can publish the type from the user interface. If admin, only administrators (system, project, or product administrators) can publish from the user interface.</p>
webpage.autoload	<p><b>Default Value:</b> single</p> <p><b>Synopsis:</b> Enables autoloading of files in ProductView.</p> <p><b>Description:</b> Allows ProductView to autoload files. If value is "true" all files will be autoloading. If value is "single", PVS files with a single associated file will be autoloading. If value is "false" no files will be autoloading.</p>
webpage.defaultquerytype	<p><b>Default Value:</b> 21</p> <p><b>Synopsis:</b> Sets the default query type for WVS portal, used only in Windchill.</p> <p><b>Description:</b> Specifies default query type for the WVS portal page. The types supported are: 21 = WTParts, 22 = WTDocuments, 49 = EPMDocuments, 55=ProductInstances, 111=ProductConfigurations.</p>



Windchill Visualization Service Property	Description
webpage.displayproductthumbnail	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays thumbnail for ProductInstance and ProductConfiguration.</p> <p><b>Description:</b> For the view link for ProductInstance and ProductConfiguration, displays the thumbnail image. This gives only an approximate rendition.</p>
webpage.displayviewthumbnailfor2d	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays thumbnail viewer for 2D thumbnails.</p> <p><b>Description:</b> On the Representations listing page, displays the thumbnail viewer popup icon when there is only a 2D thumbnail (that is, no 3D thumbnail); otherwise, there must be a 3D thumbnail and the thumbnail view must be installed on the server.</p>
webpage.flagoutofdate	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays warning when viewing an out of date representation.</p> <p><b>Description:</b> Displays warning when viewing a representation that has been marked as out of date.</p>
webpage.longlistingdefault	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Set the long listing default for WVS portal.</p> <p><b>Description:</b> Specifies default for the Long List setting on the WVS portal.</p>
webpage.markupenabled	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables the creation and storing of markups from ProductView.</p> <p><b>Description:</b> Enables the creation and storing of markups from ProductView.</p>
webpage.partstructureview	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables display of the part structure view option.</p> <p><b>Description:</b> Allows part structures to be viewed from the WVS portal and the Product Structure web page, without the publishing of CAD data and the creation of a representation.</p>
webpage.pastemarkupswithrepresentation	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Adds (with add from collection) annotations and groups with a representation.</p> <p><b>Description:</b> Allows any associated annotation and groups to also be added, when a representation is added from the collection.</p>
webpage.showedrdelete	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Sets the delete display for WVS portal.</p> <p><b>Description:</b> Specifies the display of a delete option on the WVS portal.</p>

Windchill Visualization Service Property	Description
webpage.showextendedclipboard	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables the add to collection and add from collection facilities to allow representations and/or markups to be copied.</p> <p><b>Description:</b> Enables additional add to collection and add from collection facilities on the <b>Representations, Annotation and Groups</b>, and <b>Visualization Collection</b> pages.</p>
webpage.showpublishfordoc	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Display publish link for documents with publishable files.</p> <p><b>Description:</b> For a document with publishable files (that is, worker mapping defined with a "worker.xxx=yyy" property) displays the publish link when there is no default representation. The default publish link publishes the first publishable file on the document.</p>
webpage.showrepfordoc	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Display link to representation or markup page for documents.</p> <p><b>Description:</b> For a document with no representations and no markups and no displayed publish link, displays the link to either a markups page or a representations page.</p>
webpage.showrepforpart	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays representation link for a part.</p> <p><b>Description:</b> When a part displays the part structure link and has no markups, displays the link to the representation listing. If false, displays the link to the markups listing.</p>
webpage.showsavezip	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays the save representation action.</p> <p><b>Description:</b> Displays the save representation action in the representation listing. This allows you to save the representation as a local ZIP or JAR file or a link to view the representation to be saved (to include in an e-mail, for example).</p>
webpage.showstructureforpart	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Displays view default structure link for a part.</p> <p><b>Description:</b> When a part has no default representation and no associated EPMDocument, displays the view default structure link.</p>

Windchill Visualization Service Property	Description
webpage.showthumbnail	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Sets the thumbnail display default for WVS portal.</p> <p><b>Description:</b> Specifies default for the display of thumbnails on the WVS portal. Individual users can set their preference for thumbnail display from preference settings.</p>
webpage.verbose	<p><b>Default Value:</b> False</p> <p><b>Synopsis:</b> Provides user interface debug information.</p> <p><b>Description:</b> Provides user interface debug information.</p>
worker.<DataFormatName>	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Maps Windchill file type to a worker type.</p> <p><b>Description:</b> Maps Windchill file type to a worker type. A worker of this type should have been configured in the agent.ini file (for example, worker.VRML=VRML).</p>
worker.<FileExtension>	<p><b>Default Value:</b> N/A</p> <p><b>Synopsis:</b> Maps file extension to a worker type.</p> <p><b>Description:</b> Maps file extension to a worker type. Note that the file extension includes the dot, and should be specified in upper case. There is an alternative technique for associating a file with a specific file viewer, using the DataFormat name. A worker of this type should have been configured in the agent.ini file (for example, worker.IGES=IGES).</p>
wvs.demo.data	<p><b>Default Value:</b> \$(wt.home)\\loadFiles\\wvs\\</p> <p><b>Synopsis:</b> Directory where WVS demo data is located.</p> <p><b>Description:</b> Specifies the location of the WVS demo data. This is referenced only in the WVS demo data ini files. A file delimiter is required at the end of the entry.</p>
wvs.edencoding	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Character encoding for communication to server.</p> <p><b>Description:</b> Specifies the character encoding used by the servlet engine.</p>
wvs.edfileencoding	<p><b>Default Value:</b></p> <p><b>Synopsis:</b> Character encoding for reading PVS files.</p> <p><b>Description:</b> Specifies the default encoding to use for reading PVS files. If none specified, then the system uses the default character encoding of the server. See also <b>edrload.encoding</b>.</p>

Windchill Visualization Service Property	Description
wvs.enabled	<p><b>Default Value:</b> True</p> <p><b>Synopsis:</b> Enables Windchill Visualization Service.</p> <p><b>Description:</b> Enables Windchill Visualization Service. When set to false, users will not see thumbnails or be able to launch ProductView. Before visualization can be fully used, other parts of the system, such as CAD Workers, need to be installed and configured.</p>
wvs.etbencoding	<p><b>Default Value:</b> UTF-8</p> <p><b>Synopsis:</b> The character encoding used for the ProductView ETB file.</p> <p><b>Description:</b> The character encoding used for the ProductView ETB file.</p>

# 21

## Auditing

This chapter provides information about auditing events, license usage reporting, organization usage reporting, and an example of creating an audit report using the Auditing Administration utility.

Topic	Page
Overview of the Auditing Event Framework .....	21-2
Out-of-the-Box Configuration for Audit Event Recording.....	21-2
Supported Auditing Events .....	21-3
Accessing the Auditing Administration Utility.....	21-12
Example Audit Report Query .....	21-16

## Overview of the Auditing Event Framework

The auditing framework provides a mechanism to enable event-based logging that provides a historical record of who did what that caused changes in the Windchill database. The security auditing feature also provides a mechanism to retain historical records of security related events, such as privilege changes, team changes, and denial of access. Security audit reporting is helpful to our customers in highly regulated industries that need to identify events that may have resulted in a potential security breach. The following list provides possible application uses that can be recorded:

- Who accessed the application
- Who changed an object
- Who viewed or downloaded an object
- Who granted access to a specific user

The Auditing framework is a Windchill service that captures events generated through normal application use, using a collection of classes and configuration files that a customer modifies to indicate which specific events are to be recorded in the audit log.

## Out-of-the-Box Configuration for Audit Event Recording

Out-of-the-box, the only events that are enabled out-of-the-box are **License Usage Reporting** and **Organization Usage**. Any other specific events that you want recorded in the audit log must be enabled on an individual basis.

The auditing configuration file (Windchill/codebase/registry/auditing/configAudit.xml by default) determines which events are recorded for which object types. The configuration file is delivered with some predefined service events and summary events that can be captured and reported on. You will need to analyze your company's auditing rules to determine which specific events you want audited.

If you choose to enable additional events, add, or change the out-of-the-box auditing rules, PTC recommends that you follow the maintenance best practices for modifying PTC files as described in the *Windchill Customizer's Guide*. Using the best practices, you create a backup of the original and apply your changes to a new file.

By enabling only specific events, you thereby minimize the size of the audit log by preventing all events from automatically accumulating in the audit log.

## Supported Auditing Events

The two types of predefined events that are supported are service events and summary events. The following sections provide a list of each type of event and their descriptions. For more information on enabling these events, see the *Windchill Customizer's Guide*.

### Service Events

Some service level events are captured by the auditing service. The service events captured are based on Windchill's service event architecture that was designed to manage inter-service communication between independent services. Each service establishes "listeners" to other services' events in order to be notified of and react to other significant events occurring within the system.

The following table lists the supported service events the auditing service will capture when enabled:

Service Event	Description
Archive	Whenever an archive operation is performed, service event is emitted to support auditing of archive operations.
Context Logon	Whenever a user accesses an organization container or any of the application containers in the organization, this event is emitted.
Project Plan Deadline Date Change	Whenever a deadline changes for Project or Program Activity, Milestone, or Deliverable this event will be emitted.
Download	Whenever the primary or secondary content of an access controlled object is downloaded or a direct download URL is provided (i.e., requires read content permission), a service event is emitted.  Viewing representations is not captured with this event.
Project Plan Finish Date Change	Whenever a Finish date changes for a Project or Program Activity, Milestone, or Deliverable this event will be emitted.
Log off	Whenever a user's session times out after the user has been inactive for a site defined period (30 minutes by default) this event will be emitted.

<b>Service Event</b>	<b>Description</b>
Logon	Whenever a user successfully accesses the URL to the system that prompts the user to log in and the user successfully logs in, the Log-On service event is emitted.
Modify Group	<p>Whenever a user modifies a Windchill group, this event is emitted and identifies the specific group change that occurred.</p> <p>This event is emitted when a user is added or removed from a user-defined group or a system group. And when a group is added or removed from a user-defined group or a system group.</p>
New Discussion	Whenever a new discussion is started for a Discussion Posting or Discussion Topic this event will be emitted.
Not Authorized Access	<p>This event is emitted whenever a user is denied access to an object that the user explicitly tries to access.</p> <p>This event will not be emitted when a user views an information page for a document that the user has Read access to but the user does not have access to the principal that created the document.</p>
Owner Change	Whenever an owner changes for a Project or Program Activity, Milestone, or Deliverable this event will be emitted.
Percent Change	Whenever a percent changes for a Project or Program Activity, Milestone, or Deliverable this event will be emitted.
Purge	Whenever a purge operation is performed, this event is emitted.
Restore	Whenever a restore of previously archived data is performed, this event is emitted.
Search	Whenever a search from the Windchill search application is executed, an event recording the search criteria will be emitted.
Status Change	Whenever a status changes for a Project or Program Activity, Milestone, or Deliverable this event will be emitted.



<b>Service Event</b>	<b>Description</b>
User Password Change	Whenever a user changes their password using Windchill's password change portal, this event is emitted.
View Attribute	Whenever a properties page is displayed, this event is emitted.  This event is not emitted when reviewing the result set of a search.
View Representations	Whenever a representation is downloaded, this event is emitted with the target of the event being the object that is the source of the representation.
Project Plan State Change	Whenever the state changes for a project plan or a program, this event will be emitted.
Workflow State Change	Whenever a workflow process state changes, this event will be emitted.
Workflow Variable Change	Whenever the variable values change for the workflow process, this event will be emitted.
Workflow Activity State Change	Whenever a workflow activity state changes, this event will be emitted.
Workflow Activity Variable Change	Whenever the variable values change for the workflow activity, this event will be emitted.
Add Role to Team	A role is added to a team object.
Remove Role from Team	A role is removed from a team object.
Add Role to Organization	A role is added to an organization group.
Remove Role from Organization	A role is removed from an organization group.
Project Plan Deadline Date Change	A project plan or program's deadline date is changed.
Project Plan Finish Date Change	A project plan or program's finish date is changed.
Project Plan Owner Change	A project plan or program's owner is changed.

<b>Service Event</b>	<b>Description</b>
Project Plan Percent Complete Change	The percent complete for a project plan or program is changed.
Project Plan Health Status Change	A project plan or program's health status is changed.

## Summary Events

Summary events will be emitted when a significant application level event occurs that would be of interest to the system auditor. The events provide a meaningful description of what happened in the database at the time the event took place. The events provide a set of attributes that identify who did what and provide context for the operation. Summary events were designed to be standalone so analysis can be done outside of Windchill using your favorite analysis tool.

The following table lists the supported summary events the auditing service will capture when enabled:

<b>Summary Event</b>	<b>Description</b>
Change Access Permissions	The access control or team defined for an object changed or the object is shared.
Check In	An object is checked in; a shared object is checked in from a project or program.
Check Out	An object is checked out.
Complete	The state of an object is set to Complete. This applies to objects such as plan objects.
Copy	An object is copied.
Delete	An object is deleted or moved out of a location.
Edit Attributes	One or more attributes for an object or its relationship to other objects are edited.
Edit Content	One or more content files for an object have been added, removed, or changed.
Edit Identity	One or more attributes that serves as a unique identifier for an object is edited.
Life Cycle Status	The life cycle state of an object is changed.
Markup	Markup of a viewable is saved or a user has annotated the product structure.
Move	An object is moved to a new folder.

Summary Event	Description
New Object	A new object is created; an object is imported or moved from a previous location.
New One-off Version	A new one-off version of an object is created, for example, by using Product Structure Explorer or by sharing an object from Windchill PDMLink to a Windchill ProjectLink project.
New Revision	A new version of an object is created as the result of a <b>New Revision</b> action performed on the object. This does not apply to new iterations resulting from an edit action or check in action.
New View Version	A new view version of an object is created.
Undo Check Out	A checkout is undone.
Used By	An object is defined as a child dependency in a relationship with another object in the product structure.
Change Access Policies	<p>This event is emitted whenever the ad-hoc access policies for an object change or the object changes domains. This event will also be emitted whenever the access policies for a domain change with the domain being the target of the event.</p> <p><b>Note:</b> Domain administered objects will not have this event emitted for them if the domain policy rules change for the domain in which they are contained.</p>
Export	An object is exported from the system. This event is not emitted when a search results table is exported to an Excel spreadsheet.
Import	An object is imported into the system.
Modify Team	A context team or team is modified.
Share	An object is shared from one context to another.

## Attributes Captured with Events

Attributes based on the target object of the event are captured with each event and written to the audit record.

In addition to capturing a set of attributes that will give the auditor a complete context of the event, transaction descriptions are captured for some events. Transaction descriptions are used to provide the auditor with a meaningful summary of the transaction. Auditing captures the attributes of the target object of the event at the beginning of the transaction. Additional attributes that are required to provide the auditor with the appropriate context for the events are captured as necessary.

The following attributes are captured (based on the target object):

Attribute	Description
Event Key	An internal identifier for the event.
Event Label	The name of the event.
Date/Time	The date and time when the event occurred.
User's Organization Name	The name of the organization that the user is a member of.
User Name (UID)	String that the user logs on with.
User ID	An identifier for the user.
Object Name	The name assigned to the object.
Object Reference	An internal identifier for the object.
Object Type	The type of the object.
Object Type Branch ID	An internal identifier for the type.
Object ID	The identifier for the object.
Object Number	The number assigned to the object.
IP Address	The IP address of the user who initiated the event.
Cage Code (if the organization is owned)	The cage code that is part of the object's identity if the object is organization-owned.
Branch ID	An internal identifier for the version tree branch.
Working Branch ID	An internal identifier for the version tree branch.

Attribute	Description
Version ID	The identifier for the version.
Master ID	The identifier for the object master.
Organization Container ID (referred to as the <i>organization identifier</i> throughout the documentation)	An internal identifier for the organization container.
Organization container name (referred to as <i>organization context</i> throughout the documentation)	The name of the organization container.
Application container ID (referred to as <i>application context</i> throughout the documentation)	An internal identifier for the application container.
Application container name (referred to as <i>application context</i> throughout the documentation)	The name of the organization container.
Application Container Type Branch ID	An internal identifier for the application container type.
Folder path	The full qualified path of the object.
Domain path	The fully qualified path of the domain hierarchy.
Identity (via identity service)	The full identity for the object.
Life Cycle State	The current state of the object that is the target of the object.
Transaction description(s)	Describes more details about what was happening in the application when the event was fired.

The table below summarizes the additional event dependent attributes specific to events that are captured and written to the audit log:

<b>Event</b>	<b>Additional Attributes</b>
Change Access Permissions	1. Whether the event represents an ad-hoc change or a domain change. 2. The ad hoc ACL and domain ACL for the target object.
Check In	The original (previous) iteration.
Check Out	The working copy folder path.
Copy	The new object (via identity service).
Edit Content	The names of the content items and whether they were added or removed.
Edit Identity	Old identity - the identity of the object before it was renamed.
Life Cycle State	From State.
Markup	Markup or AnnotationIdentity - the identity of the markup or the annotation.
Move	From folder path.
New One-off Version	Source object identity (via the identity service).
New Revision	Source object identity (via the identity service).
New View Version	Source object identity (via the identity service).
Used By	Identity of the object being used.
Change Access Policies	The principals, permissions, types, and states of the policies and whether they were added, modified, or removed.
Modify Team	Role participants (roles and principals) and whether they were added or removed.
Share	To folder path.
Download	Stream Data Filename - the name of the file that was downloaded.
Log-off	concurrencyValue - the number of concurrent users.

<b>Event</b>	<b>Additional Attributes</b>
Log-On	concurrencyValue - the number of concurrent users.
Modify Group	<ol style="list-style-type: none"> <li>1. changeType - the type of change made to the group.</li> <li>2. memberRef - an internal reference to the group.</li> </ol>
Not Authorized Access	<ol style="list-style-type: none"> <li>1. permission</li> <li>2. message - the authorization message.</li> <li>3. permissionMask - the full set of permissions associated with the user and object for the object that was the target of the event.</li> </ol>
Search	Search Criteria - the criteria entered when performing the search.
View Representations	Representation Name - the name of the representation that was viewed.
PROCESS_CONTEXT_CHANGED	<ol style="list-style-type: none"> <li>1. name - The name of the variable.</li> <li>2. variable type - The type of variable.</li> <li>3. value - The value for the variable.</li> </ol>
ACTIVITY_CONTEXT_CHANGED	<ol style="list-style-type: none"> <li>1. name - The name of the variable.</li> <li>2. variable type - The type of variable.</li> </ol>

## Accessing the Auditing Administration Utility

The **Auditing Administration** utility can be accessed from either the **Site** or the **Organization** tab. Click the **Utilities** link, then scroll to the bottom of the page and select one of the links under the **Auditing Administration** heading.

The **Auditing Administration** utility is used by site administrators for the following:

- Create and view report queries using a variety of filtering mechanisms (see [Audit Report Queries](#))
- View, generate, and export previously saved report queries (see [Viewing and Generating Report Queries](#))
- Manage the license count and distribute license usage reports (see [License Usage Reporting](#))
- Purge the audit log (see [Purging the Audit Log](#))

The **Auditing Administration** utility is used by organization administrators for the following:

- Create and view report queries using a variety of filtering mechanisms (see [Audit Report Queries](#))
- View, generate, and export previously saved report queries (see [Viewing and Generating Report Queries](#))
- Manage the organization usage (see [Organization Usage Reporting](#))

For information on how to use the Auditing Administration utility, see the online help.

## Audit Report Queries

The **Security Audit Reporting** utility provides the ability to create and view security audit reports and save report queries. The audit reports are created by filtering specific criteria in the audit log. Once the report criteria has been determined, a subset of the report content can be previewed, or the report can be generated in either a CSV or XML file format. The specified criteria can also be saved as a report query for later execution.

The filtering criteria includes the following:

- Time Period
- Event
- Organization



- Context or Context Type
- Object or Object Type
- User

For procedures on how to create or generate a report query, see the online help.

## Viewing and Generating Report Queries

From the **Security Audit Report Queries** link, you can view a list of saved audit report queries. Queries can also be generated and exported to a file. For details about the available actions, see the online help.

Saved queries remain private to the context in which they are created. For example, a query created and saved at the site cannot be seen from any of the organizations and any query created and saved within an organization remains visible only to that organization.

## Purging the Audit Log

The **Audit Log Purge Manager** removes large chunks of data by date from the audit log, using a First-In-First-Out (FIFO) methodology. The data in the audit log is never individually updated or deleted. The purge manager utility is only available from the site context to ensure that the audit records are purged in their entirety and not in a partial manner from one or more organizations.



**Caution:** Once the audit records are purged, they are permanently deleted. To preserve the data, it is recommended that you generate and export your queries to a file prior to purging the audit log.

For detailed procedures on creating, generating and exporting audit report queries, see the online help.

When specifying the date through which the audit records will be purged, the site administrator can choose to save the purged records to a Windchill document (wt.doc.WTDocument). These documents are in XML format and the files are listed in the **Archived Results** column of the **Audit Log Purge Manager** table. You can download the documents and view them using an XML viewer; however, the records are no longer available in the audit tables. For example, if you purge the audit log through December 31, 2006 and want to create a new query, only events that occurred after the last purged date will be pulled into the security audit reports.

The **Audit Log Purge Manager** table provides information about the purged results including the date the records were purged, the site administrator who purged them, the date through which the records were purged, and the number of records in each purge.

## License Usage Reporting

The **License Usage Reporting** link is available only from the Site context. Site administrators can manage the license count, view and distribute license usage reports, as well as determine license usage compliance. The content can also be exported to a file.

This feature only reports the actual usage, and does not provide a mechanism for preventing more users than are licensed from logging on.

## Licensing for Windchill and Arbortext

Windchill PDMLink, Windchill ProjectLink, and Arbortext licenses are purchased on a per-user basis. The **License Usage Reporting** utility provides a mechanism for administrators to view the number of users for a given time period and license compliance for that usage period.

A site is considered compliant if, during a given month, the number of internal users does not exceed the number of purchased internal licenses, and the number of external users does not exceed the number of purchased external licenses.

An internal user is defined as any user that is a member of the internal organization (the organization that is specified as the *initial organization* during installation). An external user is defined as any user that is not a member of the internal organization.

The license usage report contains the following data:

- **Usage Period** - the time interval being reported
- **Internal License Count** - the number of purchased internal licenses
- **Internal Users** - the number of internal users
- **External License Count** - the number of purchased external licenses
- **External Users** - the number of external users
- **Compliant** - an indication of whether or not the site is compliant for the reported time period

The user login and session expiration is captured as an event. When a session is initiated or terminated, an event records the session's user, decides if the user is internal or external, and captures the total number of sessions in the system. The session is terminated after 30 minutes of inactivity (this is a site configurable value). The system counts the license usage by the number of unique users that have accessed the system within a given day.

License counts are maintained on a monthly basis. If the counts are modified multiple times during a month, the last count of the month is interpreted as the count of record for determining license compliance. The site administrator can modify the license count at any time during the month.

## Licensing for Pro/INTRALINK

Pro/INTRALINK software licenses are purchased on a concurrent user basis. The **License Usage Reporting** utility provides a way for administrators to view the peak number of concurrent users during a usage period, and whether or not that usage is in compliance with the number of purchased licenses. The term *concurrent users* is defined as the total number of users accessing the Pro/INTRALINK system at any given time.

A site is considered compliant if, during a given month, the number of concurrent Pro/INTRALINK users does not exceed the number of purchased licenses.

A user login is captured as an event. When a session is initiated, an event captures the session's user and the total number of users in the system. The session is terminated after 30 minutes of inactivity (this is a site configurable value).

## License Compliance Notification Emails

License usage reports are e-mailed to a designated list of recipients at the beginning of each month. The list of individuals receiving the e-mail is configurable, and site administrators can choose to include the e-mail address of a license auditing representative. The distribution list consists only of e-mail addresses, so the system makes no distinction between notifying Windchill users or non-Windchill users. The e-mail is delivered in HTML format.

## Organization Usage Reporting

The **Organization Usage Reporting** utility is available only from an organization context and is not available in Pro/INTRALINK.

Organization usage reports allow administrators to track system usage on an organization-by-organization basis. Windchill licenses are sold only at the site level, so the organization usage report contains no license compliance data.

An organization is considered to have been accessed by a user when data within the organization or information about the organization is displayed. The count of users accessing an organization is based on unique user accesses, not total accesses to the organization. Therefore, the count is incremented once per unique user.

Organization use is split into two categories: use by internal users and use by external users. The terms internal user and external user have a slightly different meaning than in the context of license usage reports. Internal users are users that are members of the organization being reported on, and external users are users that are not members of the organization.

The organization usage report contains the following data.

- **Usage Period** - the time interval being reported
- **Details** - link to the details page for the indicated usage period
- **Internal Users** - the number of internal users that have accessed the system within the reported time frame
- **External Users** - the number of external users that have accessed the system within the reported time frame

An audit event is only triggered the first time a user accesses an organization within a specific session.

## Example Audit Report Query

The following sections provide both general information for creating an audit report query and example for creating, saving, and exporting a query to a file.

The online help contains all the procedures for how to use the multiple features of the Auditing Administration utility.

### Creating an Audit Report Query

The following is an example for creating and saving an audit report query for the first quarter of 2007 for roles that were added to any organization and any team:

1. Navigate to the **Utilities** page on either the **Site** or the **Organization** tab.
2. Click the **Security Audit Reporting** link under the **Auditing Administration** heading.

The **Security Audit Reporting** screen appears.

**Security Audit Reporting**

**Time Period:**

☒ Duration: Today ▼

☐ Start date:  📅 yyyy-mm-dd End date:  📅 yyyy-mm-dd

**Event:**

☒ All event types

☐ Only events of type:  Find...

**Organization:**

☒ All organizations

☐ One organization:  Find...

**Context:**

☒ All contexts

☐ One context:  Find...

☐ Only contexts of type:  Find...

**Object:**

☒ All objects

☐ One object:  Find...

☐ Only objects of type:  Find...

**User:**

☒ All users

☐ One user:  Find...

Generate... Preview Save Query...

3. In the **Time Period** section, specify the time period for which you want the audit information captured in the report by selecting one of the following:

- From the **Duration** drop-down list, select one of the following options; **Today**, **Yesterday**, **Last 7 Days**, **Current Month**, **Last Month**, **Current Year**, or **Last Year**.

**Note:** All options in the **Duration** drop-down list are dynamic time periods and will always return data from the audit log for the respective time period based on the date the report is executed.

- Select the **Start date** and **End date** by clicking the calendar icon, or manually enter a date.

For this example, enter 2007-01-01 in the **Start date** and 2007-03-31 in the **End date**.

4. In the **Event** section, select the events you want to capture in the security audit report by using one of the following options:

- Select **All event types** to include all events in the report.
- Select the bullet for **Only events of type** and click **Find** to limit the events that you want included in the report.

For this example, select **Find** and select:

- **Add Role to an Organization**
- **Add Role to Team**

5. In the **Organization** section, select the organizations you want to capture in the security audit report by using one of the following options:

**Note:** The **Organization** section only appears on the Site tab.

- Select **All organizations** to include all organizations in the report.
- Select the **One organization** bullet and click **Find** to specify the organization that you want included in the report.

For this example, select **All organizations**.

6. In the **Context** section, select the contexts you want to capture in the security audit report by using one of the following options:

- Select **All contexts** to include all contexts in the report.
- Select the **One context** bullet and click **Find** to specify the context you want included in the report.
- Select the bullet for **Only contexts of type** and click **Find** to specify the context types that you want returned in the audit report.

For this example, select **All contexts**.

7. In the **Object** section, select the objects you want to capture in the security audit report by using one of the following options:

- Select **All objects** to include all objects in the report.
- Select the bullet for **One object** and click **Find** to specify the object that you want included in the report.
- Select the bullet for **Only objects of type** and click **Find** to specify the object types that you want returned in the audit report.

For this example, select **All objects**.

8. In the **User** section, select the users you want to capture in the security audit report by using one of the following options:
  - Select **All users** to include all users in the report.
  - Select the bullet for **One user** and click **Find** to specify the user you want included in the report.

For this example, select **All users**.

9. Click one of the following buttons:
  - **Generate** - to generate the report and export it to a file using CSV or XML file format.
  - **Preview** - to view a limited subset of the audit events.

**Note:** The **Preview** button is provided so the contents can be viewed prior to saving the query.

- **Save Query** - to save the defined parameters into a query to be executed at a later time.

**Note:** The saved report query can be accessed from the **Security Audit Report Queries** link on the **Utilities** page.

For this example, click **Save Query** and specify the name "Roles for Org and Team" in the window that appears.

A confirmation will appear that the report has been successfully saved.

## Example of Exporting a Saved Query

The following is an example for exporting the report query that was created and saved in the above example:

1. From the **Utilities** page, click the **Security Audit Report Queries** link.

The **Security Audit Report Queries** table displays the report that was just saved as **Roles for Org and Team**.

2. Click the **Actions** link and select **Generate Report**.

The **Generate Report and Export to File** window appears.

**Note:** You can also click the name of the report or the information icon ⓘ to view the details of the saved query.

3. From the **File Type** drop-down, select the type of file format in which you want the report saved.
  - **CSV** - (Comma Separated Values)  
This file format is compatible with Microsoft Excel.
  - **XML** - (Extensible Markup Language)  
This file format can be used with any XML editor.

For this example, select **CSV**.
4. Click **OK** to save the file.
5. A **File Download** window appears in which you can choose a location to save the report to the client.
6. Browse to the location to which you saved the file and open it in Excel.



# A

## Creating XML Files for Templates and Shared Teams

This appendix provides information to assist in creating business XML files that can be used when creating an organization template, creating or importing a product, library, program, or project template, or importing a shared team.

The appendix provides an explanation of the XML elements that are required and optional in the business XML files.

<b>Topic</b>	<b>Page</b>
Overview .....	A-2
Creating Business XML Files for Organization Templates .....	A-2
Creating Business XML Files for Product Templates.....	A-7
Creating Business XML Files for Library Templates .....	A-10
Creating Business XML Files for Program and Project Templates .....	A-13
Creating Business XML Files for Shared Teams .....	A-48
Common XML Elements Used in Multiple Business XML Files .....	A-50

## Overview

Business XML files are used as input for the **Import Template**, **Create Template**, and **Import Shared Team** actions provided in all Windchill solutions.

For details on the use of the **Import Template** and **Create Template** actions, see [Creating Context Templates](#) in the Context Templates chapter.

For details on the use of the **Import Shared Team** action, see the online help available from the **Teams** link on the **Organization** tab.

The definition of the XML elements can be found in standardX10.dtd.

## Creating Business XML Files for Organization Templates

The content of the business XML file used as input when creating an organization template consists of a single, complex XML element named OrganizationConfig. This element is defined by the following DTD elements:

```
<!ELEMENT OrganizationConfig (OrgStructure?,
                               ContainerDefaults*,
                               ProductTemplate*,
                               LibraryTemplate*,
                               ProjectTemplate*,
                               ContainerTemplate*,
                               loadXMLFile*,
                               NmLoader*,
                               WTypeDefinitions*,
                               TypeBasedRule*,
                               SeedObjectRuleSet*,
                               createFromRule*,
                               ProductStructure?,
                               AccessControlRule*,
                               DocumentTemplate*,
                               ProjectEnumType*,
                               DiscussionForumTemplate?,
                               NotebookTemplate?)>

<!--ATTLIST OrganizationConfig
      autoPopulateCreators (true | false) #IMPLIED
-->
```

The following subelements contained in the OrganizationConfig element are not supported:

- ContainerDefaults
- ProductTemplate -- superseded by the ContainerTemplate element
- LibraryTemplate -- superseded by the ContainerTemplate element
- ProjectTemplate -- superseded by the ContainerTemplate element
- NmLoader
- WTypeDefinitions -- for information about defining data types, see the [Type and Attribute Manager](#) chapter

- SeedObjectRuleSet -- superseded by the TypeBasedRule element
- createFromRule -- superseded by the TypeBasedRule element
- ProductStructure
- DocumentTemplate

To create the business XML file for an organization template, PTC recommends that you make a copy of an out-of-the-box organization template file (located in the <Windchill>/loadXMLFiles directory) and modify that template to produce a business XML file that you can use as input. For details on creating an organization template, see [Required Contents of ZIP File Used for Importing a Context Template](#) in the Context Templates chapter

**Tip:** When modifying the XML file, include comments to document your changes and use indentation to indicate the hierarchical nature of the XML structure. Also use an XML editor that can validate your XML document against the DTD.

The supported subelements of OrganizationConfig are described in the following sections.

## OrgStructure Element

OrgStructure is an optional element of OrganizationConfig that is also used within elements other than OrganizationConfig.

Typically, the OrgStructure element is not used within OrganizationConfig and can be omitted.

For the details on this element, see [Common OrgStructure Element](#).

## ContainerTemplate Element

ContainerTemplate is an optional element of OrganizationConfig that is used to define the product, project, program, and library template files that are available in an organization context.

The ContainerTemplate element can occur any number of times and is defined by the following DTD element.

```
<!ELEMENT ContainerTemplate (name,  
                             description?,  
                             containerClassName,  
                             defaultValuesRule?,  
                             locale?,  
                             enabled?,  
                             (xmlText | loadPath | zipEntry)?)>
```

If product, library, program, or project template files are referenced in an organization template file, then the templates are made available for use when the organization context is created. This means that the organization administrator does not need to create the templates. However, the organization administrator does need to assign Product, Library, or Program Creators if the templates are product, library, or program templates.

The following example illustrates how to use the ContainerTemplate element to add a product template, project template, library template, and program template to an organization template.

**Note:** The content of the containerClassName element must match these examples. Although, the DTD indicates that the business XML file can be specified in terms of XML text (xmlText element), a file path (loadPath element), or as a ZIP file (zipEntry element), only the loadPath element should be used in an organization template XML file. The value of the loadPath element is relative to the wt property wt.ixb.defaultLoadPath. This property defaults to *<Windchill>/loadXMLFiles*. For example, to specify a site-specific ProjectConfig element for a project template and put it into file *<Windchill>/loadXMLFiles/siteSpecificProjectTemplate.xml*, the entry is *<loadPath>siteSpecificProjectTemplate.xml</loadPath>*. (The zipEntry element is used when the ContainerTemplate element is used to define a template being imported; see [ZIP File Contents of an Exported Context Template](#) in the Context Templates chapter for details.)

```
<ContainerTemplate>  
  <name>Product Template Example</name>  
  <description>Description goes here</description>  
  <containerClassName>wt.pdmlink.PDMLinkProduct</containerClassName>  
  <enabled>true</enabled>  
  <loadPath>sampleProductTemplate.xml</loadPath>  
</ContainerTemplate>
```

```
<ContainerTemplate>  
  <name>Project Template Example</name>  
  <description>Description goes here</description>
```

```

    <containerClassName>wt.projmgmt.admin.Project2</containerClassName>
    <enabled>true</enabled>
    <loadPath>sampleProjectTemplate.xml</loadPath>
</ContainerTemplate>

<ContainerTemplate>
  <name>Library Template Example</name>
  <description>Description goes here</description>
  <containerClassName>wt.inf.library.WTLibrary</containerClassName>
  <enabled>true</enabled>
  <loadPath>sampleLibraryTemplate.xml</loadPath>
</ContainerTemplate>

<ContainerTemplate>
  <name>Program Template Example</name>
  <description>Description goes here</description>
  <containerClassName>wt.projmgmt.admin.Project2.Program</containerClassName>
  <enabled>true</enabled>
  <loadPath>sampleProgramTemplate.xml</loadPath>
</ContainerTemplate>

```

## loadXMLFile Element

loadXMLFile is an optional element of OrganizationConfig that is used to include inline reading of any number of other XML files. It is also used within elements other than OrganizationConfig.

For the details on this element, see [Common loadXMLFile Element](#).

## TypeBasedRule Element

TypeBasedRule is an optional element of OrganizationConfig that defines default values for objects. The element is also used within elements other than OrganizationConfig.

For the details on this element, see [Common TypeBasedRule Element](#).

## AccessControlRule Element

AccessControlRule is an optional element of OrganizationConfig that defines access control policy rules and is used within elements other than OrganizationConfig.

For the details on this element, see [Common AccessControlRule Element](#).

## ProjectEnumType Element

**Note:** ProjectEnumType is only applicable when Windchill ProjectLink is installed.

ProjectEnumType is an optional element of OrganizationConfig and is used to define the project and program roles. It is also used within elements other than OrganizationConfig.

For the details on this element, see [Common ProjectEnumType Element](#).

## DiscussionForumTemplate Element

DiscussionForumTemplate is an optional element of OrganizationConfig and is ignored for organization templates.

**Note:** This element is not ignored when used within ProjectConfig elements.

## NotebookTemplate Element

NotebookTemplate is an optional element of OrganizationConfig and is ignored for organization templates.

**Note:** This element is not ignored when used within ProjectConfig elements.

## Creating Business XML Files for Product Templates

The content of the business XML file used as input when creating a product template consists of a single, complex XML element named ProductConfig. This element is defined by the following DTD element:

```
<!ELEMENT ProductConfig (SharedTeamDef?,
    projectMember*,
    OrgStructure?,
    DiscussionForumTemplate?,
    ExportedRoleMemberMap?,
    ExportedGuestMembers?,
    ExportedTemplateFiltering?,
    TeamTemplate*,
    NmLoader*,
    AccessControlRule*,
    IndexPolicyRule*,
    TypeBasedRule*,
    TemplateDocument*,
    DocumentTemplate*,
    NotebookTemplate?,
    ProductContainer*,
    AdHocACLEntrySet?,
    PreferenceInstance*)>
```

The following subelements contained in the ProductConfig element are not supported:

- projectMember -- preferred method of defining roles is using the ExportedRoleMemberMap element (described in [Common ExportedRoleMemberMap Element](#))
- TeamTemplate
- NmLoader
- IndexPolicyRule
- TemplateDocument
- DocumentTemplate
- ProductContainer

PTC recommends that you export an existing product as a template, modify the business XML file (and associated files), and then import the template. For import details, see [Required Contents of ZIP File Used for Importing a Context Template](#) in the Context Templates chapter.

**Tip:** When modifying the XML file, include comments to document your changes and use indentation to indicate the hierarchical nature of the XML structure. Also, use an XML editor that can validate your XML document against the DTD.

The supported subelements of ProductConfig are described in the following sections.

### **SharedTeamDef Element**

SharedTeamDef is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common SharedTeamDef Element](#).

### **OrgStructure Element**

OrgStructure is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common OrgStructure Element](#).

### **DiscussionForumTemplate Element**

DiscussionForumTemplate is an optional element of ProductConfig and is ignored for product templates.

**Note:** This element is not ignored when used within ProjectConfig elements.

### **ExportedRoleMemberMap Element**

ExportedRoleMemberMap is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common ExportedRoleMemberMap Element](#).

### **ExportedGuestMembers Element**

ExportedGuestMembers is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common ExportedGuestMembers Element](#).

### **ExportedTemplateFiltering Element**

ExportedTemplateFiltering is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common ExportedTemplateFiltering Element](#).

### **AccessControlRule Element**

AccessControlRule is an optional element of ProductConfig that defines access control policy rules and is used within elements other than ProductConfig.

For the details on this element, see [Common AccessControlRule Element](#).



## **TypeBasedRule Element**

TypeBasedRule is an optional element of ProductConfig that defines default values for objects. The element is also used within elements other than ProductConfig.

For the details on this element, see [Common TypeBasedRule Element](#).

## **NotebookTemplate Element**

NotebookTemplate is an optional element of ProductConfig and defines notebook templates. The element is also used within elements other than ProductConfig.

For the details on this element, see [Common NotebookTemplate Element](#).

## **AdHocACLEntrySet Element**

AdHocACLEntrySet is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common AdHocACLEntrySet Element](#).

## **PreferenceInstance Element**

PreferenceInstance is an optional element of ProductConfig that is also used within elements other than ProductConfig.

For the details on this element, see [Common PreferenceInstance Element](#).

## Creating Business XML Files for Library Templates

The content of the business XML file used as input when creating a library template consists of a single, complex XML element named LibraryConfig. This element is defined by the following DTD element:

```
<!ELEMENT LibraryConfig (SharedTeamDef?,
                          projectMember*,
                          OrgStructure?,
                          DiscussionForumTemplate?,
                          ExportedRoleMemberMap?,
                          ExportedGuestMembers?,
                          ExportedTemplateFiltering?,
                          TeamTemplate*,
                          NmLoader*,
                          AccessControlRule*,
                          IndexPolicyRule*,
                          TypeBasedRule*,
                          TemplateDocument*,
                          DocumentTemplate*,
                          NotebookTemplate?,
                          AdHocACLEntrySet?,
                          PreferenceInstance*)>
```

The following subelements contained in the LibraryConfig element are not supported:

- projectMember -- preferred method of defining roles is using the ExportedRoleMemberMap element (described in [Common ExportedRoleMemberMap Element](#))
- TeamTemplate
- NmLoader
- IndexPolicyRule
- TemplateDocument
- DocumentTemplate

PTC recommends that you export an existing library as a template, modify the business XML file (and associated files), and then import the template. For import details, see [Required Contents of ZIP File Used for Importing a Context Template](#) in the Context Templates chapter.

**Tip:** When modifying the XML file, include comments to document your changes and use indentation to indicate the hierarchical nature of the XML structure. Also use an XML editor that can validate your XML document against the DTD.

The supported subelements of LibraryConfig are described in the following sections.

## SharedTeamDef Element

SharedTeamDef is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common SharedTeamDef Element](#).

## OrgStructure Element

OrgStructure is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common OrgStructure Element](#).

## DiscussionForumTemplate Element

DiscussionForumTemplate is an optional element of LibraryConfig and is ignored for library templates.

**Note:** This element is not ignored when used within ProjectConfig elements.

## ExportedRoleMemberMap Element

ExportedRoleMemberMap is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common ExportedRoleMemberMap Element](#).

## ExportedGuestMembers Element

ExportedGuestMembers is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common ExportedGuestMembers Element](#).

## ExportedTemplateFiltering Element

ExportedTemplateFiltering is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common ExportedTemplateFiltering Element](#).

## AccessControlRule Element

AccessControlRule is an optional element of LibraryConfig that defines access control policy rules and is used within elements other than LibraryConfig.

For the details on this element, see [Common AccessControlRule Element](#).

## **TypeBasedRule Element**

TypeBasedRule is an optional element of LibraryConfig that defines default values for objects. The element is also used within elements other than LibraryConfig.

For the details on this element, see [Common TypeBasedRule Element](#).

## **NotebookTemplate Element**

NotebookTemplate is an optional element of LibraryConfig and defines notebook templates. The element is also used within elements other than LibraryConfig.

For the details on this element, see [Common NotebookTemplate Element](#).

## **AdHocACLEntrySet Element**

AdHocACLEntrySet is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common AdHocACLEntrySet Element](#).

## **PreferenceInstance Element**

PreferenceInstance is an optional element of LibraryConfig that is also used within elements other than LibraryConfig.

For the details on this element, see [Common PreferenceInstance Element](#)

# Creating Business XML Files for Program and Project Templates

The content of the business XML file used as input when creating a program or project template consists of a single, complex XML element named ProjectConfig. This element is defined by the following DTD element:

```
<!ELEMENT ProjectConfig (BaseProjectData?,
    SharedTeamDef?,
    (ExportedRoleMemberMap | projectMember*)?,
    ExportedGuestMembers?,
    ExportedTemplateFiltering?,
    OrgStructure?,
    NmLoader*,
    (TypeBasedRule | SeedObjectRuleSet)*,
    createFromRule*,
    AccessControlRule*,
    ProductStructure?,
    DocumentTemplate*,
    TemplateDocument*,
    sandboxBaseline?,
    shareMapSet?,
    projectPlan*,
    ProjectEnumType*,
    DiscussionForumTemplate*,
    ForumSpec*,
    NotebookTemplate?,
    NotebookSpec?,
    AdHocACLEntrySet?,
    PreferenceInstance*)>
```

The following subelements contained in the ProjectConfig element are not supported:

- BaseProjectData
- projectMember -- preferred method of defining roles is using the ExportedRoleMemberMap element (described in [Common ExportedRoleMemberMap Element](#))
- NmLoader
- SeedObjectRuleSet-- superseded by the TypeBasedRule element
- createFromRule -- superseded by the TypeBasedRule element
- DocumentTemplate
- sandboxBaseline
- TemplateDocument
- ForumSpec
- NotebookSpec

PTC recommends that you export an existing project or program as a template, modify the business XML file (and associated files), and then import the template. For import details, see [Required Contents of ZIP File Used for Importing a Context Template](#) in the Context Templates chapter.

**Tip:** When modifying the XML file, include comments to document your changes and use indentation to indicate the hierarchical nature of the XML structure. Also use an XML editor that can validate your XML document against the DTD.

The supported subelements of ProjectConfig are described in the following sections.

## **SharedTeamDef Element**

SharedTeamDef is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common SharedTeamDef Element](#).

## **ExportedRoleMemberMap Element**

ExportedRoleMemberMap is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common ExportedRoleMemberMap Element](#).

## **ExportedGuestMembers Element**

ExportedGuestMembers is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common ExportedGuestMembers Element](#).

## **ExportedTemplateFiltering Element**

ExportedTemplateFiltering is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common ExportedTemplateFiltering Element](#).

## **OrgStructure Element**

OrgStructure is an optional element of ProjectConfig that is used for defining domains, folders, user-defined groups, and access control policy rules. The element is also used within elements other than ProjectConfig.

For the details on this element, see [Common OrgStructure Element](#).

## TypeBasedRule Element

TypeBasedRule is an optional element of ProjectConfig that defines default values for objects. The element is also used within elements other than ProjectConfig.

For the details on this element, see [Common TypeBasedRule Element](#).

## AccessControlRule Element

AccessControlRule is an optional element of ProjectConfig that defines access control policy rules and is also used within elements other than ProjectConfig.

For the details on this element, see [Common AccessControlRule Element](#).

## ProductStructure Element

ProductStructure element is an optional element of ProjectConfig that contains all of the information required to define parts and documents within a product.

The ProductStructure element is defined by the following DTD element:

```
<!ELEMENT ProductStructure ((WTypeDefinitions | ibaDefinitions
| EPMDocument | EPMBuildLinksRule | EPMContainedIn
| EPMDescribeLink | EPMMemberLink | EPMReferenceLink
| EPMVariantLink | EPMBuildHistory
| WTPart | WTPartReferenceLink | WTPartUsageLink
| WTPartDescribeLink)*)>
```

The following subelements contained in the ProductStructure element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported:

- WTypeDefinitions -- for information about defining data types, see the [Type and Attribute Manager](#) chapter
- ibaDefinitions

Many of the supported subelements referenced in the ProductStructure element are complex and are described in the following sections.

## EPMDocument Element

The EPMDocument element contains information that defines a CAD document and each CAD document represents a CAD model. The CAD model can be a component or an assembly. CAD document can contain a number of data files, URLs, or both data files and URLs. Therefore, a CAD document can store alternate representations of the CAD model. The primary content of the CAD document is usually the model file.

The EPMDocument element is defined by the following DTD elements:

```
<!ELEMENT EPMDocument (dtd?,
    ObjectID,
    objectContainerPath?,
    objectContainerOid?,
    ownerApplication?,
    authoringApplication,
    organizationId?,
    number,
    masterUfid?,
    masterExternalTypeId?,
    masterTypeOid?,
    masterIba*,
    name,
    CADName?,
    epmDocType,
    epmDocSubType?,
    defaultUnit?,
    description?,
    authoringApplicationVersion?,
    dbKeySize?,
    isVerified?,
    revisionNumber?,
    familyTableStatus?,
    epmCADReferenceControl?,
    extentsValid?,
    epmBoxExtents?,
    domainName?,
    folderPath,
    folderOid?,
    versionInfo?,
    lifecycleInfo?,
    teamIdentity?,
    contentItem*,
    externalTypeId?,
    typeOid?,
    iba*,
    excludedIba*,
    EPMPParameterMap*,
    isMissingDependents?,
    EPMFeatureValue*,
    EPMPParameterValue*,
    archiveInfo?)>
```

The following subelements contained in the EPMDocument element are not supported:

- objectContainerPath
- objectContainerOid
- organizationId
- domainName
- folderPath



- folderOid
- versionInfo
- archiveInfo

The following subelements contained in the EMPDocument element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported:

- ObjectID
- externalTypeId
- typeOid
- masterExternalTypeId
- masterTypeOid
- masterUfid
- masterIba
- iba
- excludedIba
- ownerApplication
- authoringApplication
- authoringApplicationVersion
- epmDocType
- epmDocSubType
- defaultUnit
- familyTableStatus
- isMissingDependents
- isVerified
- extentsValid
- epmBoxExtents
- EPMPParameterMap
- EPMFeatureValue
- EPMPParameterValue
- dbKeySize
- revisionNumber

- epmCADReferenceControl
- LifecycleInfo
- teamIdentity
- contentItem
- TypeBasedRule Element

**Note:** Many of the values of these subelements are derived from the model file. As a result, changing them leads to data corruption and causes discrepancy between the model file and the information in Windchill.

The following table provides additional information about the customizable subelements in the DTD elements for EMPDocument:

Subelement	Description
number	Number of the EPMDocument.
name	Name of the EPMDocument.
CADName	<p>This element is the name of the CAD file. Even though this element is optional in the DTD, CAD name is required.</p> <p>This element can be changed if the document is authored by Pro/ENGINEER. That is, the authoringApplication on this document should be PROE.</p> <p><b>Note:</b> If this document is a top generic, you must also change the name on the EPMSepFamilyTable. CAD name of the top generic and the name of the internal family table object must be the same; otherwise, the family table will get corrupted and many operations on the family table will not work properly. A document is top generic if the familyStatus on this document is 2.</p>
description	Description of the EPMDocument.

## EPMBuildLinksRule

EPMBuildLinksRule represents the relationship between the build source and the build target. This relationship is required by the build process to determine the source for a target or the target for a source. The build source of the EPMBuildLinksRule is a CAD document. The target is a WTPart object.

**Note:** You cannot associate an EPMBuildRule to an EPMDocument that has an effectivity assigned to it.

The EPMBuildLinksRule element is defined by the following DTD element:

```
<!ELEMENT EPMBuildLinksRule (dtd?,  
                                ObjectID,  
                                (buildSource, buildTarget))>
```

The following table provides additional information about the customizable subelements in the DTD elements for EMPDBuildLinksRule:

Subelement	Description
ObjectID	Object identifier of the object. The identifier is set by the export. For details on ObjectID, see <a href="#">ObjectID Element</a> .
buildSource	This element is the source CAD document.
buildTarget	This element is the target WTPart. The information on the source CAD document is propagated to the target

**Note:** Adding a new build rule to the template is allowed if the following conditions are satisfied:

- The CAD document and WTPart object are in the same template file.
- The CAD document or WTPart object is not associated to another WTPart object or CAD document using a build rule.

## EPMDescribeLink

EPMDescribeLink is a relationship specifying that a CAD document provides descriptive information for a WTPart object.

The EPMDescribeLink element is defined by the following DTD elements:

```
<!ELEMENT EPMDescribeLink (dtd?,  
                           ObjectID,  
                           describedBy,  
                           describes,  
                           isBuiltFlag?)>
```

**Note:** Modifying the information on the link is not supported if isBuiltFlag is true. You can however add a new association between a CAD document and a WTPart to the template. The CAD document and the WTPart must exist in the same template file.

The following table provides additional information about the customizable subelements in the DTD elements for EMPDescribeLink:

Subelement	Description
ObjectID	Object identifier of the object. The identifier is set by the export. For details on ObjectID, see <a href="#">ObjectID Element</a>
describes	This element is the WTPart described by the CAD document.
describedBy	This element is the CAD document describing the WTPart.
isBuiltFlag	This element should be set to false.

## EPMMemberLink

EPMMemberLink describes a uses relationship between a CAD document and a CAD document master. This relationship is derived from the CAD model file.

The EPMMemberLink element is defined by the following DTD element:

```
<!ELEMENT EPMMemberLink (dtd?,  
                          ObjectID,  
                          usedBy,  
                          uses,  
                          externalTypeId?,  
                          typeOid?,  
                          depType?,  
                          asStoredChildName?,  
                          isRequired?,  
                          uniqueLinkId?,  
                          uniqueNDId?,  
                          iba*,  
                          excludedIba*,
```

```

EPMPParameterMap*,
isSuppressed?,
name?,
identifier?,
quantityAmount?,
quantityUnit?,
isPlaced?,
(hasTransform, transform)?,
compNumber?,
compRevNumber?,
compLayerIdx?,
epmCADReferenceControl?,
occurrences*)>

```

**Note:** The subelements contained in the EPMMemberLink element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

The values of many of these subelements are derived from the model file. As a result, changing them leads to data corruption and causes discrepancy between the model file and the information in Windchill.

## EPMReferenceLink

EPMReferenceLink describes a reference relationship between a CAD document and a CAD document master. This relationship is derived from the CAD model file.

The EPMReferenceLink element is defined by the following DTD elements:

```

<!ELEMENT EPMReferenceLink (dtd?,
    ObjectID,
    referencedBy,
    references,
    externalTypeId?,
    typeOid?,
    depType?,
    asStoredChildName?,
    isRequired?,
    uniqueLinkID?,
    uniqueNDId?,
    iba*,
    excludedIba*,
    EPMPParameterMap*)>

```

**Note:** The subelements contained in the EPMReferenceLink element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

The values of many of these subelements are derived from the model file. As a result, changing them leads to data corruption and causes discrepancy between the model file and the information in Windchill.

## WTPart

A part is an item that can be produced or consumed. Parts could be assembled to create other parts. Parts used in Windchill repository have wt.part.WTPart type.

The WTPart element is defined by the following DTD element:

```
<!ELEMENT WTPart (dtd?,  
    ObjectID,  
    objectContainerPath?,  
    objectContainerOid?,  
    organizationId?,  
    ((number, masterUfid?), name, defaultUnit?),  
    partType,  
    partSource,  
    domainName?,  
    folderPath,  
    folderOid?,  
    view?,  
    versionInfo?,  
    lifecycleInfo?,  
    teamIdentity?,  
    externalTypeId?,  
    typeOid?,  
    contentItem*,  
    iba*,  
    excludedIba*,  
    archiveInfo?,  
    markUp?)>
```

**Note:** The subelements contained in the WTPart element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

## WTPartReferenceLink

A WTPart could reference a document. A part should reference a document if the document is not owned by the part or if the document could be revised or iterated independently.

The WTPartReferenceLink element is defined by the following DTD element:

```
<!ELEMENT WTPartReferenceLink (dtd?,  
    ObjectID,  
    (referencedBy, references))>
```

**Note:** The subelements contained in the WTPartReferenceLink element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

## WTPartUsageLink

The WTPartUsageLink is an association defined to be used to build structures. The WTPartUsageLink aggregated quantity could be used to indicate the amount of the component that is being consumed.

The WTPartUsageLink element is defined by the following DTD element:

```
<!ELEMENT WTPartUsageLink (dtd?,  
                           ObjectID,  
                           (usedBy, uses),  
                           externalTypeId?,  
                           typeOid?,  
                           iba*,  
                           excludedIba*,  
                           quantityAmount,  
                           quantityUnit,  
                           lineNumber?,  
                           occurences?,  
                           occurence*)>
```

**Note:** The subelements contained in the WTPartUsageLink element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

## WTPartDescribeLink

A part could be described by WTPartDescribeLink. If a particular version of a part is described by a particular version of a document, you should use this link.

The WTPartDescribeLink element is defined by the following DTD element:

```
<!ELEMENT WTPartDescribeLink (dtd?,  
                              ObjectID,  
                              (describedBy, describes))>
```

**Note:** The subelements contained in the WTPartDescribeLink element can be present in an exported XML file for a project or program, but should not be changed when the exported project or program is imported.

## shareMapSet Element

shareMapSet is an optional element of ProjectConfig that contains all of the share maps to be imported in a project or program. Typically, this element is included when a project or program that has been exported has shared objects. There is a single shareMapSet element in the import file that contains one shareMap subelement for each shared object.

**Note:** To obtain the same set of shared objects in an imported project or program template, use the shareMapSet element that was exported without making any manual changes.

The shareMapSet element can occur one time and is defined by the following DTD elements:

```
<!ELEMENT shareMapSet (shareMap*)>

<!ELEMENT shareMap (ObjectID,
                    shareSpec?,
                    shareTarget,
                    folderPath?)>

<!ELEMENT shareSpec (shareType, shareKey)>
<!ELEMENT shareType (#PCDATA)>

<!ELEMENT shareKey (#PCDATA)>

<!ELEMENT shareTarget (ObjectReference, iterationId?)>
<!ATTLIST shareTarget isVersion (true | false) "true">
```

The following table provides additional information about some of the subelements in the DTD elements for shareMapSet:

Subelement	Description
shareMap	Individual specification of a shared object.
ObjectID	Object identifier of the object (folder or single object) to share. The identifier is set by the export. For details on ObjectID, see <a href="#">ObjectID Element</a> .
folderPath	The folder specification of the target folder in which the imported objects are shared.
shareType	Valid values are: FOLDER and SINGLE
shareKey	String representation of long number (like an ID). Share key is 0 for SINGLE shares.
ObjectReference	The object reference to the object being shared.
iterationId	The version information when the object is a version share (isVersion is true); omitted when isVersion is false.
isVersion	Determines whether the share is a version or object share. True indicates version share; false indicates object share.



## projectPlan Element

projectPlan is an optional element of ProjectConfig that contains all of the information required to create a project or program plan in a specific project or program template.

```
<!ELEMENT projectPlan ((manager | managerInfo)*,
                        subscriptionPolicy?,
                        planMode?,
                        creator?,
                        projectId?,
                        projectPlanId,
                        executionInfo,
                        (Activity | Milestone | Summary |
                         SubProject)*,
                        dependencyLink*,
                        containmentLink*,
                        Deliverable*,
                        Resource*,
                        resourceLink*,
                        duration?,
                        projectVariable*)>
```

The following subelements contained in the projectPlan element are not supported:

- manager -- not used when template is imported.
- managerInfo -- not used when template is imported.
- creator -- not used from within a project or program template; the user selecting the template is the creator of the project or program.
- projectID -- not used from within a project or program template; the context that the plan is being imported into is used as the plan's project or program context.
- SubProject
- projectVariable

Many of the supported subelements referenced in the projectPlan element are complex and are described in the following sections.

## ObjectID Element

The ObjectID element is used as a subelement to identify an object.

The ObjectID element is defined by the following DTD elements:

```
<!ELEMENT ObjectID (localId?, objectId?, ufid?)>
<!ELEMENT objectId (#PCDATA)>

<!ELEMENT localId (#PCDATA)>

<!ELEMENT ufid (#PCDATA)>
```

The subelements are described in the following sections.

### localId Element

The localId element refers to an object's local identifier. Local identifiers are only valid during the import process.

The content for the localId element can be any string. The only requirement is that the localId for each object defined in the XML file is unique.

### objectId Element

The objectId element refers to an object's Windchill object ID. The Windchill object ID is valid for the entire life of the object but is only valid on one Windchill system. The object does not need to be defined in the same XML file. Notice that this element is different from parent ObjectID element in that the "o" and the "d" are lower case characters.

**Note:** Using objectId for plan items is not currently supported.

The content of the objectId element must correspond to a Windchill object identifier. The format is as follows:

```
<RT>:<classname>:<OID>
```

Where <RT> is the reference type. The reference type is one of the following:

OR - object reference

VR - version reference

<classname> is the fully-qualified Java class name for the object. For example, wt.projmgmt.execution.SummaryActivity.

<OID> is the identifier for the object. This is the value found in the ida2a2 column in the object's database table.

The following is an example of a Windchill object identifier for a summary activity:

```
OR:wt.projmgmt.execution.SummaryActivity:12612
```

An object can be identified by both a localId and an objectId. But, if localId is specified then the objectId element is ignored.

### ufid Element

The ufid element refers to an object's Unique Federation Identifier (UFID). The UFID contains the distinguished name of the object. This ID is valid for the entire life of the object and is valid across a set of federated Windchill systems. Using this ID allows an object to reference an object in another federated Windchill system.

**Note:** Using ufid element for plan items is not currently supported.

## subscriptionPolicy Element

Owner and manager subscriptions can be created for project or program plans, activities, milestones, summary activities, sub-projects and deliverables. Use the subscriptionPolicy element to specify the events for which project or program managers and object owners should receive notifications.

The subscriptionPolicy element is defined by the following DTD element:

```
<!ELEMENT subscriptionPolicy (ManagerSubscriptionPolicy?,  
                                OwnerSubscriptionPolicy?)>
```

The following sections describe the subelements of subscriptionPolicy.

### ManagerSubscriptionPolicy Element

Use the ManagerSubscriptionPolicy element to create subscriptions for the project or program manager role.

The ManagerSubscriptionPolicy element is defined by the following DTD elements:

```
<!ELEMENT ManagerSubscriptionPolicy (EventSubscription*)>  
  
<!--ATTLIST ManagerSubscriptionPolicy  
      statusRed (true | false) "false"  
      onCompletion (true | false) "false"  
>  
  
<!ELEMENT EventSubscription EMPTY>  
  
<!--ATTLIST EventSubscription  
      event (DEADLINE | DEADLINE_CHANGE | FINISH_CHANGE |  
             OBJECT_DELETION | OWNER_CHANGE | PERCENT_CHANGE |  
             STATE_CHANGE | STATUS_CHANGE | TIME_PAST_DEADLINE |  
             TIME_TO_DEADLINE | RISK_CHANGE | NODE_CREATION)  
      #REQUIRED  
      subscribe (true | false) #REQUIRED  
>
```

You subscribe to most events using an EventSubscription element for each event. Additionally, you subscribe to two special project or program manager events (Turns Red and Completion) by setting the values of attributes:

- Set statusRed attribute equal to true to subscribe to the Status Turns Red event.
- Set onCompletion attribute equal to true to subscribe to the On Completion event.

### OwnerSubscriptionPolicy Element

Use the OwnerSubscriptionPolicy element to create subscriptions for the object owner role. These events are subscribed to using an EventSubscription element.

The OwnerSubscriptionPolicy element is defined by the following DTD elements:

```
<!ELEMENT OwnerSubscriptionPolicy (EventSubscription*)>
```

```

<!ELEMENT EventSubscription EMPTY>

<!--ATTLIST EventSubscription
    event (DEADLINE | DEADLINE_CHANGE | FINISH_CHANGE |
           OBJECT_DELETION | OWNER_CHANGE | PERCENT_CHANGE |
           STATE_CHANGE | STATUS_CHANGE | TIME_PAST_DEADLINE |
           TIME_TO_DEADLINE | RISK_CHANGE | NODE_CREATION)
    #REQUIRED
    subscribe (true | false) #REQUIRED
-->

```

Use the EventSubscription element to indicate the events to which the owner role should be subscribed. Use one EventSubscription element for each event.

### subscriptionPolicy Element Example

The following example shows a set of events for manager and owner subscription using the subscriptionPolicy element:

```

<subscriptionPolicy>
  <ManagerSubscriptionPolicy onCompletion="true" statusRed="true">
    <EventSubscription event="DEADLINE" subscribe="true"></EventSubscription>
    <EventSubscription event="DEADLINE_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="FINISH_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="NODE_CREATION" subscribe="false"></EventSubscription>
    <EventSubscription event="OBJECT_DELETION" subscribe="true"></EventSubscription>
    <EventSubscription event="OWNER_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="PERCENT_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="RISK_CHANGE" subscribe="false"></EventSubscription>
    <EventSubscription event="STATE_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="STATUS_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="TIME_PAST_DEADLINE"
subscribe="true"></EventSubscription>
    <EventSubscription event="TIME_TO_DEADLINE"
subscribe="true"></EventSubscription>
  </ManagerSubscriptionPolicy>
  <OwnerSubscriptionPolicy>
    <EventSubscription event="DEADLINE" subscribe="true"></EventSubscription>
    <EventSubscription event="DEADLINE_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="FINISH_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="NODE_CREATION" subscribe="false"></EventSubscription>
    <EventSubscription event="OBJECT_DELETION" subscribe="true"></EventSubscription>
    <EventSubscription event="OWNER_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="PERCENT_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="RISK_CHANGE" subscribe="false"></EventSubscription>
    <EventSubscription event="STATE_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="STATUS_CHANGE" subscribe="true"></EventSubscription>
    <EventSubscription event="TIME_PAST_DEADLINE"
subscribe="true"></EventSubscription>
    <EventSubscription event="TIME_TO_DEADLINE"
subscribe="true"></EventSubscription>
  </OwnerSubscriptionPolicy>
</subscriptionPolicy>

```

## planMode Element

The planMode element is used to specify the plan execution control.

The planMode element is defined by the following DTD elements:

```
<!ELEMENT planMode EMPTY>

<!--ATTLIST planMode
      mode (staticPlan | activityDriven | taskDriven) "staticPlan"
-->
```

The following table provides additional information about the modes:

Mode	Description
staticPlan	Allows manual control of project or program execution
activityDriven	Automatically executes activities (without task delivery)
taskDriven	Automatically executes activities and deliver tasks

## projectPlanId Element

The projectPlanId element provides an identifier for a project or program plan and indicates the plan to which a plan node belongs. This element is required for all plan nodes.

```
<!ELEMENT projectPlanId (ObjectID)>
```

During an import, the value of this element is only used if the context that the data is being imported into is not a project or program context. Otherwise, the plan for the project or program context being imported into is used.

For details on ObjectID, see [ObjectID Element](#).

## executionInfo Element

The executionInfo element is used to specify many of the attributes that are shared by project or program plan objects. This element is used when defining plans, activities, milestones, summary activities, sub-projects, and deliverables.

The executionInfo element is defined by the following DTD elements:

```
<!ELEMENT executionInfo (name,
                          healthStatus?,
                          statusDescription?,
                          riskValue?,
                          projectCategory?,
                          projectState?,
                          projectPhase?,
                          executionDescription?,
                          percentComplete?,
                          riskDescription?,
```

```

        startDate?,
        finishDate?,
        timeToStart?,
        estimatedFinish?,
        deadline?,
        inheritedDeadline?,
        projectConfiguration?,
        workflowConfiguration?,
        timeToDeadline?,
        timePastDeadline?,
        totalWork?,
        performedWork?,
        totalWorkCost?,
        currentWorkCost?,
        priority?,
        responsibleRole?)>

<!--ATTLIST executionInfo
    enabled (true | false) "true"
-->

```

In some cases, a project or program plan object is disabled rather than being deleted. The enabled attribute on the executionInfo element is used to specify whether the object is enabled or disabled. This attribute should always be set to true when working with a template and the default is true. Therefore, you can omit this element for project or program templates.

The following subelements contained in the executionInfo element are not supported for project or program templates:

- projectCategory
- projectState
- projectPhase
- percentComplete
- startDate
- finishDate
- timeToStart
- estimatedFinish
- inheritedDeadline
- projectConfiguration
- workflowConfiguration
- totalWork
- performedWork
- totalWorkCost

- currentWorkCost
- priority
- responsibleRole

The following sections describe the supported subelements of executionInfo.

### name Element

The name element is used to name activities, summary activities, milestones, sub-projects, and deliverables.

**Note:** This element is not supported for project or program plans; the name of the project or program plan is entered when the plan is created.

The name element is defined by the following DTD element:

```
<!ELEMENT name (#PCDATA)>
```

### healthStatus Element

The healthStatus element is used to state the health status for activities and deliverables. It is also used to state the health status for milestones when no deliverables are included.

**Note:** This element is not supported for project or program plans or summary activities; the health status is rolled up from child nodes for these objects. The element is also not supported for milestones when there is an associated delivery; the health status is rolled up from its deliverables.

The healthStatus element is defined by the following DTD elements:

```
<!ELEMENT healthStatus (#PCDATA)>

<!ATTLIST healthStatus
    status (GREEN | YELLOW | RED | UNAVAILABLE | CUSTOM)
    #REQUIRED
>
```

The CUSTOM health status value is not supported.

### statusDescription Element

The statusDescription element describes the status of the project or program plan.

The statusDescription element is defined by the following DTD element:

```
<!ELEMENT statusDescription (#PCDATA)>
```

## riskValue Element

The riskValue element specifies the risk value of the project or program plan object.

The riskValue element is defined by the following DTD elements:

```
<!ELEMENT riskValue (#PCDATA)>

<!ATTLIST riskValue
    value (VERYLOW | LOW | MEDIUM | HIGH | VERYHIGH | UNAVAILABLE
    | CUSTOM) #REQUIRED
>
```

The content of the element is normally empty. The risk value is specified using the riskValue attribute. The CUSTOM risk value is not supported.

This element is supported when importing milestones, summaries, activities, and deliverables; it is not supported when importing project or program plans.

## riskDescription Element

The riskDescription element specifies the value of the object's risk description attribute.

The riskDescription element is defined by the following DTD elements:

```
<!ELEMENT riskDescription (#PCDATA)>
```

This element is supported when importing milestones, summaries, activities, and deliverables; it is not supported when importing project or program plans.

## executionDescription Element

The executionDescription element is used to describe activities, summary activities, milestones, sub-projects, and deliverables.

**Note:** This element is not supported for project or program plans; the description of the project or program plan is displayed in the user interface.

The executionDescription element is defined by the following DTD element:

```
<!ELEMENT executionDescription (#PCDATA)>
```



## Elements for the Project Plan Schedule

The following group of elements provide schedule information about a project or program plan object:

The scheduling elements contained in the executionInfo element are defined by the following DTD elements:

```
<!ELEMENT deadline (#PCDATA)>

<!ELEMENT timeToDeadline (#PCDATA)>
<!ATTLIST timeToDeadline
    unit (day | hour | minute | second | millisecond) #IMPLIED
>
<!ELEMENT timePastDeadline (#PCDATA)>
<!ATTLIST timePastDeadline
    unit (day | hour | second | millisecond) #IMPLIED
>
```

The following table provides additional information about the supported subelements used in providing schedule information about a project or program plan:

Subelement	Description
deadline	Specifies the deadline for a project or program plan or plan node. The format required in this element is described after this table.
timeToDeadline	<p>Specifies an integer representing the amount of time before the deadline that the Days Before Deadline notifications should be sent. The integer represents either days, hours, seconds, or milliseconds (depending on the value of the unit attribute).</p> <p>This element is only applicable if project or program manager or object owner subscriptions have been created for the Days Before Deadline event.</p>
timePastDeadline	<p>Specifies an integer representing the amount of time after the deadline that the Days After Deadline notifications should be sent. The integer represents either days, hours, seconds, or milliseconds (depending on the value of the unit attribute).</p> <p>This element is only applicable if project or program manager or object owner subscriptions have been created for the Days After Deadline event.</p>

The format for the deadline element is as follows:

YYYY-MM-DD HH:MM:SS.NNN

Where:

- YYYY is the four-digit year
- MM is a two-digit number between 01 and 12 representing the month
- DD is a two-digit number between 01 and 31 representing the day of the month
- HH is a two-digit number between 00 and 23 representing the hour of the day
- MM is a two-digit number between 00 and 59 representing the minutes within the hour
- SS is a two-digit number between 00 and 59 representing the seconds within the minute
- NNN is a one-, two- or three-digit number representing the fraction of a second

The time zone is assumed to be GMT.

For example, use:

2005-04-18 04:00:00.0

For April 18, 2005 04:00 AM GMT

The convention used by the project or program management module is to align dates with day boundaries. A date representing the beginning of a time period (estimated start time and actual start time) is aligned with midnight on that day in the user's time zone. For example, if the user who has a preferred time zone is Central (United States) set the start date to April 18, 2005, the date would be exported as follows:

2005-04-18 05:00:00.0

This date represents April 18, 2005 5:00 AM CDT.

Dates that represent the end of a time period (estimated finish date, actual finish date and deadline) are aligned with midnight on the day following that day in the user's time zone. For example, if the user referred to above set the deadline to April 25, 2005, the date would be exported as follows:

2005-04-26 05:00:00.0

This date represents April 26, 2005 5:00 AM CDT.

## Activity, Milestone, and Summary Elements

As part of a project or program plan, you can specify one of the following subelements:

- Activity - defines activities in a project or program plan.
- Milestone - defines milestones in a project or program plan.
- Summary - defines summary activities in a project or program plan.

The Activity, Milestone, and Summary elements are defined by the following DTD elements:

```
<!ELEMENT Activity (ObjectID,
                    projectPlanId,
                    (owner | ownerInfo)?,
                    subscriptionPolicy?,
                    activityInfo,
                    resourceAssignee*,
                    duration?,
                    projectVariable*)>

<!ELEMENT Milestone (ObjectID,
                    subscriptionPolicy?,
                    projectPlanId,
                    (owner | ownerInfo)?,
                    activityInfo,
                    duration?,
                    projectVariable*)>

<!ELEMENT Summary (ObjectID,
                    projectPlanId,
                    (owner | ownerInfo)?,
                    subscriptionPolicy?,
                    activityInfo,
                    duration?,
                    projectVariable*)>

<!ELEMENT activityInfo (executionInfo, nodeNumber?)>
<!ELEMENT resourceAssignee (ObjectID, maxPercentAvailable?)>
<!ELEMENT ownerInfo (name, ObjectReference?)>
<!ELEMENT duration (#PCDATA)>
<!ATTLIST duration
    unit (day | hour | second | millisecond) #IMPLIED
>
```

The following subelements are not supported:

- owner
- maxPercentAvailable
- projectVariable

The following table provides additional information about the subelements that are used in one or more of the Activity, Milestone, Summary, and SubProject elements:

Subelement	Description
ObjectID	ID of the object being defined. The ID can be used to reference this object in other places in the XML file. For example, this ID can be used when defining a dependencyLink between two activities. For details on ObjectID, see <a href="#">ObjectID Element</a> .
projectPlanId	Specifies the identifier for a project or program plan. See <a href="#">projectPlanId Element</a> .
ownerInfo	Specifies the owner of the deliverable. Set the content of the name element (a child element of the ownerInfo element) to the user name or e-mail address of the user.  The user that is specified as an owner must be a member of the project or program. This can be accomplished by using the ExportedRoleMemberMap element to add the user to the project or program team as part of the XML import or by adding the user to the team prior to importing the XML file.
subscriptionPolicy	Specifies owner and manager subscriptions created for the activity, milestone, summary, or subproject. See <a href="#">subscriptionPolicy Element</a> .
activityInfo	Specifies information for activities, summary activities, milestones, and sub-projects. It holds the executionInfo and nodeNumber elements
duration	Specifies the duration of the activity, milestone, summary, or subproject. See <a href="#">duration Element</a> .
executionInfo	Attributes of the activity, milestone, summary, or subproject. See <a href="#">executionInfo Element</a> .

Subelement	Description
nodeNumber	Allows the ID of the plan node to be specified. The ID is the number that is displayed in the left-most column of on the plan page. The ID is used to determine the order in which plan nodes are listed in the hierarchical plan view.  The IDs must be consistent with the hierarchical structure of the plan. For example, the IDs for the children of a summary activity must immediately follow the ID for the summary activity itself.
resourceAssignee	Allows resources to be assigned to an activity.

## dependencyLink Element

The dependencyLink element is used to define a predecessor constraint.

The dependencyLink element is defined by the following DTD elements:

```
<!ELEMENT dependencyLink (predecessorID,
                           successorID,
                           nodeLinkType?)>
<!ELEMENT predecessorID (ObjectID)>
<!ELEMENT successorID (ObjectID)>
<!ELEMENT nodeLinkType (customType?)>
<!--
  linkType (FINISH_START | FINISH_FINISH | START_FINISH |
  START_START | CUSTOM) #REQUIRED
-->
```

The following table provides additional information about the dependencyLink subelements:

Subelement	Description
predecessorID	ID of the predecessor node. For details on ObjectID, see <a href="#">ObjectID Element</a> .
successorID	ID of the successor node. For details on ObjectID, see <a href="#">ObjectID Element</a> .
nodeLinkType	Specifies the constraint type. Currently only FINISH_START is supported. This constraint type specifies that the successor can start as soon as the predecessor has finished.

**Note:** Cyclical dependency links are not allowed. This means that the successorID and the predecessorID must refer to distinct objects and the it must not be possible to traverse a series of dependency links and encounter the same node twice.

The following example shows the use of dependencyLink element:

```
<dependencyLink>
  <predecessorID><ObjectID>
    <localId>wt.projmgmt.execution.Milestone:105878</localId></ObjectID>
  </predecessorID>
  <successorID><ObjectID>
    <localId>wt.projmgmt.execution.SummaryActivity:106075</localId></ObjectID>
  </successorID>
  <nodeLinkType linkType="FINISH_START"></nodeLinkType>
</dependencyLink>
```

## containmentLink Element

The containmentLink element is used to define parent child relationships between certain plan objects and summary activities. Only milestones, activities, summary activities, and sub-projects can be contained in a summary activity. Deliverables cannot be contained in a summary activity.

The containmentLink element is defined by the following DTD elements:

```
<!ELEMENT containmentLink (summaryID, nodeID*)>
<!ELEMENT summaryID (ObjectID)>
<!ELEMENT nodeID (ObjectID)>
```

The following table provides additional information about the containmentLink subelements:

Subelement	Description
summaryID	ID of the summary activity. For details on ObjectID, see <a href="#">ObjectID Element</a> .
nodeID	ID of the child node. The child node must be a milestone, activity, summary activity, or subproject. For details on ObjectID, see <a href="#">ObjectID Element</a> .

**Note:** Cyclical containment links are not allowed. This means that the summaryID and the nodeID must refer to distinct objects and the it must not be possible to traverse a series of containment links and encounter the same node twice.

The following example shows the use of containmentLink element:

```
<containmentLink>
  <summaryID>
    <ObjectID>
      <localId>wt.projmgmt.execution.SummaryActivity:106075</localId>
    </ObjectID>
  </summaryID>
  <nodeID>
    <ObjectID>
      <localId>wt.projmgmt.execution.ProjectActivity:106103</localId>
    </ObjectID>
  </nodeID>
</containmentLink>
```

## Deliverable Element

The Deliverable element defines a deliverable to include in a project or program plan.

The Deliverable element is defined by the following DTD elements:

```
<!ELEMENT Deliverable (ObjectID?,
                        subscriptionPolicy?,
                        projectPlanId?,
                        activityID?,
                        mileStoneId*,
                        (owner | ownerInfo)?,
                        executionInfo,
                        targetConfig?,
                        targetReference?,
                        duration?)>

<!ATTLIST Deliverable
  orphan (true | false) #IMPLIED
>

<!ELEMENT activityID (ObjectID)>

<!ELEMENT mileStoneId (ObjectID)>

<!ELEMENT owner (#PCDATA)>

<!ELEMENT ownerInfo (name, ObjectReference?)>

<!ELEMENT targetReference (ObjectID | createFromRule)>

<!ELEMENT duration (#PCDATA)>
<!ATTLIST duration
  unit (day | hour | second | millisecond) #IMPLIED
>
```

**Note:** If a user selects Deliverables when exporting a project or program template, all deliverables are exported as orphans.

If a deliverable is an orphan deliverable, then the following attributes are ignored:

projectPlanId  
activityID  
mileStoneId

The following subelements are not supported:

- owner
- orphan
- ObjectReference
- targetConfig
- createFromRule

The following table provides additional information about the supported Deliverable subelements:

Subelement	Description
ObjectID	ID of the deliverable. For details on ObjectID, see <a href="#">ObjectID Element</a> .
subscriptionPolicy	Specifies owner and manager subscriptions created for the deliverable. See <a href="#">subscriptionPolicy Element</a> .
projectPlanId	Specifies the identifier for a project or program plan. See <a href="#">projectPlanId Element</a> .
activityID	<p>Specifies the ID of the activity. Allows the deliverable to be associated with an activity.</p> <p>If you have specified link deliverables in your project or program configuration, then you ensure only one deliverable is associated with an activity.</p> <p>For details on ObjectID, see <a href="#">ObjectID Element</a>.</p>
mileStoneId	Specifies the ID of the milestone. This allows the deliverable to be associated with one or more milestones. One element is created per milestone reference. For details on ObjectID, see <a href="#">ObjectID Element</a> .



Subelement	Description
ownerInfo	Specifies the owner of the deliverable. Set the content of the name element (a child element of the ownerInfo element) to the user name or e-mail address of the user.  The user that is specified as an owner must be a member of the project or program. This can be accomplished by using the ExportedRoleMemberMap element to add the user to the project or program team as part of the XML import or by adding the user to the team prior to importing the XML file.
executionInfo	Attributes of the deliverable. See <a href="#">executionInfo Element</a> .
targetReference	Specifies the ID of the reference. This allows the deliverable to be associated with a target document or part. The ObjectID element is used to provide a reference to the target document or part. For details on ObjectID, see <a href="#">ObjectID Element</a> .
duration	Specifies the duration of the deliverable. See <a href="#">duration Element</a> .

## Resource Element

The Resource element defines the resources of the project or program plan.

The Resource element is defined by the following DTD elements:

```
<!ELEMENT Resource (ObjectID?, projectPlanId, resourceInfo)>

<!ATTLIST Resource
  resourceType (PERSON | ROLE | EQUIPMENT | MATERIAL |
INFORMATION | FACILITY) #REQUIRED
>
```

**Note:** Only PERSON, EQUIPMENT, MATERIAL, and FACILITY are supported resource types.

The following table provides additional information about the supported Resource subelements:

Subelement	Description
ObjectID	ID of the resource. For details on ObjectID, see <a href="#">ObjectID Element</a> .
projectPlanId	Specifies the identifier for a project or program plan. See <a href="#">projectPlanId Element</a> .
resourceInfo	Specifies the details about the resource. See <a href="#">resourceInfo Element</a> .

### resourceInfo Element

The resourceInfo element defines the resource information for a resource.

The resourceInfo element is defined by the following DTD elements:

```

<!ELEMENT resourceInfo (name,
                        description?,
                        resourceCategory?,
                        label?,
                        maxUnits?,
                        resourceReference?,
                        standardRate?,
                        costPerUse?,
                        overtimeRate?,
                        maxPercentUtilization?)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT description (#PCDATA)>

<!ELEMENT maxUnits (#PCDATA)>

<!ELEMENT resourceReference (userInfo | ObjectReference)>

<!ELEMENT userInfo (name | ObjectReference)>

<!ELEMENT standardRate (#PCDATA)>

<!ATTLIST standardRate
    currencyLocale CDATA #IMPLIED
    unit CDATA #IMPLIED
>

<!ELEMENT costPerUse (#PCDATA)>

<!ATTLIST costPerUse
    currencyLocale CDATA #IMPLIED
>

<!ELEMENT overtimeRate (#PCDATA)>

<!ATTLIST overtimeRate
    currencyLocale CDATA #IMPLIED

```

```

        unit CDATA #IMPLIED
    >

    <!ELEMENT maxPercentUtilization (#PCDATA)>

```

The following subelements are not supported:

- resourceCategory
- label
- ObjectReference
- currencyLocale
- unit

The following table provides additional information about some of the supported resourceInfo subelements:

Subelement	Description
maxUnits	Specifies the maximum units for material resources. This element is only valid if the resourceType is equal to MATERIAL and is an integer.
resourceReference	Name of the resource as defined in the userInfo element.  ID of the resource. For details on ObjectID, see <a href="#">ObjectID Element</a> .
standardRate	Specifies a value for the resource's standard rate. The content is a decimal number representing the standard rate. For time resources (person, equipment and facility), this is the standard cost per hour. For material resources, this is the cost per unit.
costPerUse	Specifies the cost that is accrued each time a resource is used. The content is a decimal representing the cost per use.
overtimeRate	Specifies a value for the resource's overtime rate. The content is a decimal number representing the standard rate. For time resources (person, equipment and facility), this is the overtime cost per hour.  This element is not supported for material resources.

Subelement	Description
maxPercentUtilization	Specifies the maximum allocation percentage for time-based resources such as person, equipment, and facility resources. This element is only valid if the resourceType is PERSON, EQUIPMENT, or FACILITY.  The content of this element is an integer representing the maximum percent allocation.

## resourceLink Element

The resourceLink element assigns resources to activities.

The resourceLink element is defined by the following DTD elements:

```
<!ELEMENT resourceLink (resourceHolderID,
                        resourceID,
                        totalResources?)>

<!ELEMENT resourceHolderID (ObjectID)>

<!ELEMENT resourceID (ObjectID)>

<!ELEMENT totalResources (#PCDATA)>
```

The following table provides additional information about the resourceLink subelements:

Subelement	Description
resourceHolderID	ID of the activity. For details on ObjectID, see <a href="#">ObjectID Element</a> .
resourceID	ID of the resource. For details on ObjectID, see <a href="#">ObjectID Element</a> .
totalResources	Specifies how much of the resource is allocated to the activity. For Person, Equipment, and Facility resources this is a percentage allocation. For Material resources this is the number of units allocated to the activity.

The following example shows the use of resourceLink element:

```
<resourceLink>
  <resourceHolderID>
    <ObjectID>
      <localId>wt.projmgmt.execution.ProjectActivity:106103</localId>
    </ObjectID>
  </resourceHolderID>
  <resourceID>
    <ObjectID>
      <localId>wt.projmgmt.resource.PersonResource:106029</localId>
    </ObjectID>
  </resourceID>
  <totalResources>50</totalResources>
</resourceLink>
```

## duration Element

The duration element specifies the duration of a plan object. The duration can be specified in one of four different units: days, hours, seconds, or milliseconds using the unit attribute.

The duration element is defined by the following DTD elements:

```
<!ELEMENT duration (#PCDATA)>
<!ATTLIST duration
  unit (day | hour | second | millisecond) #IMPLIED
>
```

## ProjectEnumType Element

ProjectEnumType is an optional element of ProjectConfig and is used to define the project or program roles for a specific type of project or program. It is also used within elements other than ProjectConfig.

For the details on this element, see [Common ProjectEnumType Element](#).

## DiscussionForumTemplate Element

DiscussionForumTemplate is a element common to Windchill context templates but is only used only within the ProjectConfig element. It defines discussion forum templates for project or program contexts.

The DiscussionForumTemplate element is defined by the following DTD elements:

```
<!ELEMENT DiscussionForumTemplate (name,  
                                   description,  
                                   xmlLobString)>  
  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT description (#PCDATA)>  
<!ELEMENT xmlLobString (#PCDATA)>
```

Forums have their own XML loader. On import, the handler extracts the xmlLobString element and uses the service to instantiate the forum. On export, the xmlLobString element is extracted.

The following example defines the My Forum Template:

```
<DiscussionForumTemplate>  
  <name>My Forum Template</name>  
  <description>Defines my forum topics</description>  
  <xmlLobString>  
    :  
    :  
  </xmlLobString>  
</DiscussionForumTemplate>
```

When used within the DiscussionForumTemplate element, the xmlLobString element can be either a CDATA section or fully-escaped XML that names topics for the discussion forum template being defined. The following xmlLobString example uses fully-escaped XML to name the General, Documents, and Parts topics:

```
<xmlLobString>  
  &lt;Structure&gt;  
    &lt;Topic&gt;  
      &lt;Name&gt;General&lt;/Name&gt;  
    &lt;/Topic&gt;  
    &lt;Topic&gt;  
      &lt;Name&gt;Documents&lt;/Name&gt;  
    &lt;/Topic&gt;  
    &lt;Topic&gt;  
      &lt;Name&gt;Parts&lt;/Name&gt;  
    &lt;/Topic&gt;  
  &lt;/Structure&gt;  
</xmlLobString>
```

**Note:** Instead of fully-escaped XML, you can use a CDATA section as is shown in the [Common NotebookTemplate Element](#) example.

## **NotebookTemplate Element**

NotebookTemplate is a element common to Windchill context templates and defines notebook templates for project or program contexts. The element is also used within elements other than ProjectConfig.

For the details on this element, see [Common NotebookTemplate Element](#).

## **AdHocACLEntrySet Element**

AdHocACLEntrySet is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common AdHocACLEntrySet Element](#).

## **PreferenceInstance Element**

PreferenceInstance is an optional element of ProjectConfig that is also used within elements other than ProjectConfig.

For the details on this element, see [Common PreferenceInstance Element](#).

## Creating Business XML Files for Shared Teams

Shared teams are the roles and corresponding members in the roles that are defined in an organization context. After a shared team has been created, it can be used in many application context teams that are created from within the organization context. For information on shared teams and how they are used in context teams see the [Teams](#) chapter.

The content of the business XML file used as input when importing a shared team consists of a single, complex XML element named SharedTeamDef. This element is defined by the following DTD element:

```
<!ELEMENT SharedTeamDef (sharedTeamRef?,
                          sharedTeamDescription?,
                          sharedTeamOwner?,
                          sharedTeamInvitation?,
                          ExportedRoleMemberMap?,
                          ExportedGuestMembers?,
                          enabled?,
                          extendable?,
                          AccessControlRule*,
                          AdHocACLEntrySet*)>

<!ELEMENT sharedTeamRef (#PCDATA)>

<!ELEMENT sharedTeamDescription (#PCDATA)>

<!ELEMENT sharedTeamOwner (WTPrincipalReference)>

<!ELEMENT sharedTeamInvitation (#PCDATA)>

<!ELEMENT enabled (#PCDATA)>

<!ELEMENT extendable (#PCDATA)>
```

PTC recommends that you export an existing shared team, modify the business XML file (if needed), and then import the shared team. For import details, see [Shared Teams](#).

**Tip:** When modifying the XML file, include comments to document your changes and use indentation to indicate the hierarchical nature of the XML structure. Also use an XML editor that can validate your XML document against the DTD.

The following table provides additional information about the some of the SharedTeamDef subelements:

Subelement	Description
sharedTeamRef	Name of the shared team.
sharedTeamDescription	Description of the shared team.



Subelement	Description
sharedTeamOwner	Owner of the shared team as identified by the <a href="#">Common WTPPrincipalReference Element</a> . If the owner is not specified, the user doing the import becomes the owner.
sharedTeamInvitation	Text to use as the invitation.
enabled	A true value enables the shared team for use. A false value disables the shared team.
extendable	A true value allows the shared team to be extended locally when the team is used in a context team. A false value does not allow the shared team to be extended.

The supported subelements of SharedTeamDef that are complex common elements used in other XML files are described in the following sections.

### ExportedRoleMemberMap Element

ExportedRoleMemberMap is an optional element of SharedTeamDef that is also used within elements other than SharedTeamDef.

For the details on this element, see [Common ExportedRoleMemberMap Element](#).

### ExportedGuestMembers Element

ExportedGuestMembers is an optional element of SharedTeamDef that is also used within elements other than SharedTeamDef.

For the details on this element, see [Common ExportedGuestMembers Element](#).

### AccessControlRule Element

AccessControlRule is an optional element of SharedTeamDef that defines access control policy rules and is used within elements other than SharedTeamDef.

For the details on this element, see [Common AccessControlRule Element](#).

### AdHocACLEntrySet Element

AdHocACLEntrySet is an optional element of SharedTeamDef that is also used within elements other than SharedTeamDef.

For the details on this element, see [Common AdHocACLEntrySet Element](#).

# Common XML Elements Used in Multiple Business XML Files

The following sections describe the XML elements that are used in more than one type of business XML file.

## Common AccessControlRule Element

AccessControlRule is an element common to Windchill context templates that is used to define access control policy rules for the objects within a context.

**Note:** This element replaces the element named ACLRule that was supported in earlier releases.

Access control policy rules are associated with domains. For a detailed explanation of access control policy rules and how they are associated with domains, see the [Access Control](#) chapter.

AccessControlRule is defined by the following DTD elements:

```
<!ELEMENT AccessControlRule (domainName,
                             (logicalId | externalTypeId |persistedType),
                             lifecycleState?,
                             WTPPrincipalReference,
                             ((grantPermissionSet| denyPermissionSet) |
                              (grantPermissionSet,denyPermissionSet)))

<!ELEMENT persistedType (#PCDATA)>

<!ELEMENT grantPermissionSet (AccessPermissionSet)>

<!ELEMENT denyPermissionSet (AccessPermissionSet)>

<!ELEMENT AccessPermissionSet ((permissionField|
                               permissionKey)+)>

<!ELEMENT permissionField EMPTY>

<!ATTLIST permissionField name (ALL|READ|MODIFY|CREATE|DELETE|
                               ADMINISTRATIVE|REVISE|
                               NEW_VIEW_VERSION|
                               CHANGE_PERMISSIONS|DOWNLOAD|
                               MODIFY_CONTENT|CHANGE_DOMAIN|
                               CREATE_BY_MOVE|CHANGE_CONTEXT|
                               SET_STATE|MODIFY_IDENTITY)
                               #REQUIRED>

<!ELEMENT permissionKey EMPTY>

<!ATTLIST permissionKey value (-1|0|1|2|3|4|5|6|7|8|9|10|11|12|
                               13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|
                               34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|
                               55|56|57|58|59|60|61|62|63) #REQUIRED>
```

The following table provides additional information about some of the subelements in the DTD elements for AccessControlRule:

Subelement	Description
domainName	Domain path relative to the context.
logicalId externalTypeId persistedType	<p>The logical identifier of a type, the external type identifier, or persisted type definition.</p> <p>Exported context templates use the external type identifier (shown starting with WCTYPE ). You can use the exported type or use the logical identifier for the type as describe in the <a href="#">Type and Attribute Manager</a> chapter.</p>
lifecycleState	State name; either "ALL" or a key defined in StateRB.rbInfo (wt.lifecycle package). If omitted, defaults to "ALL".
WTPPrincipalReference	Names a user or group for the rule. See <a href="#">Common WTPPrincipalReference Element</a> .
grantPermissionSet denyPermissionSet or both grant and deny sets	In an access control rule, you must include at least one of the sets using the AccessPermissionSet element; you can both a grant set and a deny set.
AccessPermissionSet	<p>Identifies the set of permissions. The set can consist of field names (permissionField) or key values (permissionKey).</p> <p>Unless you are using custom permissions, use the field names from the permissionField element.</p>

**Note:** The domain, principal, and type specified by domainName, WTPPrincipalReference, and logicalId or externalTypeId respectively must exist.

The following illustrates an out-of-box access control policy rule for the WTPart type:

```
<AccessControlRule>
  <domainName>/Default</domainName>
  <externalTypeId>WCTYPE|wt.part.WTPart</externalTypeId>
  <lifecycleState>INWORK</lifecycleState>
  <WTPrincipalReference isInternal="true">
    <groupName>teamMembers</groupName>
    <groupType>teamMembers</groupType>
  </WTPrincipalReference>
  <grantPermissionSet><AccessPermissionSet>
    <permissionField name="MODIFY"></permissionField>
    <permissionField name="CREATE"></permissionField>
    <permissionField name="DELETE"></permissionField>
    <permissionField name="MODIFY_CONTENT"></permissionField>
    <permissionField name="CHANGE_DOMAIN"></permissionField>
    <permissionField name="CREATE_BY_MOVE"></permissionField>
    <permissionField name="CHANGE_CONTEXT"></permissionField>
  </AccessPermissionSet></grantPermissionSet>
</AccessControlRule>
```

## Common AdHocACLEntrySet Element

AdHocACLEntrySet is an element common to Windchill context templates that is used to define ad hoc access control rules for the objects within a context.

**Note:** The name of this element contains the uppercase characters ACL. This element replaces the element named AdHocAclEntrySet that was supported in earlier releases.

AdHocACLEntrySet is defined by the following DTD elements:

```
<!ELEMENT AdHocACLEntrySet (AdHocACLEntry*)>

<!ELEMENT AdHocACLEntry (WTPrincipalReference,
                        AccessPermissionSet,
                        ((ownerField|ownerKey)+, ownerId?)>

<!ELEMENT AccessPermissionSet ((permissionField|
                                permissionKey)+)>

<!ELEMENT permissionField EMPTY>

<!ATTLIST permissionField name (ALL|READ|MODIFY|CREATE|DELETE|
                                ADMINISTRATIVE|REVISE|
                                NEW_VIEW_VERSION|
                                CHANGE_PERMISSIONS|DOWNLOAD|
                                MODIFY_CONTENT|CHANGE_DOMAIN|
                                CREATE_BY_MOVE|CHANGE_CONTEXT|
                                SET_STATE|MODIFY_IDENTITY)
                                #REQUIRED>

<!ELEMENT permissionKey EMPTY>

<!ATTLIST permissionKey value (-1|0|1|2|3|4|5|6|7|8|9|10|11|12|
                                13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|
```

```

34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|
55|56|57|58|59|60|61|62|63) #REQUIRED>

<!ELEMENT ownerField EMPTY>

<!--ATTLIST ownerField name (WNC_LIFECYCLE | WNC_NOTIFY |
WNC_WORK_ITEM|WNC_ACCESS_CONTROL|WNC_TEAM|WNC_MEETING|
WNC_SESSION_ITERATION|WNC_CONTAINER| WNC_SHARING) #REQUIRED>

<!ELEMENT ownerKey EMPTY>

<!--ATTLIST ownerKey value (0|1|2|3|4|5|6|7|8) #REQUIRED>

<!ELEMENT ownerId (#PCDATA)>

```

The following table provides additional information about some of the subelements in the DTD elements for AdHocACLEntrySet:

Subelement	Description
WTPrincipalReference	Names a user or group for the rule. See <a href="#">Common WTPrincipalReference Element</a> .
AccessPermissionSet	Identifies the set of permissions. The set can consist of field names (permissionField) or key values (permissionKey).  Unless you are using custom permissions, use the field names from the permissionField element.
ownerField ownerKey	Identifies the owner of permissions. The owner can be a field name (ownerField) or key value (ownerKey).  Unless you are using custom owners, use the field names from the ownerField element.
ownerId	ownerId is not used in typical import and export scenarios, and should be omitted when creating files for import.  In the circumstances where this element is used, the ownerId is the object identifier of a service that is granting the permission.

The following example grants the Change Permissions permission for the Guest system group:

```
<AdHocACLEntrySet>
  <AdHocACLEntry>
    <WTPrincipalReference isInternal="true">
      <groupName>GUEST</groupName>
      <groupType>GUEST</groupType>
    </WTPrincipalReference>
    <AccessPermissionSet>
      <permissionField name="CHANGE_PERMISSIONS">
        </permissionField>
      </AccessPermissionSet>
      <ownerField name="WNC_ACCESS_CONTROL"></ownerField>
    </AdHocACLEntry>
  </AdHocACLEntrySet>
```

## Common ExportedGuestMembers Element

ExportedGuestMembers is an element common to Windchill context templates that is used to define guests that are in the context team.

ExportedGuestMembers is defined by the following DTD elements:

```
<!ELEMENT ExportedGuestMembers (WTPrincipalReference*)>
```

This element uses the WTPrincipalReference element to identify guests in a specific context template. Use the uid subelement to name the guests.

The following example defines user1 as a guest:

```
<ExportedGuestMembers>
  <WTPrincipalReference isInternal="false">
    <uid>uid=user1,ou=people,cn=pdmlinkprojectlink_9.0,
      dc=host,dc=acme,dc=com|Ldap.acme.com|Ldap.acme.com</uid>
  </WTPrincipalReference>
</ExportedGuestMembers>
```

For additional information about the WTPrincipalReference element, see [Common WTPrincipalReference Element](#).

## Common ExportedRoleMemberMap Element

ExportedRoleMemberMap is an element common to Windchill context templates that is used to define the roles (and optionally, their members) associated with the context.

ExportedRoleMemberMap is defined by the following DTD elements:

```
<!ELEMENT ExportedRoleMemberMap (roleAccess?, projectMember*)>
<!ELEMENT roleAccess (Role*)>
<!ELEMENT projectMember (Role?, WTPPrincipalReference*)>
<!ELEMENT Role (customType?, uiComponent*)>
<!--ATTLIST Role
      roleType CDATA #REQUIRED
-->

<!--ELEMENT uiComponent (#PCDATA)-->
<!--ATTLIST uiComponent
      name CDATA #REQUIRED
      value (true | false) #REQUIRED
-->
```

The following table provides additional information about some of the subelements in the DTD elements for ExportedRoleMemberMap:

Subelement	Description
roleAccess	Identifies roles for which access to specific actions has been configured. Configuring roles is described in the <a href="#">Restricting the Visibility of Actions in the User Interface through Teams</a> section of the Teams chapter.  <b>Note:</b> The subelements contained in the roleAccess element can be present in an exported XML file, but should not be changed when the exported project or program is imported.
Role	Associates a role with the parent context. A role is a function that can be performed by a user or group. A role is defined by the roleType attribute.
WTPPrincipalReference	Associates a user or group with a role. See <a href="#">Common WTPPrincipalReference Element</a> .
customType	When roleType="CUSTOM", the customType element specifies a custom role name.

Subelement	Description
uiComponent	<p>When roles have been configured for access in a project or program that has been exported, this element contains the name of an action and a value. The value is true if the action is allowed and false if it is not allowed. Valid action names are maintained in an XML file and correspond to action that can be selected from the <b>Configure Roles</b> window.</p> <p><b>Note:</b> Adding or modifying existing uiComponent elements that are in an exported project or program template is not supported.</p>

The following example illustrates an ExportedRoleMemberMap element with several projectMember subelements.

```
<ExportedRoleMemberMap>
  <projectMember><Role roleType="MEMBERS"></Role></projectMember>
  <projectMember><Role roleType="COLLABORATION MANAGER"></Role></projectMember>
  <projectMember><Role roleType="CHANGE ADMINISTRATOR II"></Role></projectMember>
  <projectMember><Role roleType="PROMOTION REVIEWERS"></Role></projectMember>
  <projectMember><Role roleType="VARIANCE APPROVERS"></Role></projectMember>
  <projectMember><Role roleType="CHANGE REQUEST REVIEW BOARD"></Role>
</projectMember>

  <projectMember><Role roleType="CHANGE ADMINISTRATOR III"></Role></projectMember>

  <projectMember>
    <Role roleType="PRODUCT MANAGER"></Role>
    <WTPrincipalReference isInternal="false">
      <ufid>uid=demo,ou=people,cn=pdm,dc=ptc,dc=com</ufid>
    </WTPrincipalReference>
    <WTPrincipalReference isInternal="false">
      <ufid>uid=admin,ou=people,cn=pdm,dc=ptc,dc=com</ufid>
    </WTPrincipalReference>
  </projectMember>

  <projectMember><Role roleType="PROMOTION APPROVERS"></Role></projectMember>
  <projectMember><Role roleType="CHANGE ADMINISTRATOR I"></Role></projectMember>
</ExportedRoleMemberMap>
```



## Common ExportedTemplateFiltering Element

ExportedTemplateFiltering is an element common to Windchill context templates that is used to identify the visibility of each template in the context template being defined.

ExportedTemplateFiltering is defined by the following DTD elements:

```
<!ELEMENT ExportedTemplateFiltering ((filteredTemplateType,  
                                     filteredTemplateName,  
                                     filtered)*)>  
  
<!ELEMENT filteredTemplateType (#PCDATA)>  
  
<!ELEMENT filteredTemplateName (#PCDATA)>  
  
<!ELEMENT filtered (#PCDATA)>  
<!ATTLIST filtered filterValue (true|false) #REQUIRED>
```

The following table provides additional information about some of the subelements in the DTD elements for ExportedTemplateFiltering:

Subelement	Description
filteredTemplateType	Identifies the type of template that is filtered. Supported types include the following:  EPMDocument  LifeCycleTemplateMaster  TaskFormTemplateMaster  TeamTemplate  WfProcessTemplateMaster  WTContainerTemplateMaster  WTDDocument
filteredTemplateName	Name of the template as it appears on the templates page.
filterValue	When this value is set to true, the template is hidden. When it is set to false, it is not hidden.

## Common loadXMLFile Element

loadXMLFile is an element common to Windchill context templates that is used to include inline reading of other XML files.

Additional files that can be loaded are life cycle templates, workflow templates, and object initialization rules. The base location for these additional files is `<Windchill>/loadXMLFiles`. For example, to load the `part_lifecycle.xml` file that is in the `<Windchill>/loadXMLFiles/lifecycles` directory, include the following element:

```
<loadXMLFile>lifecycles/part_lifecycle.xml</loadXMLFile>
```

For assistance in using this element, contact PTC technical support.

## Common NotebookTemplate Element

NotebookTemplate is a element common to Windchill context templates. It defines notebook templates for product, library, project or program contexts.

The NotebookTemplate element is defined by the following DTD elements:

```
<!ELEMENT NotebookTemplate (name, description, xmlLobString)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT xmlLobString (#PCDATA)>
```

Notebooks have their own XML loader. On import, the handler extracts the `xmlLobString` element and uses the service to instantiate the notebook. On export, the `xmlLobString` element is extracted.

The following example defines the My Notebook Template:

```
<NotebookTemplate>
  <name>My Notebook Template</name>
  <description>Defines my notebook folders</description>
  <xmlLobString>
    :
    :
  </xmlLobString>
</NotebookTemplate>
```

When used within the NotebookTemplate element, the xmlLobString element can be either a CDATA section or fully-escaped XML that defines notebook folders and optional bookmarks under each folder. The following xmlLobString example uses the CDATA section to define the General, Parts, Documents, and Links folders. Under the Links folder, the PTC bookmark is defined.

```
<xmlLobString><![CDATA[
    <Notebook>
      <folder>
        <name>General</name>
      </folder>
      <folder>
        <name>Parts</name>
      </folder>
      <folder>
        <name>Documents</name>
      </folder>
      <folder>
        <name>Links</name>
        <bookmark>
          <name>PTC</name>
          <url>http://www.ptc.com</url>
        </bookmark>
      </folder>
    </Notebook>]]>
</xmlLobString>
```

**Note:** Instead of the CDATA section, you can use fully-escaped XML as is shown in the [DiscussionForumTemplate Element](#) example.

## Common OrgStructure Element

OrgStructure is an element common to Windchill context templates that has subelements for defining domains, folders, user-defined groups, and access control policy rules. OrgStructure is defined by the following DTD element:

```
<!ELEMENT OrgStructure (DomainStructure?,
                        FolderStructure?,
                        (OrgGroup?)*)>
```

The following sections describe the subelements in OrgStructure.

### DomainStructure Element

DomainStructure is an optional element of OrgStructure that defines a domain hierarchy for a context. This section provides the syntax of the DomainStructure element. For a detailed explanation of domains and the out-of-the box Site domains that are created, see [Administering Domains and Policies](#) in the [Contexts](#) chapter.

The DomainStructure element is defined by the following DTD elements:

```
<!ELEMENT DomainStructure (domainLevel*)>
<!ELEMENT domainLevel (parentDomain?,
                      domainName,
```

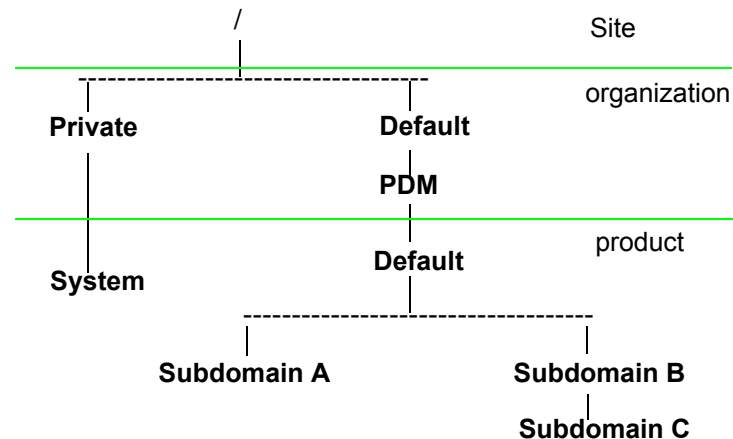
```
description?,
domainLevel*)>
```

The following table provides additional information about the subelements in the DTD elements for domainLevel.

Subelement	Description
parentDomain	<p>Specifies the domain to use as the parent domain for the domain being defined in the domainLevel element. The value is a domain path that is relative to the context in which the parent domain resides. When defining a domain that is a top-level domain in a context, the parent domain must be in the parent context of the context being defined in the template. For example, if the context being defined through the template is a product context and a top-level domain is being defined, then the parent domain must be a domain in an organization context.</p> <p>Always start the path with the forward slash (/) and use the forward slash to separate the names of the domains in the hierarchy relative to the context.</p> <p>When defining the top-level domains in a context, always specify the parentDomain element.</p> <p><b>Note:</b> For backward compatibility, if parentDomain is not specified in the outermost domainLevel subelement of the DomainStructure element, then the parent domain is assumed to be the /Default domain in the context being defined.</p> <p>In nested domainLevel elements, omit the parentDomain element. When the parentDomain element is not specified in a nested domainLevel element, then the parent domain is derived using the domainName and parentDomain elements specified in the enclosing domainLevel element.</p> <p>If parentDomain is specified in a nested domainLevel subelement, then the domainLevel element is treated as if it was not in a nested structure, meaning that the included parentDomain element determines the parent domain.</p> <p>See the examples provided later in this section for suggested uses of this element.</p>
domainName	Specifies the name of the domain.

Subelement	Description
description	Specifies the description of the domain. You can enter a maximum of 200 characters in the description.
domainLevel	Allows you to nest domains.

The following example illustrates the domain structure of an exported product template where three subdomains have been included in addition to the /Default and /System domains normally defined in a product context. Specifically, it creates subdomains under the /Default domain in a product context as follows:



The following example DomainStructure element shows the domain structure of an exported product template with the new subdomains in the context as illustrated previously:

```

<DomainStructure>
  <domainLevel>
    <parentDomain>/Private</parentDomain>
    <domainName>System</domainName>
  </domainLevel>

  <domainLevel>
    <parentDomain>/Default/PDM</parentDomain>
    <domainName>Default</domainName>

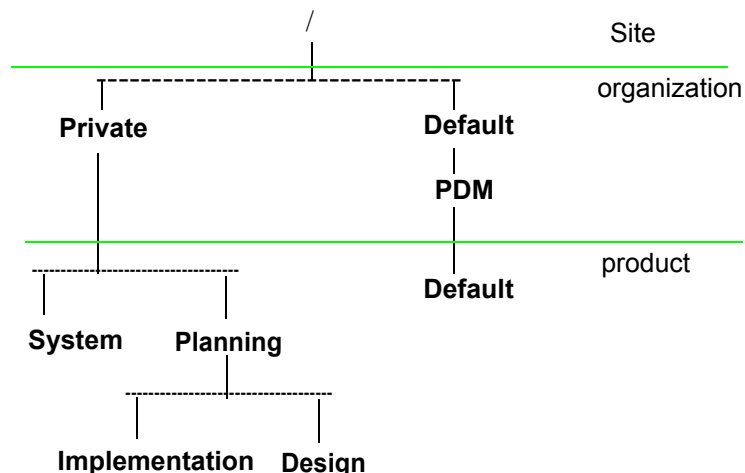
    <domainLevel>
      <domainName>Subdomain A</domainName>
    </domainLevel>

    <domainLevel>
      <domainName>Subdomain B</domainName>
      <domainLevel>
        <domainName>Subdomain C</domainName>
      </domainLevel>
    </domainLevel>
  </DomainStructure>

```

In this example, two top-level domains are defined for the product context: /System and /Default. Then under the /Default domain are subdomains named /Subdomain A and /Subdomain B. Under /Subdomain B is /Subdomain C.

The following example illustrates the addition of a top-level domain named /Planning in a product context and the addition of two domains named /Implementation and /Design that are nested under the /Planning domain. Specifically, it creates the top-level /Planning domain whose parent domain is the /Private domain that is in the organization context. Under the /Planning domain, there are two subdomains named /Implementation and /Design as follows:



The following example DomainStructure element shows the domain structure of an exported product template with the new /Planning top-level domain in the context as illustrated previously:

```

<DomainStructure>
  <domainLevel>
    <parentDomain>/Private</parentDomain>
    <domainName>System</domainName>
  </domainLevel>
  <domainLevel>
    <parentDomain>/Private</parentDomain>
    <domainName>Planning</domainName>
    <domainLevel>
      <domainName>Implementation</domainName>
    </domainLevel>
    <domainLevel>
      <domainName>Design</domainName>
    </domainLevel>
  </domainLevel>
  <domainLevel>
    <parentDomain>/Default/PDM</parentDomain>
    <domainName>Default</domainName>
  </domainLevel>
</DomainStructure>

```

In this example, three top-level domains are defined for the product context: /System, /Planning, and /Default. Then under the /Planning domain are subdomains named /Implementation and /Design.

## FolderStructure Element

FolderStructure is an optional element of OrgStructure that defines the initial folder structure defined in a context. Folders are available to organize information.

Additionally, you can specify ad hoc access control rules for folders using this element. For products and libraries, PTC recommends that you use a domain and access control policy rules for that domain instead of ad hoc rules. For details, see [Assigning Domains to Folders in Solutions with Products and Libraries](#) in the Contexts chapter.

FolderStructure is defined by the following DTD elements:

```
<!ELEMENT FolderStructure (cabinet,  
                           domainName?,  
                           rootPath,  
                           AdHocACLEntrySet?,  
                           (subFolder | nestedFolder | folderLink |  
                            folderContentObjectReference)*,  
                           folderAcl*)>  
  
<!ELEMENT AdHocACLEntrySet (AdHocACLEntry+)>  
  
<!ELEMENT nestedFolder (name,  
                        domainName?,  
                        AdHocACLEntrySet?,  
                        nestedFolder*)>  
  
<!ELEMENT folderLink (domainName?,  
                      folderPath,  
                      name,  
                      urlString,  
                      description?)>  
  
<!ELEMENT folderContentObjectReference (folderPath,  
                                         ((referenceName,referenceNumber) | ObjectID))>  
<!ELEMENT ObjectID (localId?, objectId?, ufid?)>
```

The following table provides additional information about some of the subelements in the DTD elements for FolderStructure:

Subelement	Description
applicationKey	Not supported.
folderContentObjectReference	Not supported.
domainName	Relative domain path, starting at a top-level domain in the context. Include the forward slash (/) at the beginning of the path.  <b>Note:</b> The format of this element is different from the format of the domainName subelement contained in the DomainStructure element.
AdHocACLEntrySet	List of permissions to grant for the cabinet or folder.  See <a href="#">Common AdHocACLEntrySet Element</a> .
folderAcl	Provided for backward compatibility; not supported in exported files.
aclPermissionList	Provided for backward compatibility; not supported in exported files.

The following example illustrates a folder structure that parallels the additional domains illustrated in the previous section:

**Note:** A unique domain is assigned to two of the folders. Using a unique domain allows special access control policy rules to be set for those folders. Also note that the nestedFolder element is used to define each folder. A subFolder element can not be used because that element does not include a subelement for associating a domain.

```
<FolderStructure>
  <cabinet>Default</cabinet>
  <rootPath>/</rootPath>

  <nestedFolder>
    <name>Folder A</name>
  </nestedFolder>

  <nestedFolder>
    <name>Folder B</name>
    <domainName>/Default/Subdomain B</domainName>
    <nestedFolder>
      <name>Folder C</name>
```



```

        <domainName>/Default/Subdomain B/Subdomain C</domainName>
    </nestedFolder>
</nestedFolder>
</FolderStructure>

```

The following example illustrates a folder structure that defines the Specifications subfolder and ad hoc access control rules that apply to the subfolder:

```

<FolderStructure>
  <cabinet>/Default</cabinet>
  <domainName>/Default</domainName>
  <rootPath>/</rootPath>
  <AdHocACLEntrySet>
    <AdHocACLEntry>
      <WTPPrincipalReference isInternal="true">
        <groupName>teamMembers</groupName>
        <groupType>teamMembers</groupType>
      </WTPPrincipalReference>
      <AccessPermissionSet>
        <permissionField name="READ"></permissionField>
        <permissionField name="MODIFY"></permissionField>
        <permissionField name="DOWNLOAD"></permissionField>
        <permissionField name="MODIFY_CONTENT"></permissionField>
      </AccessPermissionSet>
      <ownerField name="WNC_ACCESS_CONTROL"></ownerField>
    </AdHocACLEntry>
  </AdHocACLEntrySet>
  <nestedFolder>
    <name>Specifications</name>
    <domainName>/Default</domainName>
    <AdHocACLEntrySet>
      <AdHocACLEntry>
        <WTPPrincipalReference isInternal="true">
          <groupName>GUEST</groupName>
          <groupType>GUEST</groupType>
        </WTPPrincipalReference>
        <AccessPermissionSet>
          <permissionField name="READ"></permissionField>
          <permissionField name="DOWNLOAD"></permissionField>
        </AccessPermissionSet>
        <ownerField name="WNC_ACCESS_CONTROL"></ownerField>
      </AdHocACLEntry>
    </AdHocACLEntrySet>
  </nestedFolder>
</FolderStructure>

```

## OrgGroup Element

OrgGroup is an optional element of OrgStructure that defines the names of the user-defined groups defined in a context. The OrgGroup element can occur any number of times.

The OrgGroup element is defined by the following DTD element:

```
<!ELEMENT OrgGroup (groupName, description?)>
```

The following example illustrates the creation of some user-defined groups in a context.

```
<OrgGroup>
  <groupName>Group_1</groupName>
  <description>Description of Group_1</description>
</OrgGroup>

<OrgGroup>
  <groupName>Group_2</groupName>
  <description>Description of Group_2</description>
</OrgGroup>
```

## Common PreferenceInstance Element

PreferenceInstance is an element common to Windchill context templates that has subelements for defining a preference setting within a context template.

PreferenceInstance is defined by the following DTD elements:

```
<!ELEMENT PreferenceInstance (name,
                                value,
                                preferenceClient?,
                                locked?)>

<!ELEMENT preferenceClient (#PCDATA)>

<!ELEMENT locked (#PCDATA)>
```

The subelements name, value, and preferenceClient that are contained in the PreferenceInstance element are present in an exported XML file for a context when the exported file includes the preferences set in the context. When the exported context is imported, do not change these element values.

Valid values for the locked subelement are true and false. The default value is false. If locked for the context, this value overrides the value set at the user level.

## Common ProjectEnumType Element

**Note:** ProjectEnumType is only applicable when Windchill ProjectLink is installed.

ProjectEnumType is an element common to Windchill context templates that is used to define the project or program roles. If this element is set in both OrganizationConfig and ProjectConfig, then the roles are combined.

Any number of ProjectEnumType elements can be specified.

ProjectEnumType is defined by the following DTD elements:

```
<!ELEMENT ProjectEnumType (typeName, projectId?, typeValues)>

<!ATTLIST ProjectEnumType
    excludeByDefault (true | false) "true"
>
<!ELEMENT projectId (ObjectID)>
<!ELEMENT typeValues (typeEntry*)>
<!ELEMENT typeEntry (#PCDATA)>
```

The following table provides additional information about some of the subelements in the DTD elements for ProjectEnumType:

Subelement	Description
projectId	Object identifier of the project or program. For details on ObjectID, see <a href="#">ObjectID Element</a> .
excludeByDefault	<p>Defines what to do if the current typeName has no excluded values specified yet in the target context. If there are excluded values already present for the type in the target context, then this value is ignored; otherwise, a value of true tells the system to exclude any static types not specified in the list.</p> <p>A value of false tells the system that no static values are to be included. The net effect is that if any static values are added to this tag they will be ignored. A static value is one defined in the rbInfo file for the enumerated type.</p>

The following example illustrates ProjectEnumType:

```
<ProjectEnumType>
<typeName>wt.project.Role</typeName>
<typeValues>
    <typeEntry>APPROVER</typeEntry>
    <typeEntry>CONSULTANT</typeEntry>
    <typeEntry>CONTRACTOR</typeEntry>
    <typeEntry>DESIGNER</typeEntry>
    <typeEntry>ENGINEER</typeEntry>
    <typeEntry>GROUP LEADER</typeEntry>
    <typeEntry>MANUFACTURER</typeEntry>
```

```

<typeEntry>MEMBERS</typeEntry>
<typeEntry>OBSERVER</typeEntry>
<typeEntry>OEM</typeEntry>
<typeEntry>ORGANIZATION LEADER</typeEntry>
<typeEntry>PRODUCT MANAGER</typeEntry>
<typeEntry>PROGRAM DIRECTOR</typeEntry>
<typeEntry>PROGRAM MANAGER</typeEntry>
<typeEntry>PROJECT ACCOUNTANT</typeEntry>
<typeEntry>PROJECT ADMINISTRATOR</typeEntry>
<typeEntry>PROJECT MANAGER</typeEntry>
<typeEntry>PROJECT SPONSOR</typeEntry>
<typeEntry>RESOURCE MANAGER</typeEntry>
<typeEntry>REVIEWER</typeEntry>
<typeEntry>SUPPLIER</typeEntry>
<typeEntry>TEAM LEADER</typeEntry>
</typeValues>
</ProjectEnumType>

```

## Common SharedTeamDef Element

SharedTeamDef is an element common to Windchill application context templates that is used to define the name of the shared team that to be used in the context team.

To use a context template that names a shared team, the shared team must exist in the organization context from which the application context is being created.

For information about shared teams, see [Context Teams](#).

SharedTeamDef is defined by the following DTD elements:

```

<!ELEMENT SharedTeamDef (sharedTeamRef?,
                        sharedTeamDescription?,
                        sharedTeamOwner?,
                        sharedTeamInvitation?,
                        ExportedRoleMemberMap?,
                        ExportedGuestMembers?,
                        enabled?,
                        extendable?,
                        AccessControlRule*,
                        AdHocACLEntrySet*)>

<!ELEMENT sharedTeamRef (#PCDATA)>

```

The following example specifies a shared team named Design Team:

```
<SharedTeamDef><sharedTeamRef>Design Team</sharedTeamRef></SharedTeamDef>
```

**Note:** This element is also used for defining shared teams for import. See [Creating Business XML Files for Shared Teams](#).

## Common TypeBasedRule Element

TypeBasedRule is an element common to Windchill context templates that is used to define default values for objects when those objects are initialized or copied (object Initialization rules).

The TypeBasedRule element can occur any number of times and is defined by the following DTD elements:

```
<!ELEMENT TypeBasedRule (EngineRule, className)>

<!ELEMENT EngineRule (ruleName, ruleSpecification?, ruleType)>
<!--ATTLIST EngineRule
      isDefault (true | false) "false"
      enabled (true | false) "true"
-->

<!ELEMENT ruleType EMPTY>
<!--ATTLIST ruleType
      type (INIT | COPY | TYPE) #REQUIRED
-->
```

To specify an object initialization rule in the TypeBasedRule element, include the following elements:

```
<TypeBasedRule>
  <EngineRule isDefault="false" enabled="true">
    <ruleName>name_of_rule</ruleName>
    <ruleSpecification><![CDATA[
      :
      :
    ]]>
    </ruleSpecification>
    <ruleType type="INIT"></ruleType>
  </EngineRule>
  <className>object_type</className>
</TypeBasedRule>
```

In this set of elements, specify your own rule name in place of *name\_of\_rule* and specify the object type for the object in which the rule is being set in place of *object\_type*.

Also, place the actual rule after the following line:

```
<ruleSpecification><![CDATA[
```

For additional information on the rules that you can specify in the ruleSpecification element, see the [Object Initialization Rules](#) chapter. For a detailed example of the ruleSpecification content, see [Setting File-based or State-based Versioning for Objects](#) in the Object Initialization Rules chapter.

The following example shows all XML elements that are required in an XML file used for importing a context template that sets the default document folder path for a document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TypeBasedRule SYSTEM "standardX05.dtd">
<!-- WTDocument -->
<TypeBasedRule>
  <EngineRule isDefault="false" enabled="true">
    <ruleName>Document Folder Configuration</ruleName>
    <ruleSpecification><![CDATA[
      <AttributeValues objType="wt.doc.WTDocument">
        <AttrValue id="folder.id" algorithm=
"com.ptc.core.foundation.folder.server.impl.FolderPathAttributeAlgorithm">
          <Arg>/Default</Arg>
        </AttrValue>
      </AttributeValues>
    ]]>
    </ruleSpecification>
    <ruleType type="INIT"></ruleType>
  </EngineRule>
<className>wt.doc.WTDocument</className>
</TypeBasedRule>
```

The following example illustrates setting initialization values for life cycles associated with Design and BOM documents:

```
<!-- Design -->
<TypeBasedRule>
  <EngineRule isDefault="false" enabled="true">
    <ruleName>DESIGN_RULE_NAME</ruleName>
    <ruleSpecification><![CDATA[
      <AttributeValues objType="wt.doc.WTDocument">
        <!-- set the life cycle -->
        <AttrValue id="lifeCycle.id"
          algorithm="com.ptc.core.foundation.lifecycle.server.impl.
            LifeCycleTemplateAttributeAlgorithm">
          <Arg>BASIC_LIFECYCLE_NAME</Arg>
        </AttrValue>
      </AttributeValues>]]>
    </ruleSpecification>
  </EngineRule>

  <!-- Value of DESIGN_TYPE_NAME cannot contain spaces -->
  <className>wt.doc.WTDocument|DESIGN_TYPE_NAME</className>
</TypeBasedRule>

<!-- Bill of Material (BOM) -->
<TypeBasedRule>
  <EngineRule isDefault="false" enabled="true">
    <ruleName>BOM_RULE_NAME</ruleName>
    <ruleSpecification><![CDATA[
      <AttributeValues objType="wt.doc.WTDocument">
        <!-- set the lifecycle -->
        <AttrValue id="lifeCycle.id"
          algorithm="com.ptc.core.foundation.lifecycle.server.impl.
            LifeCycleTemplateAttributeAlgorithm">
          <Arg>BASIC_LIFECYCLE_NAME</Arg>
        </AttrValue>
      </AttributeValues>]]>
    </ruleSpecification>
  </EngineRule>

  <!-- Value of BOM_TYPE_NAME cannot contain spaces -->
  <className>wt.doc.WTDocument|BOM_TYPE_NAME</className>
</TypeBasedRule>
```

## Common WTPrincipalReference Element

WTPrincipalReference is an element common to Windchill application context templates and shared team import files that is used to define a user, user-defined group, system group, or organization.

WTPrincipalReference is defined by the following DTD elements:

```
<!ELEMENT WTPrincipalReference (ufid|name|
                                (groupName,groupType))>

<!ATTLIST WTPrincipalReference
  isInternal (true|false) #REQUIRED
  classType CDATA #IMPLIED
  name CDATA #IMPLIED
  userEmail CDATA #IMPLIED
>

<!ELEMENT groupName (#PCDATA)>
<!ELEMENT groupType (#PCDATA)>
```

The following table provides additional information about some of the subelements in the DTD elements for WTPrincipalReference:

Subelement	Description
isInternal	This required element identifies whether definition is for a system group or for a user, user-defined group, or organization. Specify true to define a system group and false to define a user, user-defined group, or organization.
ufid	Specifies the distinguished name of the principal and identifies the directory service where the LDAP entry for the user, user-defined group, or organization resides. For more information, see <a href="#">Principals (Users, Groups, and Organizations)</a> . Use this subelement when isInternal is set to "false".
name	Specifies the principal name as defined in Windchill. This subelement is used within load files to identify principals created as part of the installation process. Do not use this subelement in context templates.
classType	Optional element that specifies the type associated with the principal. For example, users have a class type of wt.org.WTUser. This subelement is included in an exported template when isInternal is set to "false".



Subelement	Description
name	Optional element that specifies the name of a group or organization principal.  This subelement is included in an exported template when isInternal is set to "false" and the principal is a group or organization.
userEmail	Optional element that specifies the e-mail address of a user.  This subelement is included in an exported template when isInternal is set to "false" and the principal is a user in which the e-mail address is known.
groupName, groupType	Specifies the name and type associated with a system group. Windchill uses the information provided to find the group.  Use this subelement when isInternal is set to "true".  To obtain correct the name and type for the system groups that you want to specify, export an existing context that uses the groups and inspect the XML file that is produced.

The following WTPrincipalReference elements are examples:

```

<WTPrincipalReference isInternal="false">
  <ufid>uid=user1,ou=people,cn=pdmlinkprojectlink_9.0,
    dc=host,dc=acme,dc=com|Ldap.acme.com|Ldap.acme.com</ufid>
</WTPrincipalReference>

<WTPrincipalReference isInternal="true">
  <groupName>PRODUCT CREATOR</groupName>
  <groupType>PRODUCT CREATOR</groupType>
</WTPrincipalReference>

<WTPrincipalReference isInternal="true">
  <groupName>teamMembers</groupName>
  <groupType> ((null))</groupType>
</WTPrincipalReference>

<WTPrincipalReference isInternal="true">
  <groupName>Shared Team Creator</groupName>
  <groupType>Shared Team Creator</groupType>
</WTPrincipalReference>

<WTPrincipalReference isInternal="true">
  <groupName>GUEST</groupName>
  <groupType> ((null))</groupType>
</WTPrincipalReference>

```



# Index

## A

### Access control

- cabinets, 12-23
- content holder objects, 12-19
- deleted organizations, 8-24
- deleted user-defined groups, 8-21
- deleted users, 8-15
- domains, 12-3
- life cycles and, 15-31, 15-44
- overview, 12-2
- Process Manager, 16-41
- Product Design template, 15-15

### Access control lists

- about, 12-10
- ad hoc, 12-2, 12-15, 12-24, 15-31, 15-44
- defining, 12-3
- derived from access control policies, 12-11–12-12
- policy, 12-2, 12-24, 15-31, 15-44
- types of, 12-2

### Access control policies

- defining, 4-7
- deriving access control lists from, 12-11–12-12
- domain, 12-11
- libraries, 7-5
- organizations, 5-7
- products, 7-5
- programs, 6-7
- projects, 6-7

### Access control rules

- about, 12-10
- creating, 12-4
- default, 12-16
- Default domain, 3-12
- default, for cabinets, 12-23
- defined, 12-3
- domains, 12-14
- examples, 12-2, 12-14–12-15
- introduction, 2-8
- root domain, 3-10
- System domain, 3-13
- User domain, 3-12

User/Unaffiliated domain, 3-12

Windchill PDMLink, 7-15, 7-34

window, 12-8

AccessControlRule element, A-50

ACL. See Access control lists

ACLRule element, A-50

Action items, creating and editing, 6-5

### Activities

completing, 6-5

managing, 6-5

Actors, 9-2

Ad hoc access control rules, 12-2, 12-4

Ad hoc ACLs. See Access control lists, ad hoc

AdHocACLEntrySet element, A-52

Administrative directory, 1-2

Administrative Domain window, 12-5, 18-3

Administrative notifications, 8-26

Administrative permission, 12-8

Administrator user, 1-15, 2-5

### Administrators

access to objects, 2-5

group, 2-5

selecting users, 1-15

Windchill PDMLink, 1-13

Windchill ProjectLink, 1-14

### Adobe

Designer software, 16-71

Forms software, 16-71

task form templates, 16-71

XDP templates, 16-71

### Algorithms

AttrConstraint, 11-13

AttrValue, 11-10

conditional logic, 11-16

All permission, 12-6

ALL pseudo-group, 12-11

AndTest algorithm, 11-18

Application context managers, 3-3

### Application contexts

libraries, 3-5

products, 3-5

programs, 3-5

projects, 3-5

types, 3-2, 3-5

- Arbortext
  - dynamic documents, 13-43
  - folders, 7-47
  - introduction, 2-22
  - products, 7-2
  - publishing, 2-21
  - soft attributes, 13-46
  - soft types, 13-44
- Arbortext Content Manager
  - getting started, 1-2
  - home page, 1-7
- Archiving, 4-10, 5-9
- ASME Y 14.35M Standard, 11-36
- Assigned activities. *See* Workflow activities
- Attributes
  - Arbortext, 13-46
  - constraints, 13-31
  - creating, 13-7, 13-22
    - See also* Permissions
  - dynamic documents, 13-46
  - file-based, 13-43
  - nonfile-based, 13-43
  - setting access in CAD applications, 13-43
  - setting access in Windchill, 13-43
- Audit report query
  - creating, 21-16
  - example, 21-16
- Auditing
  - captured attributes, 21-8
  - creating queries, 21-12
  - generating queries, 21-13
  - licensing, 21-14
  - organization, 5-7
  - purging logs, 21-13
  - service events, 21-3
  - summary events, 21-6
  - system information, 4-7
  - viewing queries, 21-13
- Auditing Administration Utility
  - accessing, 21-12
  - Audit Log Purge Manager, 21-13
  - License Usage Reporting, 21-14
  - Organization Usage Reporting, 21-15
  - Security Audit Report Queries, 21-13
  - Security Audit Reporting, 21-12

## B

- Background queues, 2-3
- Best practices
  - object initialization rules, 7-53
  - organizations, 5-25
  - site, 4-16
- BooleanBranch, 11-18
- Bootstrap client, 2-3
- Burst configuration files, 2-19, 2-20
- Business XML files, 10-9, A-1

## C

- Cabinets
  - access control, 12-20
  - default access control rules, 12-23
  - owners, 12-21
  - personal
    - defined, 12-21
    - deleted users, 8-16
    - new user, 12-23
  - shared
    - defined, 12-21
    - rules, 12-28
- Caching
  - access control lists, 12-11, 12-14
  - notification lists, 19-4
- CAD agent
  - characteristics, 20-5
  - timeouts, 20-9
- CAD documents
  - automated publishing, 20-10
  - object type, 2-19, 13-39
  - templates, 2-21, 3-15
  - type, 2-21
  - visualization, 2-20
- CAD workers
  - configuring, 4-9
  - definition, 20-5
- CAGE code, 5-20
- Calendar
  - managing, 4-9
  - wt.property, 4-10
- CaseBranch, 11-18
- Change Activity workflow, 16-53
- Change Activity, team roles, 9-13
- Change Admin roles, 7-14, 7-26, 7-33, 7-37
- Change Content permission, 12-8
- Change Domain permission, 12-8
- Change Management objects, 9-13
- Change modification tracking, 7-9

- Change Monitor Reports, 14-5
- Change Notice workflow, 16-55
- Change objects, 12-19
- Change Permissions permission, 12-8
- Change Request workflow, 16-54
- Change tab, 1-4
- Change transition rule, 15-7, 15-10
- Collaboration, 2-23
- Collections
  - defined, 18-2
  - deleted users, 8-16
- Composite object initialization rules, 11-5
- Configuring
  - preferences, 4-4
  - properties, 4-4
- Constraints
  - Constraint Editor, 13-26
  - example, 13-19
  - out-of-the-box, 13-30
  - set on attributes, 13-13
  - setting, 13-25
- Containers
  - overview, 2-3
  - See also Contexts
- ContainerTemplate element, A-4
- Content holder objects, 12-19
- Content replication
  - programs, 6-8
  - projects, 6-8
  - system, 2-3
- Context configuration
  - description, 3-4
  - editing, 3-6
  - private, 3-5
  - public, 3-5
- Context data
  - description, 3-13
  - editing, 3-14
  - installation, 3-13
  - organizations, 5-10
  - site, 4-15
- Context environment
  - programs, 6-10
  - projects, 6-10
- Context object initialization rules
  - accessing, 11-3
  - loading, 11-3
  - site, 4-15
  - specifying, 11-7
  - XML, 11-8
- Context participation
  - description, 3-7
  - editing, 3-8
  - groups, 3-7, 3-8
  - installed, 3-8
  - organization, 5-10
  - programs, 6-10
  - projects, 6-10
  - roles, 3-7, 3-8
  - site, 4-15
- Context policies
  - description, 3-9
  - installation, 3-9
  - organization, 5-10
  - site, 4-15
  - updating, 3-13
- Context preferences, 3-23
- Context structure
  - description, 3-6
  - editing, 3-7
  - installed, 3-6
  - organizations, 5-10
  - programs, 6-10
  - projects, 6-10
- Context team, 9-2
- Contexts
  - creating, 3-23
  - definition, 1-10, 2-4
  - domains, 3-32
  - hierarchy, 3-2
    - products and libraries, 2-7
    - projects and programs, 2-7
  - introduction, 1-10
  - libraries, 7-2
  - organization, 3-4
  - preferences, 3-23
  - products, 7-2
  - site, 3-2, 3-4
  - teams, 2-15, 9-3
  - templates, 2-5, 3-15, 4-15, 10-2
  - use, 3-3
- Copy forward
  - markups, 20-19
  - properties, 20-20
  - representations, 20-19
- Create By Move permission, 12-7, 15-34
- Create permission, 12-7, 15-34

## Creating

- access control rules, 12-4
- action items, 6-5
- activities, 6-5
- attributes, 13-7
- audit report query, 21-16
- context templates, 10-4
- contexts, 3-23
- deliverables, 6-5
- documents and folders, 6-5
- explicit mapping, 13-42
- indexing rules, 18-3–18-4
- libraries, 7-43
- life cycle templates, 15-17
- measurement systems, 13-7, 13-15
- milestones, 6-5
- notification rules, 19-2
- organization folders and documents, 5-5
- organizations, 4-3, 5-17, 8-22, 8-23
- products, 7-42
- profiles, 8-29
- programs, 6-3
- projects, 6-3
- quantities of measure, 13-7, 13-16
- resources, 6-5
- roles, 9-26
- site folders and documents, 4-4
- soft attribute, 13-22
- soft types, 13-19
- types, 13-19
- user, 8-12
- user-defined groups, 8-18
- users, 8-12
- views and view associations, 17-2
- workflow
  - activities, 16-19
  - process links, 16-37
  - subprocesses, 16-34

## Creator

- actor, 9-2
- product and library, 1-13, 5-11
- program, 1-14
- project, 1-14
- projects and programs, 5-11

## Custom reports, 14-4, 14-5

## Customizing

- online help, 2-3
- online tutorials, 2-3

## D

### Data types, 2-20

Default domain, 2-9, 3-12, 3-29

DefaultEPMDocument, 2-21, 13-39

Defining, 16-34

Delete permission, 12-8

### Deleting

- organizations, 8-23

- user-defined groups, 8-18, 8-20

- users, 8-12, 8-15

### Deliverables

- completing, 6-5

- creating and editing, 6-5

Disconnected principals, 8-28

DiscussionForumTemplate element, A-46

Document Library template, 7-10

### Document templates

- definition, 3-15

- list, 3-18

- preferences, 3-21

### Document to document relationships

- libraries, 7-53

- products, 7-53

- programs, 6-18

- projects, 6-18

### Documents

- content holder, 12-19

- organization, 5-5

### Domains

- access control rules, 12-14, 12-18, 12-21

- administering, 3-25

- assign to folders, 3-34

- creating, 3-30

- definition, 3-25

- display, 12-21

- indexing rules and, 18-3, 19-3

- installed, 3-25

- introduction, 1-10

- naming, 3-30

- overview, 2-3, 3-31

- Root, 3-29, 3-30, 11-25

- Site container, 3-29

- subfolders, 12-21

- System, 3-29, 11-25

Download permission, 12-7

- DTD
  - importing, 10-8
  - library elements, A-10
  - object initialization rules, 11-7
  - organization elements, A-2
  - product elements, A-7
  - project elements, A-13
  - shared team elements, A-48
- DUNS number, 5-20
- Dynamic documents
  - attributes, 13-46
  - folders, 7-47
  - introduction, 2-21
  - managing, 7-46
  - numbering scheme, 7-47
  - publish rules, 20-23
  - soft type, 2-19, 2-20, 2-21, 13-43

## E

- Editing
  - action items, 6-5
  - activities, 6-5
  - deliverables, 6-5
  - documents and folders, 6-5
  - life cycle, 15-20, 15-26
  - members, 4-4
  - organization folders and documents, 5-5
  - organizations, 8-23
  - programs, 6-3
  - projects, 6-3
  - resources, 6-5
  - types, 13-28
  - user-defined groups, 8-18
  - users, 8-12
- Electronic signatures, 8-12, 16-72
- E-mail notifications. See Notifications
- End users, 1-15
- Enterprise directory, 1-2, 1-16, 8-5
- Enumerated types, 9-27
- EnumTypeConstant, 11-10, 11-12
- EPMDocument, 2-19, 2-21, 20-4
- EqualsTest, 11-17
- ERP system, working with, 4-10
- Events
  - captured attributes, 21-8
  - event dependent attributes, 21-10
  - service, 21-3
  - summary, 21-6
- Execution flags
  - activity, 16-65
  - modifying, 16-68

- process, 16-61
  - process and activity, 16-67
  - workflow, 16-60
- Execution Options, 16-28
- executionInfo element, A-29
- ExportedGuestMembers element, A-54
- ExportedRoleMemberMap element, A-55
- ExportedTemplateFiltering element, A-57
- Exporting
  - information, 4-9
  - life cycles, 15-40
  - templates, 10-4
  - watermarks, 20-18
  - workflow process templates, 16-39
  - ZIP file contents, 10-6
- External file vaults, 2-3

## F

- File vaults, 2-3
- File-based versioning, 11-29
- Folder members, 12-20, 12-28
- Foldered objects. See Folder members
- FolderPathAttributeAlgorithm, 11-10, 11-11
- Folders
  - access control, 12-20
  - assigning domains, 3-34
  - dynamic documents, 7-47
  - managing, 6-5
  - organization, 5-5
  - products and libraries, 7-4
  - table, 4-5
- Full name, 9-28

## G

- General Library template, 7-10, 7-33
- General Product template, 7-10, 7-33
- GetDiscreteSetConstraint, 11-13
- GetHiddenConstraint, 11-13
- GetImmutableConstraint, 11-13, 11-14
- GetServerAssignedConstraint, 11-13, 11-14
- GetServerPreGeneratedValue, 11-13, 11-14
- Groups
  - administrative permissions, 13-18
  - deleting, 8-20
  - managing, 8-18
  - organization, 5-4
  - setting permissions, 12-8
  - workflow security, 16-5
- Guests role, 7-14, 7-17, 7-33, 7-35

## H

- Harvard series, 11-25
- Help, customizing, 2-3
- Home page
  - Arbortext Content Manager, 1-7
  - Pro/INTRALINK 9.0, 1-9
  - Windchill PDMLink, 1-4
  - Windchill ProjectLink, 1-6
- Home tab, 1-4, 1-6, 1-8, 1-9

## I

- IfContainerTypeEqualsProjectTest, 11-18
- IfNotNullTest, 11-17
- IfNullTest, 11-17
- Import and export
  - libraries, 7-8
  - organizations, 5-9
  - products, 7-8
  - programs, 6-6
  - projects, 6-6
  - site, 4-9
- Importing
  - information, 4-9
  - life cycles, 15-40
  - Microsoft Project plan, 6-7
  - templates, 10-5
  - workflow process templates, 16-39
  - ZIP file contents, 10-8
- importList.txt file, 10-6
- Indexing lists, 18-4
- Indexing policies, 18-4
- Indexing rules
  - combinations, 18-5
  - creating, 18-2, 18-3–18-4
  - creating and managing, 18-3
  - defined, 18-3
  - object types, 18-4
  - root domain, 3-13
  - See also Collections
- Integral Windchill solutions, 2-8
- ISO6523, 5-20
- Iterated objects, 15-17

## J

- JAR files, 15-40, 15-41
- java.security.acl package, 12-10, 12-12

## L

- LDAP, 8-5
- Legacy reports, 14-4
- Libraries
  - context, 3-2
  - creating, 7-43
  - tab, 1-5, 1-8, 1-10
  - templates, 7-9, 10-3, A-10
- Library creators, 1-13, 5-11
- Library domains
  - access control, 7-15, 7-34
  - teams, 9-16
- Library manager
  - access rules, 7-36
  - creating, 1-13
  - definition, 1-12
  - next steps, 1-17
  - role, 7-33
  - typical duties, 7-2
- Library templates, 3-15, 7-10, A-10
- Library Utilities page, 7-46
- LibraryConfig element, 10-9, A-10
- License
  - compliance, 21-15
  - usage reports, 21-14, 21-15
    - Arbortext, 21-14
    - Pro/INTRALINK, 21-15
    - Windchill, 21-14
- Life cycle access control rules
  - defining, 15-31
  - overview, 12-24
- Life Cycle Administrator
  - accessing, 15-2
  - creating, 15-18
  - editing, 15-18
  - export, 15-40, 15-41
  - import, 15-41
  - iteration history, 15-17
  - overview, 15-2
  - toolbar, 15-18
- Life cycle phases
  - access control, 15-25
  - associating workflow processes, 15-36
  - defining, 15-21
  - properties, 15-19, 15-24
  - roles, 15-25



- Life cycle roles
  - assigning permission, 15-32
  - mapping to team roles, 12-25
  - mappings, 15-29
  - predefined, 15-29
  - role resolution, 9-20, 16-23
  - selecting, 15-28
- Life cycle states
  - about, 15-3
  - defining, 15-43
  - indexing rules, 18-2
  - managing, 15-5
- Life cycle templates
  - advanced, 15-5
    - PDMLink, 15-15
    - ProjectLink, 15-17
  - basic, 15-4
    - PDMLink, 15-14
    - ProjectLink, 15-16
  - checking out, 15-17
  - context, 3-15
  - creating, 15-17
  - installation, 3-17
  - out-of-the-box, 15-13
- Life cycle-managed information, 12-24
- Life Cycles
  - associating workflow processes, 15-36
- Life cycles
  - access control, 12-24, 15-44
  - default, 15-3, 15-20
  - definition, 15-2
  - editing, 15-17
  - importing and exporting, 15-40
  - promotion criteria, 15-38
  - properties, 15-19, 15-22
  - team roles, 9-17
  - team templates, 9-12
- LifeCycleTemplateAttributeAlgorithm, 11-10, 11-11
- loadXMLFile element, A-58
- Local team, 9-3, 9-4
- Lock transition rule, 15-10
- Logging on, 1-3
- Logical identifier
  - attributes, 13-14
  - specifying Arbortext types, 13-46
  - specifying Windchill types, 13-35
  - types, 13-21

## M

- Maintenance only release, 15-40, 16-39
- Managing
  - documents and folders, 6-5
  - organization folders and documents, 5-5
  - organizations, 4-3
  - project roles, 6-4
  - project team members, 6-4
  - project templates, 6-6
  - site folders and documents, 4-4
  - views and view associations, 17-3
- Members
  - adding to a site, 4-4
  - organization, 2-16
  - role, 7-14, 7-33
- Milestones
  - completing, 6-5
  - creating and editing, 6-5
- MilSpec, 11-36
- MILSTD, 11-32, 11-36
- Modify Content permission, 12-7
- Modify Identity permission, 12-7
- Modify permission, 12-7

## N

- New View Version permission, 12-8
- Nodes. See Workflow processes, node types
- Note templates, 3-16
- Notification lists, 19-2, 19-4–19-5
- Notification Policy Manager, 19-7
- Notification rules
  - combinations of, 19-5
  - creating, 19-2
  - creating and managing, 19-3
  - defined, 19-2
  - object types, 19-4–19-5
- Notifications
  - administrative, 8-26
  - defined, 19-2
  - deleted organizations, 8-24
  - deleted user-defined groups, 8-21
  - policies, defined, 19-2
  - process, 19-6–19-7
  - queue, 19-7
  - See also Workflow notifications
- NotifyPrincipalDisabled.xml, 8-26
- NotifyPrincipalRepair.xml, 8-26
- Null permission, 12-12
- NumberGenerator, 11-10

- Numbering schemes
  - changing, 11-21
  - configuring, 5-8
  - dynamic documents, 7-47
  - libraries, 7-6
  - out of the box, 3-35
  - overview, 3-35
  - products, 7-6
  - programs, 6-9
  - projects, 6-9

## O

- o attribute, 1-15, 2-5, 4-3
- Object initialization rules
  - accessing, 11-3
  - customizing, 11-20
  - definition, 15-3
  - description, 3-21
  - installation, 3-22
  - life cycle, 15-21, 15-47, 15-48
  - loading, 11-3
  - object type, 15-3
  - organization-level, 5-6
  - overview, 7-5, 11-2
  - site installed, 3-22
  - site-level, 4-6
  - specifying, 11-7
  - updating, 7-5, 11-2
  - versioning schemes, 15-25
  - XML, 11-8, A-69
- Object types
  - access control, 12-11, 12-15, 12-28
  - business data, 2-18
  - change, 9-15
  - defined, 12-3, 18-3, 19-2
  - indexing rules, 18-4
  - life-cycle managed, 12-24
  - team templates, 9-15
- OL files, 20-2
- One Phase Development template, 7-11, 15-15
- Optegra Gateway, 20-10
- Organization administrator
  - contexts, 3-3, 5-11
  - creating types, 13-18
  - definition, 1-12
  - getting started, 1-13
  - next steps, 1-17
  - responsibilities, 5-2
  - typical duties, 5-3
- Organization ID, 5-20, 5-21
- Organization ID Type, 5-20

- Organization Utilities page, 5-22
- OrganizationConfig element, 10-9, A-2
- Organizations
  - affiliation, 2-16
  - attribute, 2-5, 4-3, 8-17
  - contexts, 1-10, 3-4
  - creating, 5-17
  - deleting, 8-23
  - folders and documents, 5-5
  - groups, 5-4
  - managing, 8-22
  - members, 5-4
  - roles, 5-4
  - tab, 1-5, 1-7, 1-8, 1-10
  - templates, 3-15, 3-16, 5-10, 10-2, A-2
  - usage reporting, 21-15
  - viewing, 8-22
- OrganizationSync utility, 5-23
- OrgStructure element, A-59
- OrTest algorithm, 11-18
- Out-of-the-box
  - business object types, 2-18
  - constraints, 13-30
  - container participation, 7-14, 7-33
  - numbering schemes, 3-35
  - process images, 16-59
  - profiles, 8-32
  - translated context templates, 3-24
  - Windchill PDMLink library templates, 7-9
  - Windchill PDMLink product templates, 7-9
- Out-of-the-box templates
  - life cycle, 15-13
  - organization, 5-10
  - product and library, 7-9
  - project and program, 6-10
  - using, 3-24
- Workflow
  - Change Activity, 16-53
  - Change Notice, 16-55
  - Change Request, 16-54
  - Problem Report, 16-56
  - Promotion Request Approval Process, 16-57
  - Promotion Request Review Process, 16-58
  - Variance, 16-59
- OWNER pseudo-user, 12-11
- Owning organization principals
  - creating and updating, 5-20
  - defined, 5-19
  - domains, 5-20

## P

Part Library template, 7-10

Part to document relationships

- libraries, 7-48

- products, 7-48

- projects, 6-13

Part to part relationships

- libraries, 7-51

- products, 7-51

- projects, 6-16

Participants

- access control, 12-3

- assigning

  - to life cycle roles, 15-30

  - to team roles, 9-28

  - workflow activities, 16-23

- definition, 9-2

- searching, 8-8

- window, 15-30

PDM domain, 2-13

Permissions

- access control lists, 12-12

- ad hoc access control lists, 15-32

- calculating, 12-15

- create and modify attributes, 13-18

- create and modify types, 13-18

- create and modify units, 13-18

- create and modify workflows, 16-5

- defined, 12-4

- examples, 12-14–12-15

- setting, 12-6–12-8

- See also Access control, Access control rules

Personal cabinet names, 8-13

planMode element, A-29

PLT files, 20-2

Policies

- administering, 3-25

- definition, 3-25

- overview, 3-31

Policy ACLs. See Access control lists, policy

Policy Administrator, 3-25, 3-31

Policy rules, 3-31, 3-33, 12-2, 12-4

Post-installation activities, 2-25

Predefined reports

- introduction, 14-2

- Windchill PDMLink, 14-3

- Windchill ProjectLink, 14-3

Preference Manager, 3-38, 3-39, 15-9

PreferenceInstance element, A-66

Preferences

- context, 3-38

- hierarchy, 3-38

- libraries, 7-7

- organization-level, 5-9

- products, 7-7

- promotion process, 15-9

- setting, 3-39

- site-level, 4-5

- templates, 10-3, 10-4

- user, 3-38

Principal Administrator Utility

- accessing, 8-9

- managing organizations, 8-22

- managing user-defined groups, 8-18

- managing users, 8-12

- using, 1-16, 8-9

Principal cache, 8-13, 8-19, 8-23, 8-27

Principals

- access control, 12-3

- definition, 8-2, 9-2, 19-3

- overview, 8-2

- repair, 8-28

- searching, 8-8

- XML element, A-72

Pro/ENGINEER Wildfire, 2-3

Pro/INTRALINK 9.0

- getting started, 1-3

- home page, 1-9

Pro/INTRALINK Gateway, 20-10

Problem Report workflow, 16-56

Process Manager toolbar, 16-41

Processes

- change, 2-23

- life cycle, 2-23

- notification, 19-6–19-7

- workflow, 2-23

Product creators, 1-13, 5-11

Product Design template

- access control, 15-15

- contents, 7-11

- overview, 7-9

Product domains

- access control, 7-15, 7-34

- teams and, 9-16

Product manager

- access control rules, 7-26, 7-36

- creating, 1-13

- definition, 1-12

- next steps, 1-17

- role, 7-14, 7-33

- typical duties, 7-2

Product Utilities page, 7-46

ProductConfig element, 10-9, A-7

- Products
  - context, 3-2
  - creating, 7-42
  - tab, 1-4, 1-8, 1-9
  - templates, 3-15, 7-9, 10-3, A-7
- ProductStructure element, A-15
- ProductView, 2-20, 20-2, 20-3
- Profiles
  - context managers, 8-31
  - creating, 8-29
  - default behavior, 8-30
  - global default settings, 8-31
  - management, 8-29
  - organization, 5-8
  - out-of-the-box, 8-32
  - overriding, 8-31
  - users, 8-14
- Program creators, 1-14, 5-11
- Program managers, 6-2
  - creating, 1-14
  - definition, 1-12
  - next steps, 1-17
- Program team members, 6-4
- Program templates, 6-6, 6-10, 10-3, A-13
- Program Utilities page, 6-8
- Programs
  - context, 3-2
  - creating and editing, 6-3
  - introduction, 6-2
  - tab, 1-6
- Project creators, 1-14, 5-11
- Project managers, 1-14, 6-2
  - definition, 1-12
  - next steps, 1-17
  - typical duties, 6-2
- Project roles resolution, 9-18
- Project team members, 6-4
- Project templates, 3-15, 6-6, 6-10, 10-3, A-13
- Project Utilities page, 6-8
- ProjectConfig element, 10-9, A-13
- ProjectEnumType element, A-67
- ProjectLink. See Windchill ProjectLink
- projectPlanId element, A-29
- Projects
  - context, 3-2
  - creating and updating, 6-3
  - introduction, 6-2
  - tab, 1-6
  - templates, 10-3
- Promote transition rule, 15-7, 15-10
- Promotion Process preference, 15-9
- Promotion Request

- Approval Process, 15-8, 15-9, 16-57
- Review Process, 15-8, 15-9, 16-58
- subfolder, 7-14
- Promotion. See Life cycle promotion
- PromotionNotice, 2-19
- Proxy processes. See Workflow processes, subprocesses
- Publish rules, 20-23
- Publish Scheduler, 20-10
- Purge Manager, 4-10, 5-9
- Purging
  - jobs, 4-10
  - organizations, 8-23
  - user-defined groups, 8-19
  - users, 8-13
- PVS files, 20-2, 20-4
- PVZ files, 20-2, 20-4

## Q

- Query Builder, 14-4, 14-9
- Queues
  - notification, 19-7
  - visualization publish jobs, 20-5
  - Windchill Schedule, 20-10
- See also Windchill Queue Manager

## R

- Read permission, 12-6
- Related documentation, xliv
- Repair principals, 8-28
- Replication sites
  - configuring, 4-8
  - libraries, 7-8
  - products, 7-8
  - programs, 6-8
  - projects, 6-8
- Report Manager, 14-4
- Report templates, 3-15
- Reporting
  - Auditing Administration, 14-6
  - Change Monitor, 14-5
  - license usage, 21-14
  - organization usage, 21-15
  - related reports for products, 14-4
  - security audit, 21-2
  - Windchill Business Reporting, 14-4

- Reports
  - authoring, 14-9
  - editing, 14-10
  - monitoring, 14-11
  - searching, 14-11
- Reports page, 14-2
- Resources, creating and editing, 6-5
- Restoring jobs, 4-10, 5-9
- Revise permission, 12-7
- Revise transition rule, 15-7, 15-10
- Robots. See Workflow processes, robot types
- RoleRB.rbinf resource file, 9-27, 15-29, 15-43
- Roles
  - attribute administrators, 13-18
  - definition, 9-2
  - program, 6-4
  - project, 6-4
  - type administrators, 13-18
  - workflow administrator, 16-5
- Root domain, 3-10, 3-29, 3-30, 11-25, 12-23
- Root view, 17-2
- Routing events, 16-26
- Rule antecedents
  - access control lists, 12-10
  - indexing, 18-2
  - notifications, 19-2
- Rule consequents
  - access control lists, 12-10
  - indexing, 18-2
  - notifications, 19-2
- Rules. See Access control rules, Notification rules
- Runtime Services, 2-3

## S

- Schedule Publish Job, 20-11
- Searches
  - managing, 4-10
- Searching
  - deleted organizations, 8-24
  - deleted users, 8-16, 8-22
  - indexes, 18-2
  - organizations, 8-23
  - saved searches, 3-40
  - user-defined groups, 8-18
  - users, 8-12
- Selecting, 16-42
- SessionIterationDomain domain, 3-29
- Set State permission, 12-7
- Set State transition rule, 15-7, 15-10

- Setting permissions, 12-6–12-8
- Shared team, 2-6, 9-6
- Shared team manager
  - definition, 1-12
  - next steps, 1-17
  - products and libraries, 1-13
  - projects and programs, 1-14
- Shared teams
  - definition, 9-3
  - organizations, 5-5
- SharedTeamDef element, A-48
- Site
  - container domains, 3-29
  - context, 3-4, 3-23
  - document templates, 3-18
  - library templates, 3-18
  - product templates, 3-18
  - program templates, 3-18
  - project templates, 3-18
  - report templates, 3-19
  - tab, 1-5, 1-7, 1-8, 1-10
  - task form templates, 3-20
  - workflow templates, 3-17
- Site administrator
  - contexts, 3-3
  - create types, attributes, and units, 13-18
  - establishing, 1-12
  - next steps, 1-17
  - responsibilities, 4-2
  - site and organization tabs, 1-12
  - typical duties, 4-2
  - workflow security, 16-3
- Soft types. See Types
- State property, 15-24
- State-based versioning, 11-31
- StateRB.rbinf resource file, 15-43
- StringConstant, 11-10
- StringEqualsTest, 11-17
- StringInListTest, 11-18
- StringRegExEqualsTest, 11-17
- Subfolders
  - defined, 12-21
  - domains, 12-21
  - moving, 12-21
- Subscriptions, 19-5
- System administrators, 1-12
- System domain, 2-9, 3-13, 3-29, 11-25
- System groups, 9-9
- System information, auditing, 4-7

## T

TAG-0.xml file, 10-6

TAG-ContainerTemplate XML file, 10-6

Task form templates

- about, 3-16

- creating, 16-71

- HTML, 16-71

- PDF, 16-71

- workflow, 16-71

Team roles

- creating, 9-26

- defined, 9-2

- products and libraries, 7-4

- See also Life cycle roles, Participants

Team templates

- Change Activity Team, 9-13

- Change Request Team, 9-14

- context, 3-15

- Default, 9-15

- definition, 9-2

- installation, 3-17

- out-of-the-box, 9-13

- Problem Report Team, 9-13

Teams

- association, 9-16

- association rules, 9-16

- chapter, 9-1

- context, 2-15, 9-3

- creating, 9-26

- libraries, 7-45

- library, 10-3

- local, 2-15, 9-4

- overview, 9-2

- products, 7-45

- roles resolution, 9-17

- shared, 2-15, 9-6

- See also Team templates

TeamTemplateAttributeAlgorithm, 11-10, 11-12

Technical support, xlvii

Templates

- creating, 10-4

- description, 3-15

- document, 3-15, 3-18

- enabling, 10-11

- exporting, 10-4

- importing, 10-5

- installation, 3-16

- JSP, 16-71

- library, 7-9, 7-33, A-10

- object initialization rules, 11-20

- organization, 5-6, 5-10, 10-2, A-2

- product, 7-9, 7-33, 10-3, A-7

- Product Design, 7-9

- program, 10-3, A-13

- project, 10-3, A-13

- removing, hiding, disabling, 3-20

- site-level, 4-6

- task form, 16-71

- translated, 3-24

- updating, 3-20

- visibility, 10-3, 10-4, 10-11

- XDP, 16-71

This Org, 5-2

Thumbnail generator, 20-4

Thumbnail images, 20-4

Transitions property, 15-24

Transitions, for workflow activities, 16-27

Translated context templates, 3-24

Tutorials, customizing, 2-3

Two Phase Development

- state-based versioning, 11-34

- template, 7-11, 15-15

Type and Attribute Manager Utility

- Attribute Definition Manager Tab, 13-9, 13-12

- glossary, 13-51

- Measurement System Manager Tab, 13-9, 13-15

- overview, 13-7

- Quantity of Measure Manager Tab, 13-9, 13-16

- Type Manager Tab, 13-9, 13-10

- using, 13-19

TypeBasedRule element, A-69

Types

- CAD documents, 13-39

- change objects, 13-36

- creating, 13-19

- See also Permissions

- deleting, 13-30

- display lists, 13-37

- documents, 13-36

- dynamic documents, 13-43

- editing, 13-28

- enumerated, 9-27

- excluding from search pages, 13-38

- localizing soft types, 13-6

- logical identifier, 13-35

- organization, 5-5

- parts, 13-36

- preferences for default types, 13-36

- searching, 13-51

- site, 4-5

- soft, 2-20

- specifying, 13-35

## U

- UFID, 8-2, 8-24
- Unaffiliated domain, 2-17, 3-12, 3-29
- Unique Federation Identifier. See UFID
- Units
  - supported, 13-17
- Updating
  - life cycles, 15-17
  - types, 13-28
  - workflow process templates, 16-8
- User domain, 2-16, 3-12, 3-29, 12-23
- User ID, 8-12
- User name, 8-12, 9-28
- Users
  - creating, 8-12, 12-23
  - deleting, 8-15
  - editing domain, 8-15
  - end users, 1-15
  - establishing, 1-15
  - managing, 2-5, 2-16, 8-12
  - notifying. See Notifications
  - objects, 8-3
  - organization affiliation, 2-16, 8-17
  - profiles, 8-14
  - setting permissions, 12-8
  - undoing checkout, 5-9, 6-7, 7-8

## V

- Variance workflow, 16-59
- VariantSpec, 2-19
- Version Series property, 15-24
- VersionInfoGenerator, 11-10
- Versioning schemes
  - administering, 3-36
  - configuring, 5-8
  - file-based, 11-29
  - Harvard series, 11-25
  - setup, 11-34
  - state-based, 11-31
- Versioning XML file, 11-35
- View Administrator Utility, 17-2
- Viewable publishing, 5-8
- Viewing
  - organizations, 8-22, 8-23
  - user-defined groups, 8-18
  - users, 8-12

- Views and view associations
  - creating, 17-2
  - managing, 17-3
  - view-dependent objects, 17-2
- Visualization collaboration interface, 20-12
- Visualization data model, 20-4
- Visualization services
  - See also Thumbnail generator, Thumbnail images
  - architecture, 20-3
  - data, 2-20
  - file types, 20-2
  - loader, 20-3, 20-6
  - properties, 20-45
  - Publish Scheduler, 20-10
  - troubleshooting, 20-6

## W

- Watermarks, 20-13, 20-18
- Windchill Business Report Author, 14-10
- Windchill Business Report Monitor, 14-11
- Windchill Business Reporting, 14-4
- Windchill environment, 2-3
- Windchill PDMLink
  - administrators, 1-13
  - container hierarchy, 2-7
  - getting started, 1-2
  - home page, 1-4
  - templates, 10-3
- Windchill ProjectLink
  - administrators, 1-14
  - container hierarchy, 2-7
  - getting started, 1-2
  - home page, 1-6
- Windchill Queue Manager, 20-10
- Windchill Runtime Architecture, 2-2
- Windchill Units, 13-17
- Windchill URL, 1-3
- Windchill Visualization Services. See Visualization services
- Windows
  - Administrative Domain, 12-5
  - Participants, 15-30
  - Team Administrator, 9-27
  - Update Life Cycle, 15-20, 15-26

- Workflow
  - definition, 16-2
  - history
    - generated events, 16-42
    - overview, 16-2
    - viewing, 16-42
  - instance states, 16-49
  - managing security, 4-7, 16-3
    - administrative groups, 16-5
    - restricting embedded java code, 16-5
    - workflow creators, 16-4
  - overview of components, 16-2
  - requiring authentication, 16-7
- Workflow activities
  - access control, 12-24
  - assigned activity, 16-19
  - assigning participants, 16-23
  - creating, 16-19
  - deadlines, 16-24
  - declaring variables, 16-25
  - defining, 16-19
  - defining teams, 16-22
  - defining transitions, 16-27
  - routing, 16-26
  - set up participants, 16-22
- Workflow Administrator Utility
  - accessing, 16-3
  - importing and exporting, 16-40
  - working with templates, 16-9
- Workflow Process Editor
  - connector types, 16-35
  - navigating diagrams, 16-12
  - overview, 16-2
  - process nodes, 16-12
  - using, 16-8
- Workflow Process Manager
  - overview, 16-2
  - toolbar, 16-41
- Workflow processes
  - associating, with life cycle phases and gates, 15-36
  - declaring variables, 16-16
  - defining connectors, 16-35
  - defining links, 16-37
  - deleted user-defined groups, 8-21
  - node types, 16-12
  - notifications, 19-2
  - robot types, 16-14
  - subprocesses, 16-34
- Workflow routing. See Workflow activities, routing
- Workflow runtime system, 16-2
- Workflow templates
  - checking out, 16-8
  - context, 3-15
  - deleted user-defined groups, 8-21
  - deleted users, 8-16
  - execution flags, 16-60
  - importing and exporting, 16-39
  - installation, 3-17
  - iterations, 16-8
  - See also Out-of-the-box templates
  - updating, 16-8, 16-11
- Workgroup Manager CAD Document soft type, 2-21
- wt.epm.EPMDocument, 2-21
- wt.principal.cache properties
  - .timeToLive, 8-27
  - .timeToLiveRandomizer, 8-27
- wt.properties file
  - collections, 18-5
  - delegating tasks, 4-10
  - domain names, 3-29
- wt.typepickerexclude, 13-37
- WTAnalysisActivity, 2-18
- WTChangeActivity2, 2-18
- WTChangeInvestigation, 2-18
- WTChangeIssue, 2-18
- WTChangeOrder2, 2-18
- WTChangeProposal, 2-18
- WTChangeRequest2, 2-18
- WTDocument, 2-18, 12-2, 12-11
- WTObject
  - access control rules, 12-23
  - cabinets, 12-28
  - notification list, 19-4
  - soft type, 12-15
- WTPart, 2-19, 20-4, 20-5
- WTPrincipalReference element, A-72
- WTProductInstance2, 2-19
- WTVariance, 2-19
- wvs.properties file, 20-6, 20-12

## X

- xconfmanager, 2-28
- XML element (Arbortext), 13-44
- XML template files, A-1
- XML versioning file, 11-35

## Z

- ZIP files
  - exporting, 10-6
  - importing, 15-41
  - life cycle, 15-40