# Agile Product Lifecycle Management

## Administrator Guide

v9.3.0.3

Part No. E23034-02
July 2011

# Oracle Copyright

*Copyright © 1995, 2011, Oracle and/or its affiliates. All rights reserved.*

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services. The RMW product includes software developed by the Visigoth Software Society.

# CONTENTS

Oracle Copyright ..... ii

**Chapter 1 ..... 1**

**Introduction ..... 1**

Welcome to Agile PLM Administrator ..... 1
- How to Find Documentation for Agile PLM ..... 1
- Documentation for the PLM Administrator ..... 1
- Getting Started with Agile PLM ..... 1
- PLM Solution-Specific User Guides ..... 2

What's New in Administrator ..... 3
- Features and Enhancements added in Rel. 9.3 ..... 3
- Installation Test Notice ..... 3

**Chapter 2 ..... 5**

**Getting Started in Administrator ..... 5**

Overview ..... 5
Logging in to Agile Clients as an Administrator ..... 5
Creating New Administrator Users ..... 6
Dates and Times in Agile PLM ..... 8
- ...at the Server Level ..... 8
- ...at the Client Level ..... 8
- Example ..... 8

Administrator User Interfaces ..... 8
- Filtering Data in Java Client ..... 9
- Administrator Nodes in Java Client ..... 9
- Administration in Web Client ..... 14

Agile PLM Variables ..... 14
- Default Value Variables ..... 15
- Privilege Mask Criteria Variables ..... 16
- Workflow Assignment and Status Type Variables ..... 17
- Routing Manager Variables ..... 17

**Chapter 3 ..... 19**

**Administrator Functions in Java Client ..... 19**

API Name ..... 19

API Name field in Java Client ..... 19
Administrator Reports ..... 20
Utilities on Admin Toolbar ..... 21
Sending Email to Users ..... 22
Global Replace ..... 22
History Tracking ..... 24
Object History and Administrative Object History ..... 24
Administrator History ..... 25
History of Deleted Objects ..... 26
Deleting Objects in Agile ..... 26
Administrator Import and Export ..... 27
Encoding Types ..... 28
Administrator Export ..... 28
Administrator Import ..... 29
Process Extensions Library ..... 29
Multiple Language Support ..... 30
Administrator Data that is Supported for Multi-Language Capability ..... 31
Administrator Data that is Not Supported for Multi-Language Capability ..... 32
Configuring a System for Multi-Languages ..... 33

**Chapter 4 ..... 35**

**Administrator Functions in Web Client ..... 35**

Overview to Administrator in Web Client ..... 35
Configuring the Dashboard ..... 36
Visual Themes ..... 36
Selecting Theme Options ..... 36
Modifying Style Schemes ..... 38
Report Templates ..... 40
Manage Report Template (AppliedTo Choice) ..... 40
Dynamic and Static Templates ..... 40
Creating Report Templates ..... 41
Reports Unsupported by BI Publisher ..... 41
Logging Configuration ..... 42
Cache Health Monitor ..... 43
How the Cache Health Monitor Works ..... 43
Automatic Synchronization using Java Client Administrator ..... 44
Logging and Debugging Errors ..... 45

Manual Triggering: 46
Automatic Triggering: 46
Deleting Objects in Agile 47

**Chapter 5 49**

**Classes 49**

Configuring Business Objects with Agile PLM Classes 49
Agile PLM Class Structure 50
Routable and Nonroutable Objects 51
Object Types in Agile PLM 51
Viewing Agile PLM Classes 51
Agile PLM's Installed Routable Object Types 52
Agile PLM's Installed Nonroutable Object Types 54
Disabling Object Types 56
Customizing Lifecycle Phases 56
Modifying a Lifecycle Phase 56
Adding a Lifecycle Phase 57
Removing a Lifecycle Phase 57
Configuring Subclasses 58
Before You Begin 59
Viewing Subclass Properties 59
Changing the AutoNumber Source Property 59
Preventing Manual Number Assignment 60
Changing a Subclass Name or Description 60
Changing the Site-Specific BOM Property 61
Creating a New Subclass 61
Deleting a Subclass 62
Configuring Tabs 63
About Page Two Tabs 63
About Page Three Tabs 63
Showing, Hiding, or Renaming a Tab 64
Modifying the Format of Tabs 64
Modifying Attribute Order on a Form Tab 65
Modifying Attribute Order on a Table Tab 66
Designing a Page Two or Page Three Tab 66
Guidelines for Designing Page Two and Page Three Tabs 66
About Attributes, Flex Fields, and Read-Through Fields 70

User-Defined Flex Fields ..... 71
Read-Through Fields ..... 73
Defining Attributes ..... 75
Recognizing Attribute Types ..... 76
Defining Attribute Properties ..... 77
Configuring Attributes ..... 82
Modifying Attributes ..... 82
Matching the Properties of Attributes Common to Several Tabs ..... 83
Setting Default Values ..... 83
Showing or Hiding an Attribute on a Tab ..... 84
Editing a Field Label or Table Column Name ..... 84
Using Custom Icons for Subclasses ..... 84
Quick Access to Objects and Files ..... 85
Agile-Generated Quick Access Object URLs ..... 86
Formats for User-Generated Smart Object URLs ..... 86
Using Quick Access Object URLs ..... 87

**Chapter 6 ..... 89**

**Character Sets ..... 89**

Available Character Sets ..... 89
Creating a Character Set ..... 90
Modifying a Character Set ..... 91

**Chapter 7 ..... 93**

**Lists ..... 93**

Overview ..... 93
Routing Manager Lists ..... 93
Creating Custom Lists and Adding to Lists ..... 94
More Kinds of Lists ..... 95
SingleList Lists ..... 96
MultiList Lists ..... 96
Cascading Lists ..... 96
Dynamic Lists ..... 97
Configuring Lists ..... 97
Setting Default Values for Lists ..... 99
Failure Mode Lists ..... 100
Creating Cascading Lists ..... 100

**Chapter 8 ..... 103**

**Process Extensions ..... 103**

Overview to Process Extensions ..... 103
Using the Process Extension Library ..... 104
Assigning Process Extensions to Classes ..... 106
Assigning Process Extensions to Workflow Statuses ..... 106

**Chapter 9 ..... 109**

**Autonumbers ..... 109**

How Autonumbers Are Used ..... 109
Modifying an Autonumber Source ..... 109
About Autonumber Sources ..... 110
Configuring Autonumber Sources ..... 112
Changing an Autonumber's Prefix or Suffix ..... 113
Changing Where an Autonumber Source Is Used ..... 113
Custom Autonumbers ..... 114
Timeout ..... 114
Creating a New Autonumber Source ..... 114
Before You Begin ..... 114
Deleting an Autonumber Source ..... 115

**Chapter 10 ..... 117**

**Criteria ..... 117**

About Reusable Criteria ..... 117
Exploring Reusable Criteria ..... 118
Viewing Reusable Criteria ..... 118
Criteria Windows and Tabs ..... 119
Creating a New Criteria ..... 119
Modifying an Existing Reusable Criteria ..... 121
Using Affected Items Tab Fields in Reusable Criteria ..... 121
How Changes Are Matched When Reusable Criteria Specify Affected Items Tab Fields 121
Difference Between "Item" and "Part" in Affected Items Names ..... 122
More about Affected Items: Adding Approvers and Observers ..... 123
Using $CURRENTREV and $LATESTREV in Criteria for Items ..... 123
About TitleBlock.Rev ..... 124
Rules for Using the $CURRENTREV Variable ..... 124

Rules for Using the $LATESTREV Variable ..... 125
Item Relationship Creation Using $CURRENTREV and $LATESTREV Criteria ..... 126
Using File Folder Attributes in Reusable Criteria ..... 126

**Chapter 11 ..... 129**

**Workflows ..... 129**

How Workflows Automate the Change Control Process ..... 129
Modifying a Workflow ..... 130
Workflow Basics ..... 130
Routing Managers ..... 130
Other Workflow Basics ..... 131
Keep It Simple! ..... 132
Default Status-Based Workflows ..... 133
Status Types Defined ..... 133
Creating New Workflows with Save As ..... 135
General Workflow Properties ..... 136
General Information Tab ..... 136
About Workflow name ..... 137
Workflow Criteria Matching Type ..... 137
Status Criteria Matching Type ..... 138
Creating and Using Custom Workflows ..... 139
Before You Begin ..... 140
Custom Workflow Example ..... 140
Guidelines for Creating Workflows ..... 141
Creating a New Workflow ..... 142
When a New Workflow is Enabled ..... 144
Routing to a User Group ..... 144
Status Properties Defined ..... 145
If No Criteria Apply At Exit, Notify ..... 145
Manual Valid Next Status ..... 145
AutoPromote ..... 146
If AutoPromote Fails, Notify ..... 147
Ad Hoc Approvers/Observers ..... 148
If Rejected, Set Status To ..... 148
If Rejected, Notify ..... 148
Reminder Period ..... 149
Review Escalation Period ..... 149

Comments for Approval ..... 150
Comments for Rejection ..... 151
Criteria-Specific Properties Defined ..... 151
Approvers and Observers ..... 151
Criteria Usage ..... 151
Default Change Analyst / Component Engineer ..... 152
Notify Upon Entry ..... 152
Pass Release Audit ..... 153
Exit Required Fields ..... 155
Workflow Functionality Testing ..... 156
Areas of Workflow Validation ..... 156
Quick Tips on Testing ..... 157
Migrating Workflows to Production ..... 157
Quick Tip on Migrating Workflows ..... 157
Workflow Migration Process ..... 157
Changing Active Workflows ..... 158
Modifying Properties in an Enabled Workflow ..... 158
Disabling, Changing, and Reactivating Workflows ..... 160

**Chapter 12 ..... 161**

**Account Policy ..... 161**

Before You Begin ..... 161
Configuring the Account Policy ..... 161
Administrator Reset of Locked Out Account ..... 163

**Chapter 13 ..... 165**

**Users ..... 165**

Adding New Users ..... 165
Before You Begin ..... 166
About Usernames ..... 167
About Passwords ..... 167
Change Internal User Password ..... 168
LDAP as a Node in Administrator ..... 168
Modifying User Properties ..... 169
Changing a User's Password ..... 169
Changing a User's Role Assignments ..... 170
User Properties Defined ..... 170

**Chapter 14 ..... 177**

**User Groups ..... 177**

Creating a New User Group ..... 177
User Group Properties ..... 179
Modifying a User Group ..... 180
Assigning Escalation Persons to User Groups ..... 180
Adding User Groups as Approvers of a Change ..... 181
Setting Up User Groups as Partners ..... 183

**Chapter 15 ..... 185**

**Supplier Groups ..... 185**

Overview ..... 185
Properties ..... 185
  Available actions ..... 185

**Chapter 16 ..... 187**

**Roles ..... 187**

Overview of Roles and Privileges in Agile PLM ..... 187
Getting Familiar with Roles ..... 188
  Viewing the Roles Window ..... 189
  Viewing a Role ..... 189
  Guidelines for Working with Roles ..... 190
Default Agile PLM Roles ..... 190
  Example Roles ..... 196
  Restricted Roles ..... 196
  Role-based Enhancements ..... 196
Securing and Maintaining Roles and Privilege Masks ..... 197
  Using Agile Configuration Propagation to Restore Admin Settings ..... 197
  Recommended Security and Maintenance Plan ..... 198
  Database Backup Procedures ..... 198
Working with Roles ..... 200
  Modifying a Role ..... 200
  Building and Testing Attributes, Read/Modify Privileges, and Roles ..... 202
  Creating a New  Role ..... 202
  Using the Example Roles ..... 205
  Deleting a Role ..... 206

**Chapter 17 ..... 207**

**Privileges and Privilege Masks..... 207**

About Privileges and Privilege Masks .....207
Getting Familiar with Privileges and Privilege Masks .....207
Viewing Privilege Masks .....208
Tasks Performed from the Privileges Window .....210
Default Agile PLM Privileges .....211
Further Discussion of Specific Privileges .....216
More Information about Selected Privileges .....217
Privilege Mask Components .....221
All Privilege Masks have a Privilege, an Object Type, and a Criteria .....222
Privilege Mask Component 1: Privileges .....224
Privilege Mask Component 2: Object Type .....225
Privilege Mask Component 3: Criteria .....226
Privilege Mask Component 4: Attributes .....226
Enabling a Privilege Mask .....227
Modifying Privilege Masks .....228
Modifying Privilege Mask Criteria .....228
AppliedTo Property in Select Privileges .....228
Creating New Privilege Masks .....233
Before You Begin .....233
Copying a Privilege Mask Using Save As .....234
Creating a New Privilege Mask from Scratch .....234
Combining Options in Criteria and AppliedTo Attributes .....235
Discovery and Read Privileges .....236
Discovery Privilege .....237
Relationships among Discovery and Read Privileges .....238
Field-Level Read Privilege .....239
Enforce Field-Level Read Privilege .....240
Modify Privilege .....242
Some 'Modify' Basics and Rules .....242
Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV .....243
Applying Create User Criteria .....246
Using Privilege Mask Criteria Variables .....247

**Chapter 18 ..... 249**

**Other User Settings ..... 249**

User Monitor ..... 249
Refreshing the User Monitor Window ..... 249
Terminating a User Session ..... 249
Deleting and Undeleting Users ..... 250
Deleting a User Group ..... 250

**Chapter 19 ..... 253**

**SmartRules ..... 253**

Modifying SmartRules Settings ..... 253
SmartRules Defined ..... 254
Allow Deletion of Attachments ..... 254
Auto-Installs from PPM ..... 254
Auto Publish Quote History ..... 254
BOM Multi-Level Recursion ..... 254
Change Status Approver Rejected Change ..... 255
Change Status No Response Change ..... 255
Change Status Observer Rejected Change ..... 255
Check In Attachment With Different File Extension ..... 255
Copy Files to Rev ..... 256
Demotion Approvers Removal ..... 256
Demotion Signoff Removal ..... 257
Detailed TimeSheet Entry ..... 257
Discovery Object Tables ..... 257
Discovery Results Table ..... 257
Display BOM Tables ..... 258
Display Structure Tables ..... 258
Duplicate Find Numbers ..... 258
Duplicate Ref Des ..... 259
Effectivity Date Order ..... 259
Effectivity Gap ..... 259
Effectivity Overlap ..... 259
Enable Addition of Activities and Gates to Completed Activities ..... 259
Enable Comments for Deleting, Archiving & Delegating ..... 260
Force Commodity and Part Family to be Identical ..... 260

Items Released First ..... 260
Many Items per PSR ..... 260
Many QCR per PSR ..... 260
Multiple Items Per Manuf Part ..... 261
Negative Value For Material Price Adder Fields ..... 261
Negative Value For Material Price Fields ..... 261
Negative Value For Non-Material Price Fields ..... 261
Notify Resource Pool Owner for Proposed Programs ..... 261
Overlap Price Line Effectivity Periods ..... 261
PSR Contains Items and Related PSRs ..... 261
Redline Attributes ..... 262
Redline Manufacturers On Change Order ..... 262
Release QCR With Un-resolved Items ..... 262
Released Rev Required ..... 262
Unrelease Change Order ..... 262
Unrelease Change Request ..... 262
Unrelease Deviation ..... 263
Unrelease Manufacturing Order ..... 263
Unrelease Stop Ship ..... 263
Workflow Matching Criteria Attribute Modification ..... 263
Zero Value For Material Price Fields ..... 263
Zero Value For Material Price Adder Fields ..... 263
Zero Value For Non-Material Price Fields ..... 264

**Chapter 20 ..... 265**

**Viewer and Files ..... 265**

Introduction ..... 265
Viewer & Files Properties ..... 265
Options for Purging and Retaining Files ..... 266
AutoPurge Enabled ..... 266
File Version Keep ..... 267
Purge Redlined Files ..... 267
Directly Use These Thumbnail Image Types ..... 267
CAD Revision BOM Attribute ..... 268
File Association tab ..... 269
Specifying Supported File Types for AutoVue for Agile ..... 269
Printing Banners and Watermarks ..... 271

Banner and Watermark Properties ..... 273
Specifying Banner Text and Location ..... 274

**Chapter 21 ..... 275**

**Notifications ..... 275**

Overview of Notifications ..... 275
Preliminary Admin Settings for Notifications ..... 275
Default Notifications ..... 276
Attributes of Default Notifications ..... 276
Event-based Notifications ..... 277
Attributes of Event-based Notifications ..... 278
Working with Events-based Notifications ..... 279
Invoking Notifications Programmatically ..... 280
Working with Default and Event-based Notifications ..... 281
Editing a Notification ..... 281
Entering Subject and Body Text ..... 281
Creating a Notification using SaveAs ..... 282

**Chapter 22 ..... 283**

**Full Text Search ..... 283**

Full Text Search Limitations ..... 283
Full Text Search Properties ..... 284
Indexing Attachment Files ..... 285
Using FileLoad and Indexing New Files ..... 285
Index Attachment Button ..... 285
"Stop" Words ..... 285

**Chapter 23 ..... 287**

**My Assignments ..... 287**

Overview ..... 287
Columns ..... 287

**Chapter 24 ..... 289**

**Unit of Measure (UOM) ..... 289**

Creating Categories of Measure and Units of Measure ..... 289

**Chapter 25 ..... 291**

**Other System Settings ..... 291**

Company Profile ..... 291
Currency Exchange Rates ..... 291

**Chapter 26 ..... 293**

**Dashboard Management ..... 293**

View Dashboard Tabs ..... 293
Adding Dashboard Tabs ..... 293
Deleting Dashboard Tabs ..... 293
Adding Tables to Optional Tabs ..... 294
  Adding Data to Tables ..... 294
Adding Charts to Optional Tabs ..... 294
  Configure Table for Advanced Search ..... 296
Adding Custom (URL) Process Extensions for Optional Tables ..... 297
Reordering Table Rows in a Tab ..... 297
Displaying Optional Tabs ..... 297

**Chapter 27 ..... 299**

**Solution-specific Administration ..... 299**

Documentation ..... 299
Configurable solutions for Administrator ..... 299

**Chapter 28 ..... 301**

**Event Management ..... 301**

Introduction to Event Management ..... 301
  Preliminary Configuration ..... 301
  Event Management Node Folder ..... 302
Anatomy of Event Subscriptions ..... 303
  Event Subscription Components are Interconnected ..... 305
  Event Subscribers Tab on All Object Types ..... 306
Handler Types ..... 307
  Process Extension–based Handler Types ..... 307
  Notification-based Handler Type ..... 308
  Notes on Event Handler Types Node ..... 308
Event Types ..... 308

Default Event Types ..... 309
Working in Events Node ..... 312
Attributes of Event Masks ..... 312
Checklist for Creating an Event Mask ..... 314
Creating an Event Mask ..... 314
Working in Event Handlers Node ..... 315
Attributes of Handler Masks ..... 316
Checklist for Creating a Handler Mask ..... 317
Creating a Handler Mask ..... 317
Working in Event Subscribers Node ..... 318
Filtering in Dialogs during Create Subscriber ..... 318
Attributes of Subscriber Masks ..... 319
Checklist for Creating a Subscriber Mask ..... 321
Creating a Subscriber Mask ..... 322
Sample Event Subscriptions ..... 323
Monitoring Events ..... 323
Event Handler Monitor ..... 324
Event Log File ..... 325
Object History Tab ..... 325
Control of Event Components ..... 325
Global Event Control ..... 326
When an Event Type is Disabled ..... 326
When a Handler Type is Disabled ..... 326
When an Event Mask is Disabled ..... 327
When a Handler Mask is Disabled ..... 327
When a Subscriber Mask is Disabled ..... 327
More about Selected Event Types ..... 327
More about Selected Global Event Types ..... 327
More about Selected Generic Object-based Event Types ..... 328
More about Selected Specific Object-based Event Types ..... 331
More about Workflow-based Event Types ..... 331

**Chapter 29 ..... 335**

**Scripting and Sample Event Subscriptions ..... 335**

Overview to Scripting in Agile PLM ..... 335
Using Groovy Scripting Language ..... 336
Choosing Script-based Handlers ..... 336

Steps to Develop or Update Handler Masks ..... 336
Editing Scripts in the Handler Mask ..... 337
Logging of Scripts ..... 337
Checklist for Specifying Script-based Event Subscriptions ..... 338
Sample Event Subscriptions with Scripts ..... 338
Update Item Description ..... 339
Notify Create User when Item is Incorporated ..... 340
Set Change Analyst ..... 341
Add Document to New Part ..... 343
Notify when Affected Items Table is Updated ..... 345

**Chapter 30 ..... 347**

**Locations and File Management ..... 347**

Configuring Web Client Notification ..... 347
Configuring Java Client Notification ..... 348
File Management in Agile PLM ..... 348
Agile File Management Server ..... 348
Distributed File Management ..... 349
File Management Security ..... 350
Configuring a File Manager ..... 352
Setting the Preferred File Manager for a User ..... 355
Purging Older Versions of Files ..... 356

**Chapter 31 ..... 357**

**Database ..... 357**

Configuration of Database Settings ..... 357
Database Properties ..... 357

**Chapter 32 ..... 359**

**LDAP ..... 359**

Configuring WebLogic Server for LDAP with Agile PLM ..... 360
Configuring WebLogic with Oracle Internet Directory ..... 360
Configuring WebLogic with Sun Java System Directory Server ..... 362
Configuring WebLogic with Microsoft Active Directory Server ..... 363
Creating Users in an LDAP Directory ..... 365
Configuring Multiple LDAP User Repositories ..... 366
Enhanced LDAP ..... 366

Synchronizing LDAP and Agile PLM....366
LDAP-controlled User Properties....367
Agile LDAP Configuration....367
Directory Service Connection Parameters....368
Configuration Parameters Defined....368
Multiple Directory Server Support....370
Configuration Scripts....370
Synchronizing Users and User Groups....370
**Chapter 33 .... 371**
**Systemwide Preferences .... 371**
Modifying Preferences....371
Default Systemwide Preferences....371
Details of Selected Preferences....377
Agile Drive....377
Allow Download of Productivity Components....377
Allow Password Reset....377
Copy Values along with Tabs....378
Content Service Wait Time....378
Content Service Maximum Retries....378
Reference Designators Allow Range Expand Collapse....378
Signoff User Dual Identification....378
Languages tab....379
**Licenses .... 380**
General Info Tab....381
Modules Tab....381
**Chapter 34 .... 383**
**Task Monitor and Task Configuration .... 383**
Task Monitor....383
Task Configuration....384
**Chapter 35 .... 387**
**Administering Attachments .... 387**
Evaluating Your Company's Needs with Attachments or Designs....387
Administrator Settings Related to Attachments....387
File Management in Agile PLM....387

How SmartRules affect Attachments....388
How Preferences affect Attachments....389
Assign Roles and Privileges for Attachments....392
Attachment Privileges....392
Markup Subclass....396
Redline Markup Default Roles and Privileges....396
Document Management Enhancements....398
View All Versions....398
Limit Attachment Size....398
Identify Checkin User....399
URLs can be up to 255 characters....399
Generating Attachment Quick Access URLs....399
Agile-Generated Attachment Quick Access URLs....399
User-Generated Attachment Quick Access Smart URL Formats....400
Accessing Attachment Files using Quick Access URLs....403

**Chapter 36 .... 405**

**Administering Designs.... 405**

Evaluating Your Company's Needs with Designs....405
Purpose of Design Data Model....405
Benefits of Designing Structures using the Designs Class....406
Designs Class Overview....406
View All Versions....407
Designs Class Tabs....407
Title Block Tab....407
Files Tab....408
Structure Tab....409
Routing Slip Tab....409
Where Used Tab....409
Administrator Settings Related to Designs....409
Display Structure Tables SmartRule....409
Design-specific Role & Privilege....410

**Chapter 37 .... 411**

**Administering Agile Drive .... 411**

Overview....411
File Management....411

Enabling Agile Drive ..... 411

**Appendix A ..... 413**

**Configuring Single Sign-On ..... 413**

Overview of Single Sign-On in PLM ..... 413
Configuring and Deploying SSO with Windows NTLM ..... 414
SSO NTLM Authentication ..... 414
Configure SSO for Windows NTLM ..... 414
Deploy NTLM ..... 416

# Preface

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

| | |
|---|---|
| **Note** | To read the PDF files, you must use the free Adobe Acrobat Reader version 9.0 or later. This program can be downloaded from the Adobe Web site http://www.adobe.com. |

The Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html can be accessed through **Help** > **Manuals** in both Agile Web Client and Agile Java Client. If you need additional assistance or information, please contact My Oracle Support (https://support.oracle.com) for assistance.

| | |
|---|---|
| **Note** | Before calling Oracle Support about a problem with an Agile PLM manual, please have the full part number, which is located on the title page. |

## TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

## Readme

Any last-minute information about Agile PLM can be found in the Readme file on the Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html.

## Agile Training Aids

Go to the Oracle University Web page http://www.oracle.com/education/chooser/selectcountry_new.html for more information on Agile Training offerings.

## Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

# Chapter 1

# Introduction

**This chapter includes the following:**


- Welcome to Agile PLM Administrator ..... 1
- What's New in Administrator ..... 3


# Welcome to Agile PLM Administrator

Welcome to Oracle's *Agile PLM Administrator Guide* for Agile Product Lifecycle Management (Agile PLM, or PLM).

## How to Find Documentation for Agile PLM

The Agile PLM documentation set is found at this Web site:
http://www.oracle.com/technology/documentation/agile-085940.html.

## Documentation for the PLM Administrator

This manual is the primary handbook for Agile PLM administrators. It is a comprehensive reference for setting up users and business objects in PLM. The *Agile PLM Administrator Guide* documents most of the nodes in the Administrator user interfaces ("Admin" UI specifics are discussed in the next chapter).

However, the administrator must access other manuals in the Agile PLM documentation set for many situations, as listed below.

### Agile Configuration Propagation (ACP) Guide

ACP is a separate utility that enables the administrator to propagate changed Administrator settings from a Test system to the Production system. The ACP utility has its own manual, *Agile Configuration Propagation Guide*, which covers installation and use by the Agile administrator.

## Getting Started with Agile PLM

*Getting Started with Agile PLM* is more than a "primer" for the PLM end-user, as it contains in-depth information about features and functions that are common to all PLM solutions. The Agile administrator will find the *Getting Started* guide to be an indispensable reference of many features in PLM, including:

- Terminology and concepts used in the Agile PLM solutions;
- General features such as:
  - Searches,
  - Workflows,

- Reports,
- Attachments and File Management;

- The user interfaces and operation of Java Client and Web Client:
  - PLM administrators generally use Java Client to set up and manage the Agile system; therefore, before starting on the present manual ('*Admin Guide*'), please examine Chapter 3, “Navigating in Agile Java Client” in *Getting Started*;
  - Most PLM users work in Web Client; some administration tasks can be done in Web Client, for example, creating users; Chapter 2, “Navigating in Agile Web Client” in *Getting Started* is recommended.

# PLM Solution-Specific User Guides

Most user guides for the PLM solutions contain a chapter (or appendix) for the administrator. These chapters deal with specific settings and configurations for each solution, and are mandatory reading for each solution purchased by your company.

For instance, every installation of PLM includes the solution called Product Collaboration. In the *Agile PLM Product Collaboration User Guide*, there is a chapter dedicated to the Agile administrator, not the end-user. The main solutions in PLM are listed in the next section.

The Agile administrator should also become familiar with the end-user documentation of PLM solutions. The manuals for end-users contain valuable information that assist the Agile administrator in grasping what the solution is all about and what the user experience incorporates.

## Agile Product Lifecycle Management Solutions

The Agile Product Lifecycle Management (PLM) suite provides the following product solutions that are accessed through Java Client and/or Web Client:

- **Agile Product Collaboration** (PC) — Management and collaboration of product record information throughout the product lifecycle, across internal organizations and the extended supply chain.
- **Agile Product Portfolio Management** (PPM) — Integration of project and product information in the context of overall product development in order to streamline business processes across the product portfolio and lifecycle.
- **Agile Product Quality Management** (PQM) — Management of customer, supplier and product quality issues, tied to the product design via a closed loop corrective action process.
- **Agile Product Cost Management** (PCM) — Management of product costs across the product lifecycle and synchronization of product cost and cost processes with both internal and external participants.
- **Agile Product Governance & Compliance** (PG&C) — Management and tracking of all substances and materials contained by any item or manufacturer part, allowing companies to meet substance restrictions and reporting requirements, design recyclable products, minimize compliance costs, and eliminate noncompliance on future products.
- **Agile Engineering Collaboration** (EC) — Management of CAD design data from the engineer's desktop directly in a central PLM product record, automating both the design and product BOM change processes and supporting IP reuse.
- **Agile Enterprise Visualization** — Visualization, markup, and analysis of native documents (office,

image, drawing, and CAD) through the Web across the extended enterprise, without the need for the original authoring tool or conversion to a neutral format.

Again, there are separate user guides for each product solution listed above, and all (except *PQM User Guide*) contain a chapter for the Agile administrator.

# What's New in Administrator

## Features and Enhancements added in Rel. 9.3

- The major feature enhancement in administrator is the Events framework for automated process extensions, script process extensions, and flexible notifications. The **Event Management** node folder has been added to **System Settings**, containing six new nodes. For more information, see Event Management on page 301 and Notifications on page 275.
- Web Client Administrator has many new features and has undergone a significant User Interface redesign. For more information, see Administrator Functions in Web Client on page 35.
- Please see Viewers & Files on page 265 for information regarding **Auto Generating Thumbnails** and thumbnail display options in Agile PLM.
- Several new roles have been added. For a list of available roles, see Roles on page 187.
- There are eight new preferences available including **Agile Drive** and **Thumbnail Support**. For more information, see Preferences on page 371.
- The Programs base class has been changed to Projects to be more consistent with industry standards.
- **API Name**: see API Name on page 19 in the Introduction chapter.
- Locations now has an **Online Help Manuals URL** field to access Online Help Manuals and User Productivity Kit (UPK) content. For more information, see Locations on page 347.

| | |
|---|---|
| **Important** | Portals is no longer supported in Agile PLM 9.3 and the Portals tab has been removed from Locations. |

## Installation Test Notice

Install and test this release on a designated development server before installing it on your production environment. Your development environment should mirror your production environment as closely as possible to provide accurate testing results. It is important to validate the installation of this release, and confirm your integrations are working correctly as part of your minimum due diligence. Any problems or questions noted during your development system testing should be resolved before installing this release on your production environment.

# Chapter 2

# Getting Started in Administrator

**This chapter includes the following:**


- Overview ..... 5
- Logging in to Agile Clients as an Administrator ..... 5
- Creating New Administrator Users ..... 6
- Dates and Times in Agile PLM ..... 8
- Administrator User Interfaces ..... 8
- Agile PLM Variables ..... 14


## Overview

This chapter collects information for the Agile administrator that is either basic or pertains to capabilities that are not found or specified by the node on the Administrator user interface.

The primary client for administering Agile PLM is Agile's Java Client.

To use Java Client, you must install it on your computer first. Java Client uses Java WebStart technology to download the software and keep it updated. For information about installing Java Client, see the *Agile PLM Installation Guide* for your application server.

Web Client is a Web-based application. You can use your browsers, as documented in the *Agile PLM Capacity Planning* guide, to run Web Client.

To see administrative features in Agile PLM clients, users must be assigned the Administrator privilege mask. The "admin" user account is pre-configured to see Administrator nodes and settings. You can use the admin account to create new users and assign the Administrator role to them. For more information, see Creating New Administrator Users on page 6.

**Important** Please examine the information in Administrator Privilege and the AppliedTo Capability on page 229. It provides information about the AppliedTo property that is a crucial part of all Administrator-related privilege masks. You do not need to grasp everything described on that page at this point, but it is crucial to know about the AppliedTo property.

## Logging in to Agile Clients as an Administrator

For security reasons, you can change the password for the admin account to something only you would know. For more information, see Changing a User's Password on page 169.

**To start Agile Java Client and log in as an administrator:**

1. Double-click the Agile icon on your desktop, or choose **Start > Agile 9.3**, or launch Agile from the URL provided by the system administrator. The Login dialog box appears.

2. Enter the username *admin* and the password *agile*. The username is not case-sensitive, but the password is case-sensitive.
3. Click **OK**. The login process is complete. The Java Client window opens, as shown in the following figure, usually with the **Search** tab displayed in the navigation pane.
4. Click the **Admin** tab in the navigation pane (on the left) to display the administrative nodes.

   A user who has not been assigned the Administrator privilege mask (or role) will not see the **Admin** tab, only the **Search** and **Analytics and Reports** tabs will be visible.

   **Note** The **Search** and **Analytics and Reports** tabs in Java Client are documented in the chapter "Navigating in Java Client" *in Getting Started with Agile PLM*.

**To start Agile Web Client and log in as an administrator:**

1. Start your browser.
2. Click the Agile Web Client bookmark, if one exists, or type the URL, for example,

   http://webserver.domainname.com/Agile/PLMServlet

   **Note** Make sure you enter the full domain name for your web server. The Web Client URL is case-sensitive.

   The Login page appears.
3. Enter the username *admin* and the password *agile*. The username is not case-sensitive, but the password is.
4. Click the **Log In** button. If you make a mistake, click **Clear** and retype your username and password.

   The login process is complete.
5. The first time you log in, the User Profile page appears. Enter the requested information. When you have finished, click **Save**.
6. After you complete this information, the Agile Web Client opens.
7. To open the administrative functions of Web Client, choose **Tools and Settings** in the menu bar and select **Administration** and then select the appropriate administrative actions from the supplied menu options.

# Creating New Administrator Users

Agile PLM uses Roles and Privileges to assign access to PLM solutions and other capabilities. The Agile administrator generally has access to the entire PLM system.

In addition, users can be given tailored privileges that enable them to administer a subset or portions of the Agile PLM system. These lesser PLM administrators are often referred to categorically as user administrators, but other names are perfectly useful, such as assistant administrator, roles administrator, or workflows administrator.

**Note** In this section, as well as in most of this manual, we are addressing users who have the Administrator or User Administrator Privilege Mask. These are powerful, system-impacting roles. They should not be confused with some other solution-specific roles in PLM that contain the word Administrator, for example, Discussion Administrator or Program Administrator. See the organization of roles in Default Agile PLM Roles on page 190.

Whether or not your company requires more than one Agile PLM administrator, it is recommended that you assign at least one user as backup PLM administrator in case the primary administrator is not available.

This section describes how to assign Administrator privileges to a new user using Java Client or Web Client. For more detailed information about how to create Agile PLM users and set user properties, see Users on page 165.

**Important** The AppliedTo property – found in the Administrator-related privilege masks, among others – is important to understand, as it controls what nodes an administrator user sees in the **Java Client > Admin** tree. The following Administrator-specific privilege masks make significant use of the AppliedTo property: Administrator (privilege mask); Admin Access for User Admin (privilege mask); and User Administrator (privilege mask). See Administrator Privilege and the AppliedTo Capability on page 229 and Options for Building Administrator Assistants on page 231.

**To create a new Administrator user in Java Client:**

1. Start Java Client and log in as “admin”.
2. Under **User Settings,** double-click **Users**. The Users window appears.
3. Click the **New** button. The Create User dialog box opens.
4. Type the **UserID, First Name**, **Last Name**, **Email,** and **Password**. Repeat the password in the **Retype Login Password** field.
5. Click **OK**. The setup window for the new user appears.
6. Click the **Role(s)** list, choose the Administrator role, and Click **OK**.
7. Fill out other user properties as needed, and then click **Save**.

**To create a new Administrator user in Web Client:**

1. In a browser, go to the Web Client URL and log in as `admin`.
2. Click **Tools and Settings > Address Book > Users**. The Users page appears.
3. Click **Add**.
4. Type the **Username** and **Login Password**. Repeat the password in the **Retype Login Password** field.
5. Then enter values for **First Name**, **Last Name**, and **Email.**
6. The General Information page appears.
7. Under the Profile heading open the **Role(s)** list, click the **Launch the Palette** button to the right of the field.
8. Choose the Administrator role and click **OK**.

9. Fill out other user properties as needed, and then click **Save**.

# Dates and Times in Agile PLM

## ...at the Server Level

The format of dates and times for object-related events (recorded by the Agile Application Server, or AAS) conforms to international standards.

The standard date format is Year [4 digits] / Month / Day, or **yyyy/MM/dd**.

The standard time format is Hour [converted to the 24-hour clock in GMT] : Minute : Second, or **hh:mm:ss**.

## ...at the Client Level

However, the format of dates and times in Agile PLM clients depends primarily on the user profile settings for **Preferred Time Format**, **Preferred Date Format**, and **Time Zone**. See User Properties Defined on page 170 for details. The settings that the user specifies do not affect how the dates are stored.

## Example

Agile PLM records object-related events based on when they occur at the AAS. However, in **History** tabs of objects, you see the local client time that an event occurred—that is, the time matches that recorded by Agile PLM clients.

For example, Jerry creates an Agile object at 5:00 PM in San Jose, California. The server that the object is created on is in San Jose, so the server records the object's creation time as 5:00 PM. Jerry's Time Zone user preference is set to "New York (EST)". This means that on his machine, the **History** tab of this object will indicate that it was created at 8:00 PM, the time it was in New York.

| | |
|---|---|
| **Note** | The administrator sets Time Zone for all users, but users can change this setting. Users should set their time zone correctly, otherwise **History** in business objects will report misleading times. |

# Administrator User Interfaces

This manual documents most of the Administrator nodes in Java Client. However, as stated in Chapter 1, documentation of the solution-specific nodes – PC, PCM, PPM, ACS, and PG&C – has been moved to the respective user guides.

# Filtering Data in Java Client

You can use a filter bar (sometimes called “table search”) to search for objects in the following nodes: **Lists**, **Criteria**, **Users**, **Roles**, **Privileges**, **Notifications**, **Event Management** and **Commodities**, as well as the **Admin History** window (available from the **Admin** tab toolbar).

**To use the filter bar in Java Client:**

1. In the filter bar’s **Filter By** dropdown list, select a value.

   The Filter By list depends on the type of object being filtered. For example, if you are searching for users, you can select First Name, Last Name, User ID, Business Phone, or Email.

2. Select an operator from the **Match If** dropdown list, such as Contains, Starts With, or Show All.
3. Type the value you want to search for in the text box.
4. Click **Apply**.

# Administrator Nodes in Java Client

The following table lists Agile PLM Administrator nodes found in Java Client.

Administration functions that are available only in Web Client are detailed in Administrator Functions in Web Client on page 35 .

Your company may not have purchased all the business solutions contained in Agile PLM; in that case, some nodes are not visible.

| Node | Description |
|---|---|
| ***Data Settings*** | |
| **Classes** | Define and manage business objects and their attributes, flex fields, and other properties |
| **Character Sets** | Configure lists of valid characters that are used to manage the text attributes |
| **Lists** | Manage lists that are used across multiple classes; also contains dynamic lists of objects such as users and items |
| **Process Extensions** | Extends the functionality of the Agile PLM system by defining custom actions to create user-driven and workflow-triggered custom actions, custom tools accessible through Agile PLM clients, custom autonumbering, and custom reports. |
| **Autonumbers** | Configure the number sources used in the automatic numbering of objects in Agile PLM |
| **Criteria** | Define criteria objects that are used in search conditions, privilege masks, and workflow settings |

| Node | Description |
|---|---|
| ***Workflow Settings*** | |
| **Workflows** | Configure, modify, and create workflows to track the progress of changes in your change control process |
| ***User Settings*** | |
| **Account Policy** | Set up systemwide rules for password age, length, and uniqueness |
| **Users** | Add new Agile PLM users to the system and modify properties for existing users, such as passwords, usernames, role and user group assignments, and more |
| **User Groups** | Define user groups and assign users to them |
| **Supplier Groups** | Select suppliers during the RFQ process; the administrator sets up global supplier groups that can be leveraged by users with appropriate roles |
| **Roles** | Define default and custom roles that control which actions groups and individual users can perform in Agile PLM |
| **Privileges** | Define privileges and privilege masks, which are assembled into user roles |
| **User Monitor** | View which users are presently logged in and terminate user sessions |
| **Deleted Users** | If needed, undelete a deleted user; the list of deleted users is maintained to provide a complete audit trail |
| **Deleted User Groups** | If needed, undelete a deleted user group; the list of deleted user groups is maintained to provide a complete audit trail |
| ***System Settings*** | |
| **SmartRules** | Manage the rules that make your change control process consistent |
| **Viewers and Files** | Define preferences for files attached to Agile PLM objects, and define viewer settings and banners and watermarks for redlines and attachments |
| **Notifications** | Customize templates for notification addressing and content |
| **Full Text Search** | Manage the indexing of attachment files |
| **My Assignments** | Configures what your users see when they click the My Assignments tab in Web Client's home page |
| **UOM** | Manage the units of measure |
| **Company Profile** | Stores useful information about the company |
| **Currency Exchange Rates** | Manage the Currency Exchange Rates table for in-system currency conversion |
| **Dashboard Management** | Enable or disable and rename the tabs of the Dashboard |
| **Product Cost Management —** PCM nodes are documented in *Product Cost Management User Guide* | |

| Node | Description |
|---|---|
| **Ship-to Locations** | Data that is factored in defining prices in RFQs |
| **RFQ Terms and Conditions** | Limit suppliers' access to an RFQ until they electronically agree to RFQ terms and conditions |
| **Product Portfolio Management —** PPM nodes are documented in *Product Portfolio Management User Guide* | |
| **Status Nodes: Cost, Quality, Resource, Schedule** | Reports whether a project's targets in the areas of cost, quality, resources, and schedule are currently being met, currently not being met, or are seriously off target |
| **Default Role** | Specifies which roles are assigned automatically to users when a task is delegated to them or when a Microsoft Project, along with its users and roles, is published to Agile PLM. |
| **UI Configuration Data** | Configures the pop-up windows on the Project Summary page and Content tab. |
| **Agile Content Service —** ACS nodes are documented in *Agile Content Service User Guide* | |
| **Subscribers** | Create profiles specifying all details of transfer orders |
| **Destinations** | Define where the output files of transfer orders will be sent |
| **Events** | Define when transfer orders are to be triggered |
| **Filters** | Define what content is included in a transfer order |
| **Package Services** | Configure the target Agile PLM system for Web services to facilitate Agile-to-Agile communication; these filters are also used for the Agile PLM Export process. |
| **Response Services** | An acknowledgment from the remote recipient of data |
| **Product Governance and Compliance —** PG&C nodes are documented in *Product Governance & Compliance User Guide* | |
| **Signoff Message** | Define message that the information supplier must agree to and sign off when submitting the declaration back to the buyer/customer |
| **Compliance Rollup Scheduling** | Define how often the system runs a scheduled rollup, and start time |
| **Compliance Rollup Rule Setting** | Control elements of suppliers' responses on declarations as they potentially impact the results of compliance rollups |
| **Supplier Declaration Process Extensions** | Set which process extensions are visible in each type of declaration request that your information suppliers receive.<br>**Note** The process extensions themselves are not defined in this node folder. |
| **Specification Mapping** | Set up a mapping between specifications and PageTwo fields that display Result Compliance for the part-spec combination. |
| **Event Management —** Event Management nodes are documented in Event Management on page 301 | |
| **Events** | Create and manage Event masks. |

| Node | Description |
|---|---|
| **Event Handlers** | Create and manage Handler masks based on "Java PX" and "Script PX" handler types.<br>Notifications, the third handler type, are created and managed in **Notifications** node. |
| **Event Subscribers** | Create and manage Subscriber masks. |
| **Event Types** | Library of preconfigured event types, upon which event masks are based. |
| **Event Handler Types** | Library of event handler types: Java PX, Script PX, and Notification. |
| **Event Handler Monitor** | Review event handler transactions, including start and execution of event handlers. |
| ***Server Settings*** | |
| **Locations** | Define Agile Application Server, File Manager, and Portal locations |
| **Database** | Set systemwide preferences for the database |
| **LDAP** | Configure a Lightweight Directory Access Protocol system and dedicated LDAP server to create user accounts |
| **Preferences** | Set systemwide preferences for your Agile PLM system |
| **Licenses** | Server licenses pertain to products; user licenses pertain to individual users of Java and Web Clients |
| **Task Monitor** | Manage and track the progress of certain scheduled tasks |
| **Task Configuration** | Manage various scheduled activities such as reports and full text search indexing |
| ***Example Nodes*** | |
| **Example Roles** | Read-only copies of all default roles |
| **Example Privileges** | Read-only copies of all default privileges |
| **Example Criteria** | Read-only copies of all default criteria |

## Administrative Capabilities

The features in Agile Administrator help you configure and adapt Agile PLM to fit the way you already do business. Administrator nodes let you manage users and tailor aspects of Agile PLM to meet your company's requirements.

- Users, user groups, roles and privileges are all "reusable objects," enabling you to easily manage user accounts, such as in the following tasks:
  - Add and remove users, or set users as inactive
  - Assign users to user groups, which consolidates previous use of departments and groups

  - Assign users to roles, and customize roles for individual users
  - Enable users to grant temporary privilege access to specified objects for other users
- Create one or more user administrators to assist in managing user accounts
- Establish a systemwide account policy for password aging, length, rules, lockout, and uniqueness
- Configure systemwide business rules that govern how the Agile PLM system responds to certain user actions
- Configure and assign AutoNumber sources
- Create workflows to route changes, each with their own set of statuses and settings, and each status with its own set of approvers, observers, and required fields
- Create subclasses of existing Agile PLM classes to create new types of objects
- Create tabs with new fields—including numeric fields and money fields—for Agile PLM classes and subclasses
- Generate reports of Agile PLM users, objects, and administrative data.

### *Guidelines for Implementing Agile PLM*

If you are setting up a new Agile PLM system, you may configure PLM by following the sequence of Administrator nodes in the previous tables. You won't necessarily set up *everything* in each node in this order, and it is useful to become familiar with all the nodes in Administrator before beginning to configure your company's business objects, users, and systemwide settings.

The tree of node folders and nodes is only a suggested starting point, not a comprehensive or certified sequence. Should you begin to configure Agile PLM for your company and find that the process is more extensive than first understood, please contact Oracle Consulting – Agile Practice or your Oracle Agile PLM sales representative.

Thorough testing before going live with Agile PLM is strongly recommended. It can take one to several weeks to configure Agile PLM to be production-ready depending on the following:

- how many Agile solutions to configure (there are several solution-specific nodes under **System Settings**, but configuring the solutions for users is a process that goes beyond the settings),
- how many Agile Users to be created, including their Roles and Privileges, and
- how complex your company's processes are (reflected in such nodes as **Classes**, **AutoNumbers**, **Workflows**, **Locations**).

You can enlist Oracle Consulting – Agile Practice to install and set up Agile PLM for your company.

## Example Nodes in Java Client

Example nodes are read-only nodes that provide a backup of default Agile PLM roles, privilege masks, and reusable criteria. The examples cannot be enabled, but they can be used to restore default functionality to the system. For example, if you modified one of the default roles and find that it is not functioning properly, you can restore the original role by saving the example.

**To save an example node as an active node:**

1. Under **Examples,** double-click **Example Roles**, **Example Privileges**, or **Example Criteria**.
2. Use the filter bar to display the example you want to save.
3. Double-click an example to open it.
4. Click the **Save As** button. The Save As dialog box opens.
5. Type the new name, and click **OK**.

**To export an example node:**

1. Under **Examples,** double-click **Example Roles**, **Example Privileges**, or **Example Criteria**.
2. Use the filter bar to display the example you want to export.
3. Select an example and click the **Export** button.
4. Type the name of the file, and click **Save**.

**Note** You can import the exported file into another Agile PLM system. When you import the example, it becomes an active role, privilege mask, or reusable criteria. For more information about importing administrative data, see Administrator Import on page 29.

## Administration in Web Client

If a user has been assigned the Administrator or User Administrator privileges, they can perform certain administrator actions in Web Client.

Click **Tools and Settings > Address Book** for user administration tasks under the **Users**, **User Groups**, and **Supplier Groups** tabs. The capabilities are documented as follows:

- **Users** – see Users on page 165
- **User Groups** – see User Groups on page 177
- **Supplier Groups** – see Supplier Groups on page 185

Click **Tools and Settings > Administration** and choose from the flyout list. The capabilities are documented as follows:

- **Dashboard Configuration** – see Dashboard Management on page 36
- **Visual Themes** – see Visual Themes on page 36
- **Report Templates** – see Report Template on page 40
- **Logging Configuration** – see Logging Configuration on page 42
- **Cache Health Monitor** – see Cache Health Monitor on page 43

# Agile PLM Variables

This section lists Agile PLM variables you can use when you configure the system.

# Default Value Variables

The following variables are used for setting default values for attribute properties.

| Variable | Description |
|---|---|
| $TODAY (must be uppercase) | The current date<br>$TODAY is not supported in Text type attributes. If you set $TODAY as the default value for a Text type attribute, the time is displayed in GMT. |
| $NOW (must be uppercase) | The current date and time<br>$NOW is not supported in Text type attributes. If you set $NOW as the default value for a Text type attribute, the time is displayed in GMT. |
| $LATESTREV (uppercase) | Defines whether or not the latest revision of an item has been released or not.<br>The only values possible are<br>$LATESTREV = Is Introductory<br>$LATESTREV = Is Released.<br>For more information, see Using $CURRENTREV and $LATESTREV on page 123. |
| $CURRENTREV (uppercase) | The current revision of the object.<br>For more information about $CURRENTREV, see next topic and Using $CURRENTREV and $LATESTREV on page 123, and Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV on page 243. |

## $CURRENTREV Criteria Logic

Using $CURRENTREV in an item criteria allows the administrator to build in privilege control based on the displayed revision. Using combinations of multiple $CURRENTREV criteria conditions allows the administrator to narrowly define which items a user is allowed to modify.

A $CURRENTREV criteria condition can be set to be **Equal To** or **Not Equal To** the following values:

- $LATEST (the latest released revision of the item)
- Status type variables, for example, $STATUSTYPE.RELEASED.
- Workflow statuses pertaining to specific workflows, for example, Default Change Orders.Pending.
- Special values for evaluating Introductory revisions:
  - $INTRODUCTORY_NOCHANGE
  - $INTRODUCTORY_PENDINGCHANGE

- $INTRODUCTORY_RELEASEDCHANGE

For detailed information about $CURRENTREV criteria values, see Using $CURRENTREV and $LATESTREV on page 123.

# Privilege Mask Criteria Variables

You may select the following variables when establishing privilege mask criteria.

| Variable | Description |
|---|---|
| $AFFECTEDITEMTYPE | Inserts the type (subclass) of object that is listed in the **Affected Items** tab or field. |
| $ATTRIBUTENAME | Inserts the name of the attribute that is selected in the **Attribute Watermark** field (see Banner and Watermark Properties on page 273). |
| $APPROVER | Inserts the name(s) of the approver(s) of the workflow |
| $CHECKOUTUSER | The user who checked out the object. |
| $CREATEUSER | The user who created the object. |
| $CURRENTREV | The current revision of an object you have selected from the rev list, or the revision of an object you are currently viewing. (See previous topic.) |
| $DEFAULTSITE | Specifies if an item's manufacturing site contains the user's default site. |
| $LATEST | The latest revision of an object; more specifically, the latest released revision for a released item, or the latest pending revision for an unreleased item (with pending changes). $LATEST is meaningful only in the Read and Modify privileges.<br>$LATEST goes on the right side of the equation. Valid use is:<br>$CURRENTREV (equal, not equal) $LATEST.<br>More information about $LATEST is in Item Relationship Creation Using $CURRENTREV and $LATEST Criteria on page 126. |
| $LATESTFILEVERSION | The latest file folder version. |
| $MAINORG | Supplier users can read and discover all users in the buyer organization as well as in their own organization (which, alone, is handled by $USERORG variable). Supplier users are prevented from reading and discovering users from other supplier organizations. |
| $NOTIFY | A default list of users selected to be receive a notification(s) |
| $OBSERVER | Inserts the name(s) of the observer(s) of the workflow |
| $USER | Inserts the name(s) of PLM users. A criteria such as "$USER Equal to $APPROVER" can be used with certain privileges, for example, Modify. |
| $USERGROUP | Inserts the name(s) of PLM user groups, based on a given user's User Group property being populated with a value, that is, a Group name. So, If a field is tied to the Admin list for User Groups, criteria can match a user to a user group; this criteria can be used with Read and Discover privileges to grant |

| Variable | Description |
|---|---|
| | object access and control to a group or team. |
| $USERORG | Used to create criteria based on the relationship between an attribute (organization or supplier) that specifies an organization, and the organization of the logged-in user. |
| $USERSITE | Used to create criteria based on the relationship between an attribute (site) and sites assigned to the logged-in user. |

## Workflow Assignment and Status Type Variables

You may select the following variables when establishing workflow assignment:

| Variable | Description |
|---|---|
| $APPROVER | The approver(s) of the workflow. |
| $OBSERVER | The observer(s) of the workflow. |
| $ORIGINATOR | The originator of the routable object. |

You may select the following “status type” variables under the **Criteria** column when setting up workflows:

| Variable | Description |
|---|---|
| $UNASSIGNED | Unassigned status type. |
| $CURRENTSTATUS | The current status type. |
| $STATUSTYPE.PENDING | Pending status type. |
| $STATUSTYPE.SUBMIT | Submitted status type. |
| $STATUSTYPE.REVIEW | Review status type. |
| $STATUSTYPE.RELEASED | Released status type |
| $STATUSTYPE.COMPLETE | Completed status type. |
| $STATUSTYPE.CANCEL | Canceled status type. |
| $STATUSTYPE.HOLD | Hold status type. |

## Routing Manager Variables

Routing manager variables facilitate management of routable objects. These variables represent the user selected from the Change Analyst, Compliance Manager, Component Engineer, Program Manager, Quality Administrator, Quality Analyst, and Price Administrator lists.

| Variable | Description | Related routable objects |
|---|---|---|
| $CHANGEANALYST | User selected from the Change Analyst list | Engineering change orders (ECOs), engineering change requests (ECRs), manufacturer change orders (MCOs), deviations, stop ships |
| $COMPLIANCEMANAGER | User selected from the Compliance Manager list | All declarations |
| $COMPONENTENGINEER | User selected from the Component Engineer list | Manufacturer change orders (MCOs) |
| $PROGRAMMANAGER | User selected from the Program Manager list | Packages |
| $QUALITYADMINISTRATOR | User selected from the Quality Administrator list | Product service requests: Problem Reports & NCRs |
| $QUALITYANALYST | User selected from the Quality Analyst list | Quality change requests: CAPAs & Audits |
| $PRICEADMINISTRATOR | User selected from the Price Administrator list | Price change orders (PCOs) |

Chapter 3

# Administrator Functions in Java Client

**This chapter includes the following:**


- API Name ..... 19
- Administrator Reports ..... 20
- Utilities on Admin Toolbar ..... 21
- History Tracking ..... 24
- Administrator Import and Export ..... 27
- Process Extensions Library ..... 29
- Multiple Language Support ..... 30


# API Name

The **API Name** field is present on many Administrator objects and PLM business objects. When it is present, API Name is a required field. It is a unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. The API Name works as a system-identifying key because it tends not to change, while the Name attribute is much more likely to change.

See the *Accessing PLM Metadata with API Name Field* chapter in *SDK Developer Guide* for more information.

## API Name field in Java Client

The following is a list of where you see the **API Name** field for objects in Agile PLM:

- **Classes >** Base Classes > Classes, Subclasses, Tabs, Tables, Attributes
- **Character Sets**
- **Lists >** List Values
- **Criteria**
- **Autonumbers**
- **Roles**
- **Privileges**
- **Workflows >** Workflow Statuses
- **Notifications**
- **UOM**
- **Dashboard Management**
- **Agile Content Service >** Subscribers, Destinations, Events, Filters, Package Services, Response

Services

- **Event Management >** Events, Event Handlers, Event Subscribers, Event Types, Event Handler Types
- **Examples >** All nodes

# Administrator Reports

You can generate Administrator reports to show the current settings for the Administrator nodes. These reports are available in the navigation pane's **Analytics and Reports** tab **> Standard Reports** folder **> Administrator Reports** subfolder. These reports are especially useful when you are planning to reconfigure an Agile PLM component.

The following table lists Agile PLM administrator reports.

| Report | Description |
|---|---|
| **Agile Classes report** | Lists current and default attribute settings for all Agile PLM classes; this is a valuable tool for Agile systems engineers when performing upgrades to the Agile PLM system |
| **Criteria Library Configuration report** | Lists existing reusable criteria and where they are used |
| **Events Configuration Report** | Lists existing event-specific objects as follows:<br>**Events**: list of existing Event Masks; each row in this section begins with chosen Event Type, then Name of the event mask, followed by all other "event"-object attributes.<br>**Event Handlers**: list of existing Handler Masks; each row in this section begins with chosen Handler Type, then Name of the handler mask, followed by all "handler"-object attributes.<br>**Note** Handlers for Notifications are not listed.<br>**Event Subscribers**: list of existing Subscriber Masks; each row in this section begins with Name of the subscriber mask, followed by all "subscriber"-object attributes. |
| **Login Failure report** | Captures failed attempts for system login and workflow signoff, and the reason for each failure. |
| **Preferences Configuration report** | Lists the current attribute and property settings of the **Preferences** and **Viewer and Files** nodes. |
| **Privilege Mask Detail report** | Lists each privilege mask by name and description, whether or not it is enabled, roles where it is used; provides Object Type, Applied To, and Criteria values |
| **Roles and Privilege Summary report** | Lists each role by name, description, and whether or not the role is enabled; lists users assigned to each role; lists the name and description of privilege masks in each role |
| **SmartRules Configuration report** | Lists the current setting of each SmartRule |

| Report | Description |
|---|---|
| **Subclass in Use report** | Lists all subclasses and where they are used: global searches, reusable criteria, workflows, privileges, and roles |
| **User Group Configuration report** | Lists all user groups, users assigned, and pertinent user properties for each user; also includes a second table that indicates each group's escalation person and criteria for escalation |
| **User Share Lists for Granted ACL report** | Lists the objects and associated roles to which users have been granted access |
| **User Subscriptions Lists report** | Lists the class attributes and actions about which users have subscribed to receive notifications |
| **User Usage report** | For each active user, lists usage over a period of time, from the start date you select to the present; includes number of logins and total online login time in minutes |
| **Users Configuration report** | Lists all users and their properties, except passwords |
| **Workflows Configuration report** | Lists all existing workflows and the properties (status properties, criteria-specific properties) of each workflow's status list |

You can save these reports for before-and-after comparisons when making changes to your database. For example, if you are going to be upgrading your Agile PLM system, you might want to first generate an updated Agile Classes report to keep a record of your current Classes configuration. Then compare it with one generated afterward.

**To generate Administrator reports:**

1. Select the **Analytics and Reports** tab and expand the **Administrator Reports** node folder.
2. Double-click the report you want. Agile Web Client opens to the specific report page, for instance, the Administrator Report: User Usage Report page.
3. Click **Execute**. The Specify Parameters dialog box opens. The User Usage report requires a From and To time range and the users you are requesting information about.
4. When you have populated the fields, click **Finish**. A message appears: "Server is generating the report. Please wait."
5. Select the encoding type—the default encoding type is Unicode (UTF-8). If you need to use a different encoding type, select it in the dropdown list. When you are ready, click **Continue**.
6. Follow the directions in the Save As dialog boxes to save the file to disk and choose a location to save it.
7. Modify the filename to include the report date and time. For example, **User_usage072604_3PM.csv**.
8. When the download is complete, click **Close** in the dialog box.

# Utilities on Admin Toolbar

In Java Client when you click the **Admin** tab, there is a toolbar of buttons that either perform a single

function or open to a utility or tool.

- **Disable System Access** – this button will close off user calls to the Agile Application Server (AAS). If you need to do this, use **Email To Users** button to let users know when and for how long they should not use the system. When you click **Disable System Access**, it changes appearance and becomes **Enable System Access**.
- **Synchronize Admin data across cluster** – this button performs a "synchronize" data routine across multiple AAS servers in a cluster; if your PLM setup uses one AAS server, this button is not enabled. An Oracle Consulting – Agile Solutions representative can provide you with more in-depth information about deploying and maintaining clustered servers.
- **Email To Users** – see Sending Email to Users on page 22
- **Global Replace** – see Global Replace on page 22
- **Admin History** – see Administrator History on page 25
- **Disable Triggering Events** – see Enable/Disable All Event Triggers on page 326

# Sending Email to Users

Use the **Email To Users** button to email a message to all active (enabled) users, all logged-in users, or to specific users that you select.

When you click the **Email To Users** button, the Email to Users window appears. You can send email to the following groups of recipients:

- **All active users** – Email all enabled users a message that they will receive the next time they log in to Agile PLM. For example, you may want to inform all enabled users that a new class or subclass or workflow is now available for their use.
- **All logged-in users** – Email all logged-in users a message that you want them to receive immediately. For example, you may want to inform all logged-in users that the Agile PLM system will be down in 10 minutes.
- **Other users** – Email a message to selected users. After you select the radio button, open the address book by clicking , from which you choose the users for your message.

Enter the message in the **Content** field, then click **Send**.

# Global Replace

The **Global Replace** button lets you replace one user with another in workflows, escalations, and transfer authority assignments. You can use this capability in a variety of situations, for instance, when an employee leaves the company or moves to another assignment. Global Replace can accomplish the following:

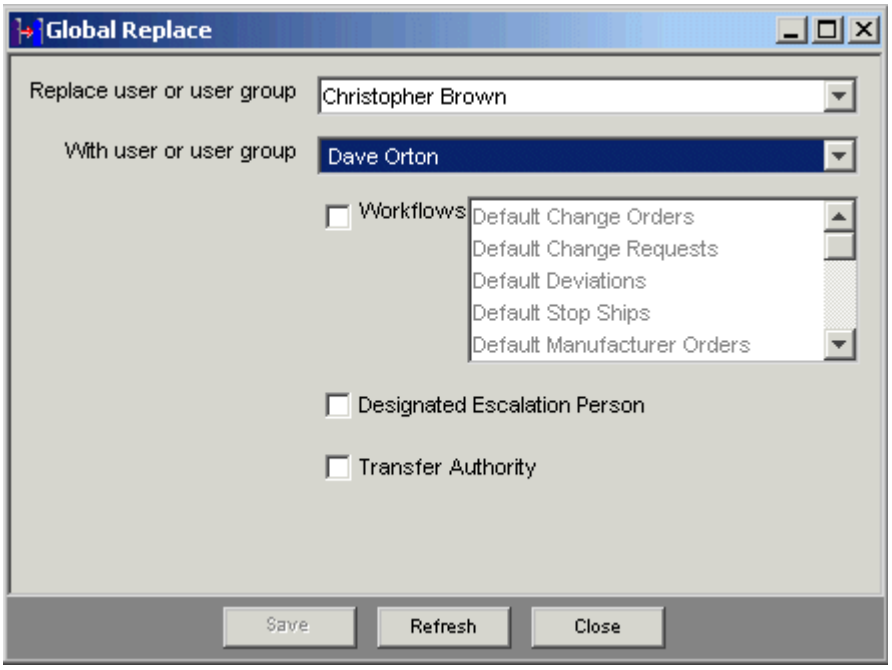
- Replace a user as an approver, observer, or routing manager in a single workflow or all existing workflows; however, this does not replace the user in the actual changes he is currently working on, which must be done manually.
- Change the designated escalation person for single or multiple users and user groups.
- Change the transfer authority person for single or multiple users and user groups. You can replace To users but not From users.

The “users” in the **From** and **Replace With** drop-down lists include all existing user groups. See Users and User Groups.

## Global Replace Properties

**To define or change the settings in the Global Replace window:**

1. Click **Global Replace** in the **Admin** tab toolbar. The Global Replace window appears.




You can specify the following values in the Global Replace window:

| Property | Description |
|---|---|
| **Replace user or user group** | The “Replace” person is the user or user group you are replacing. Select a user or user group from the dropdown list. |
| **With user or user group** | The “With” person is the user or user group who will replace the “From” person. Select a user or user group from the dropdown list. |
| **Workflows** | Select a specific workflow from the dropdown list. You can use Ctrl-shift or Shift-click to select all the workflows if you want to change every instance of the “Replace” person to the same “With” person in all workflows. With the checkbox selected, the “Replace” person is replaced with the “With” person in approver, observer, and change analyst (routing manager) capacities in the designated workflow. |
| **Designated Escalation Person** | Replaces the “Replace” person with the “With” person as the designated escalation person. (See Designated Escalation Person.) |
| **Transfer Authority** | Replaces the “Replace” person with the “With” person as the person who has the authority to approve or reject the “Replace” person’s changes. (See Transfer Authority for Self on page 220.) |

If you have selected more than one workflow in the **Workflows** field, when you click **Save** you see this confirmation message: “Are you sure you want to replace <*Replace person*> with <*With person*> for all Workflows?”

If the “Replace” person is a routing manager, and the “With” person is not, a warning appears. If you answer the prompt with **Yes** (to continue), when the “Replace” person is used in any default routing manager property (for example, Default Change Analyst), this function will be filled by a non-routing manager. If that is an unwanted result, you should instead answer the prompt with **No** and select a more appropriate “With” person.

For more information about routing managers, see Routing Managers on page 130.

# History Tracking

The Agile PLM system tracks several kinds of “history,” some available only to the administrator and some available to all users. These topics are considered in the following sections:

- **Object History and Administrative Object History** – the object history shows only the history of that object, which includes all actions performed by users and administrator users; those actions that involved an administrator also are listed in the Administrator History window.
- **Administrator History** – records administrative actions on all objects in Agile PLM. Administrator History also records history of deleted objects and settings.
- **History of Deleted Objects** – when an object is deleted, its history is preserved. This type of history is also accessed in Administrator History.
- **Previous Administrator Histories** – displays object history that took place in previous releases of Agile PLM. This kind of history is in Previous Administrator Histories. This will only have content for customers upgrading from Agile 8.5 or Agile Advantage.

## Object History and Administrative Object History

Every object in Agile PLM has a **History** tab, on which all individual actions performed are recorded, for example, Create, Modify, and Delete. Different data is recorded on the **History** tab depending on the object type. The history of actions performed by an Agile PLM administrator appears also in Administrator History (see Administrator History on page 25).

All objects have a record of the information listed in the following table on their **History** tabs. Many kinds of Agile objects have additional fields on their **History** tab.

| Property | Description |
|---|---|
| **Action** | The action that was performed, for example, create, modify, delete. |
| **User** | The user who performed the action. |
| **Local Client Time** | The date and time the action was performed. |
| **Details** | Details of the object, property, or action after the modification. |

When a modification is made to an administrative object on a node that records the assignment or reference of one object to another object, history is logged on the object that "owns" the reference or assignment. See the table below. For example, if you assign a role (object) to a user (object), the user object owns the referenced role object, and the **History** tab of the user object is updated.

| Owner (history logged on this object) | | Reference or assignment object |
|---|---|---|
| Role | owns | Privilege |
| Privilege | owns | Criteria |
| Subclass | owns | AutoNumber |
| Workflow criteria nodes (matching criteria, exit criteria) | own | Criteria |
| User | owns | Role |

# Administrator History

When a modification is made to an administrative object, the action is recorded and displayed in the Administrator History window. Administrator History also records history of deleted objects and settings.

**Note** Administrator History does not include User history or User Group history.

The following table describes the fields in the Administrator History window. These fields are similar to the history table fields for Agile PLM administrative objects.

| Field | Description |
|---|---|
| **Action** | The action that was performed, for example, create, modify, delete. |
| **Object** | The name of the object on which the action was performed, for example, the name of a workflow, privilege mask, or criteria. |
| **Username** | The user who performed the action. The username is an active link; click it to open the user's tabbed page. |
| **Local Client Time** | The date and time the action was performed. |
| **Details** | Details of the object, property, or action after the modification. |

## Purging Administrator History

The Administrator History can be cleared or purged.

| | |
|---|---|
| **Caution** | The familiar “Delete” button implies different behavior in this context. You cannot delete selected lines from the Administrator History window; this works as a true “purge” capability. Purge also removes the records in the Previous Administrator History window. |

**To purge Administrator History:**

1. On the **Admin** tab toolbar, click the **Admin History** button .
2. On the Administrator History window toolbar, click the **Purge** button .
3. You are prompted with three options to answer the question, **Would you like to export the existing history?**
   a. **If Yes, purge all history records with an export file:** If you click **Yes**, all records in Administrator History are purged, and an export file is created with all the data. This is a recommended course if your company was ever audited and these archives are required.
   b. **If No, purge all history records without an export file:** If you click **No**, all records in Administrator History are purged, and an export file is not created.
   c. **If Cancel, cancel purge action:** If you click **Cancel**, the operation is not executed.
4. If you click **Yes** or **No**, all records in Administrator History are instantly purged. They cannot be retrieved.

# History of Deleted Objects

When an object or setting is deleted, the history of the object or setting is maintained in Administrator History. However, the data is no longer maintained on the object or setting **History** tab.

For example, if you delete an object, such as a workflow, its history entries, including the delete action, are maintained in Administrator History. However, there is no longer a **History** tab for the workflow object (the object does not exist anymore), so there is no individual object history for that workflow.

If you remove one object from another—for instance, removing a privilege from a role—the removal of the privilege is recorded in **Admin History** with reference to the role.

# Deleting Objects in Agile

Java Client and Web Client support “soft” and “hard” deletes. When you soft-delete an object in the database, it is not permanently deleted. A soft-deleted object is marked “Deleted” in the database, and its object number or name is reserved. You cannot create another object with the same type and number or name as a soft-deleted object.

| | |
|---|---|
| **Note** | You *can* undelete a soft-deleted object, if you have the appropriate privileges set by the Agile administrator. |

When you run a search, soft-deleted objects do not appear in the search results table. To locate soft-deleted objects, you must run the predefined deleted object searches in the **Recycle Bin Searches** folder.

**Note** You have access to the **Recycle Bin Searches** folder if **Recycle Bin Searches** is included in the **Searches** field of your user profile. If you have the appropriate privileges, you can edit your user profile and add **Recycle Bin Searches** to the list of searches in the **Searches** field.

A hard-deleted object is permanently removed from the database.

**Caution** You *cannot* undelete a hard-deleted object.

To soft-delete, hard-delete, or undelete an Agile object:

- You must have the appropriate Delete or Undelete privileges for that object.
- The object must meet the conditions that allow it to be deleted or undeleted.

  In general, if deleting or undeleting an object would cause the compromise of data integrity, the delete or undelete action is disabled.

You cannot delete an object unless all of its relationships and subscriptions are removed.

For detailed information about deleting specific *object* types see the *Deleting Agile Objects* section of *Getting Started with Agile PLM*.

**To soft-delete an object in Java Client:**

1. Select and open the object you want to delete.
2. Click the **Delete** button, and respond **Yes** to the confirmation prompt.

The object is soft-deleted.

**To undelete an object in Java Client:**

1. Run the appropriate Deleted <*objects*> search from the **Recycle Bin Searches** folder.
2. Open the deleted object you want to restore.
3. Click the **Undelete** button.

**To hard-delete a soft-deleted object (if you have the appropriate privileges) in Java Client:**

1. Run the appropriate Deleted <*objects*> search from the **Recycle Bin Searches** folder.
2. Open the object.
3. Click the **Delete Object** button, and respond **Yes** to the confirmation prompt.

# Administrator Import and Export

In several nodes of Administrator, exporting and importing functions are available.

- **Data Settings**
  - **Criteria**
- **Workflow Settings**
  - **Workflows**

- **User Settings**
  - **Roles**
  - **Privileges**

You can export Agile PLM-object definition data to a file. You can import an Agile PLM-object definition data file to create a new object. Example objects (such as example roles) cannot be imported or exported.

**Note** On the **Tools** menu of Agile PLM clients, **Import** and **Export** utilities can be used to import and export product data, not Agile PLM system metadata. The Import and Export utilities are documented in the *Agile PLM Impor*t and *Export Guide*.

**Important** When you import Administrator data from a test system to a production system, verify the imported data as completely as is possible immediately.

# Encoding Types

When you import or export workflow data, you must select the encoding type. The following table lists available encoding types.

| Value | Note |
|---|---|
| Western European (ISO ) | ISO-8859-1 |
| Japanese ( Shift JIS ) | SJIS |
| Traditional Chinese ( Big 5 ) | BIG5 |
| Japanese ( EUC ) | EUC-JP |
| Simplified Chinese ( GB2312 ) | GB2312 |
| Unicode (UTF-8 ) | UTF8 |

# Administrator Export

The Administrator Export function exports the necessary data to create the object. Not all properties are exported. For example, exporting a privilege mask does not export the Where Used information. You can examine the exported text file to determine which properties are exported for each type of object.

**To export Administrator data:**

1. Open the node you want to work with: **Workflows**, **Criteria**, **Roles**, or **Privileges**.
2. Select the objects you want to export. You can export one object, or all objects in the table; the Workflows, Criteria, and Privileges windows have an **Export All** button.

   **Note** In windows with a filter bar (see Filtering Data on page 9), to display all the objects in the node, Select Show All in the **Match If** dropdown list, and click **Apply**.

3. Click the **Export** button in the window toolbar.

   For *workflow exports only*, the Encoding Type dialog box opens. See the table above. The encoding type generally used is Western European (ISO). If you need to use a different encoding type, select it in the dropdown list, and click **OK**.

   **Note** You must specify the same encoding type when you import the file later.

4. Follow normal Save As procedures to give the file a name and select a location to save it to disk. Modify the filename to make it easy to identify. You might include the object exported, the date, and the encoding type. For example, **ChangeOrderWorkflow072602_ISO.txt**.
5. When you are ready, click **Save**. Verify the data in the file you exported.

## Administrator Import

You can import an exported text file to create an object in the Agile PLM system.

You cannot create two objects with the same name. For example, you cannot import an object data file for a workflow named ABC if there is already a workflow named ABC in your Agile PLM system. You have the option of overriding the existing named object; in that case, the imported object replaces the existing object in the Agile PLM system.

Note that some properties—for example, user assignments—are not exported. When you import the object data text file, you will create a new object without those properties (for example, with no user assignments).

**Note** Before you import it, open the text file you want to import to verify which object properties were exported.

**To import Administrator data:**

1. Open the node you want to work with: **Workflows**, **Criteria**, **Roles**, or **Privileges**.
2. Click the **Import** button in the window toolbar. The Import File dialog opens.

   For workflow imports only, the Import File dialog includes an Encoding Type field. Make sure the encoding type matches the exported file of workflow(s).

**Caution** You must select the same encoding type you used when you exported the object. Import results are not reliable if you do not use the same encoding type.

3. Next to the Files to be Imported field, click the **Browse** button to locate and select the file you want to import. Click **Open**.
4. Click **OK**. The import operation takes place. Verify the data in the imported file.

# Process Extensions Library

In Java Client, use **Tools > Process Extensions** or the **Process Extensions** button or **Admin** tab > **Data Settings** > **Process Extensions** to access the library of Custom Process Extensions (that is, those

PXs that preceded Events-based PXs.

If there are no Custom PXs deployed on your Agile PLM system, that item is unavailable on the **Tools** menu, and "Empty" appears below the button when you click it.

For details about configuring custom process extensions, see Process Extensions on page 103. For technical information on how to develop custom process extensions for your Agile PLM system, see the *Agile PLM SDK Developer Guide*.

**Note** With the introduction of Event Management, it is possible to have more flexibility with PXs, through Java PX handlers and Script PX handlers. The previously existing Process Extensions capabilities remain intact, but Events Management allows PXs to be triggered through normal user and system actions. See Event Management on page 301. For technical information on developing the new PX handlers, see the *Agile PLM SDK Developer Guide*.

# Multiple Language Support

Agile PLM is an internationalized application that permits companies to enter non-English label data, and for users to see their client user interface in their preferred language. English is included with all licenses. Agile PLM is certified for French, German, Japanese, Korean, Russian, Simplified Chinese, and Traditional Chinese.

**Important** This list of supported languages is current as of May, 2010 and is subject to change. Please see the **Languages** tab found in the **Preferences** node for your current installation supported languages.

**Note** For the Product Governance & Compliance solution, there is a fully supported Japanese version of the Rollup in Excel template. Localized versions of the Rollup in Excel template are supported in Chinese, French, and German. Additionally, there is a fully supported Japanese version of the JGPSSI template for creating JGPSSI Declarations.

Every user has a language preference (**<user object> > Preferences > Format Preferences** (heading only) **> Language**) that you will likely set during configuration of your company's business objects (which includes, of course, Agile users).

If the company has purchased a localized language as part of PLM, that language will appear in the dropdown list for the Language property.

**To set the Language preference for a user:**

1. Open the user profile. (This will likely be set when you create your company's Agile users. This task instructs for an already-created user object.)
2. Click the **Preferences** tab.
3. Under **Format Preferences**, click for the **Language** field.

   The dropdown list shows English and any other languages your company has licensed.
4. Select a language, and click **Save**. That user will see supported elements of Agile in that language.

If you change any user's Language preference, that user (if logged in) must log out and log back in to see changes in their user interface.

Users can switch their own Language preference (again, non-English languages that the company has a license for). In Java Client, the user clicks **Current User...** from the **Settings** dropdown menu of the main toolbar and follows the previous instructions.

In Web Client, the user's path to the Language setting is **My Settings > User Profile > Preferences > Edit > Language** (under **Format Preferences**).

For more information about Agile PLM Language settings, see Systemwide Preferences on page 379.

# Administrator Data that is Supported for Multi-Language Capability

We use the term "multi-language capability" to refer both to those "Administrator-editable" objects that can hold different licensed languages inputs separately, and those non-editable objects that can be displayed in different languages according to any end-user's Language preference setting.

The following list includes those Administrator nodes whose objects are supported by the multi-language capability. If the node is listed as supported, it does not matter whether the object is created by the administrator or by a user. (As always, the ability for users to create an object in Agile is defined by their roles and privileges.) Names and Descriptions of these objects are supported for multi-languages unless otherwise noted.

## Supported Administrator Nodes

Administrator-editable items can hold names in any enabled language. The administrator can configure these items into a different language for users by setting the Language user preference. For instance, Company A has a localized version of Agile PLM with Japanese enabled (and, by default, English). The administrator creates a list; he can name the list differently for English users and Japanese users. Users with language preference set to Japanese will see the list named in Japanese as the administrator configured.

Multi-language support is provided for the following kinds of objects:

### *Data Settings*

- Classes
- Character Sets
- Lists
- Process Extensions
- Autonumbers
- Criteria

### *Workflow Settings*

- Workflows

### *User Settings*

- Roles
- Privileges

### *System Settings*

- Notifications (Name, Description, Subject, Body)
- Full Text Search is supported for attachment files in Japanese, Traditional Chinese, Simplified Chinese, German, and French localized versions
- Dashboard Management
- Product Portfolio Management:
  - Schedule Status
  - Cost Status
  - Quality Status
  - Resource Status
  - Default Role
- Agile Content Service:
  - Subscribers
  - Destinations
  - Events
  - Filters
  - Package services
  - Response services
- Product Governance & Compliance:
  - Signoff Message
  - Compliance Rollup Scheduling
  - Compliance Rollup Rule Setting
  - Supplier Declaration Process Extension
  - Specification Mapping
- Event Management

## Supported Predefined System Objects

Objects that are predefined in Agile—that is, are not editable in Administrator, for example, an Example Role—are fully supported by multi-language capability. For instance, a user whose Language preference is English who logs in will see the “Example–Administrator” role in English; when the same user logs in with the Language preference set to Japanese, he will see a translation of “Example–Administrator” as configured by the Japanese user administrator.

# Administrator Data that is Not Supported for Multi-Language

# Capability

While most types of Administrator data are supported for multi-language capability, the following types of Admin data are not supported for multi-languages, but only in localized user interfaces.

## User Settings

- Users
- User Groups
- Supplier Groups

## System Settings

- Viewers & Files
- Company Profiles
- PCM Ship-To Location

## Any values/strings input by user

## Attribute names with "01", "02", "03", etc appended

- As an example, let's say Company B has a localized version of Agile PLM with Simplified Chinese enabled (as well as English). The administrator, who has a language preference of English, creates a user group and names it UserGroup_1. Then, another Agile administrator for Company B, whose language preference is set to Simplified Chinese, changes the name of UserGroup_1 into a Chinese name with Chinese characters. Now the name of the user group appears to *everyone* as the Chinese name in Chinese characters, even those users whose language preference is English.

| | |
|---|---|
| **Note** | Full Text Search is supported for attachment files in Japanese, Traditional Chinese, Simplified Chinese, German, and French localized versions. |

# Configuring a System for Multi-Languages

When you set up fields in any language, you have to be logged in with your own Language preference set to that language. If you make changes in Japanese with your Language preference set to Japanese, and then you want to make changes in English, you must change your Language preference to English, log out, log back in, and proceed. Or you can log out and log back in as a different administrator user whose Language preference is English.

To give an example of how the Language setting for individuals is handled by the multi-language capability in Agile, let Jim and Jan be two users:

- **Case 1**: Company C has an English version of Agile only. Everyone in Company C sees Agile user interface and business objects in English; no user can change their Language user preference to another language.
- **Case 2**: Company D has a localized version of Agile with Japanese enabled (and, by default, an

English license). You have configured the company so all users' Language preference is Japanese. Jim sets his Language user preference to English. After logging back in, Jim now sees his user interface and the objects he creates in English. Jan is a typical user whose Language preference has remained Japanese: the objects that Jim created appear to Jan in Japanese.

- **Case 3**: Company E has a localized version of Agile, with Simplified Chinese enabled. The administrator has configured the users at the Hong Kong office so their Language user preference is Chinese. Other people in Company E working outside Hong Kong still keep English as their language preference. Now the administrator creates a subclass of the Parts class, and names it "Modules". Objects from this subclass appear to everyone in Company E with the name Modules, no matter which Language is set in the user preference.

  Later on, the administrator decides that the Modules subclass should be renamed in Chinese for the Hong Kong users. He sets his own Language preference to Simplified Chinese, and renames the Modules subclass as a translated name in Chinese characters. Now the Modules subclass name appears in Chinese characters to all Hong Kong users with Language user preference set to Chinese. But the English users still see the subclass name Modules in English.

- **Case 4**: Company F has a localized version of Agile, with Simplified Chinese enabled. The administrator has configured the users at the Hong Kong office so their Language user preference is Chinese. Everyone there adds information (for instance, in creating objects) in Chinese. When a user in Hong Kong creates a business object, it appears in Chinese to everyone in Company F, no matter which Language they have set in the user preference.

If you create a new subclass, its name can enter the system in localized (translated) form; however, if you rename the newly created subclass, or any subclass, it is renamed in the same language you are logged in with (Language user preference).

For Java Client to show non-English characters (in the user interface and user-entered text), it must be running on an operating system that supports a particular language. If it is not, some anomalies can appear, for example, "blank squares" instead of proper non-English characters. This is often the case for double-byte character sets such as Japanese and Chinese.

For instance, Jan is using a machine with a Japanese OS. She enters Japanese text into an ECO's **Description** field. Then Jim launches Java Client on an English OS machine; when he views this ECO, Japanese text that was entered into the **Description** field may appear corrupted (having "blank squares"): the English OS may not support Japanese characters (Windows English version requires installing the Asian Language Pack to support Japanese characters).

Chapter 4

# Administrator Functions in Web Client

**This chapter includes the following:**


- Overview to Administrator in Web Client ..... 35
- Configuring the Dashboard ..... 36
- Visual Themes ..... 36
- Report Templates ..... 40
- Logging Configuration ..... 42
- Cache Health Monitor ..... 43
- Deleting Objects in Agile ..... 47


## Overview to Administrator in Web Client

Agile Web Client contains certain Administrator capabilities, accessible only to users who have been assigned the Administrator or User Administrator role, or more specifically the Administrator or User Administrator privilege.

The "user" and "group" administrator capabilities are accessed via Web Client's **Tools and Settings > Address Book**. Other administrative capabilities are found in Web Client's **Tools and Settings > Administration** flyout menu.

This chapter documents utilities for the Agile administrator that are unique to Web Client. The table offers brief definitions of all Web Client Admin nodes, as well as cross-references to the applicable documentation.

| Node | Description |
|---|---|
| *Users and Groups*<br>– accessed in Web Client via **Tools and Settings** > **Address Book** | |
| **Users** | Add new Agile PLM users to the system and modify properties for existing users, such as passwords, usernames, role and user group assignments, and more |
| **User Groups** | Define user groups and assign users to them |
| **Supplier Groups** | Select suppliers during the RFQ process; the administrator sets up global supplier groups that can be leveraged by users with appropriate roles |
| ***Web Client Administration***<br>– accessed in Web Client via **Tools and Settings** > **Administration** | |
| **Dashboard Configuration** | Configure the PLM Dashboard – see Configuring the Dashboard on page 36. |
| **Visual Themes** | Configure the display of the Web Client user interface for all users, including the |

| Node | Description |
|---|---|
| | definition of colors, font style and size – see Visual Themes on page 36 |
| **Report Templates** | Create and manage report templates – see Report Template on page 40 |
| **Logging Configuration** | View and configure the loggers within Application – see Logging Configuration on page 42 |
| **Cache Health Monitor** | Ensure that all changes to Administrator data is properly replicated across the cluster – see Cache Health Monitor on page 43 |

# Configuring the Dashboard

Each Dashboard tab contains a set of configurable widgets that display information in tables and charts. You can configure which of these widgets should display in each tab and where they should appear.

**To configure each Dashboard tab:**

1. In Agile Web Client **Tools and Settings > Administration** menu, choose **Dashboard Configuration**.
2. In the **Configure Tab** field, select the tab you wish to configure. The widgets available for the selected tab are displayed in the **Available Content** column.
3. Move the widgets you want displayed to the **Row** columns.
4. Use the direction arrows to reorder the widgets as desired.
5. When you finish configuring all the tabs, click **Save**.
6. To exit the configuration window, click **Close**.
7. Refresh the Dashboard to see your changes.

# Visual Themes

Visual Themes lets the administrator define a visual look, or style, that the PLM users will see across the enterprise. This allows the configuration of Web Client to be consistent with other deployed applications or with corporate colors.

## Selecting Theme Options

Styles are defined and modified by choosing **Tools and Settings > Administration > Visual Themes**. The Select Theme Options page appears, with three fields.

### *Branding Logo*

There are some default Oracle branding logos in the dropdown list. To use your company's branding logo, a graphic file will have to be added to the web.xml file. For instructions on this procedure, see Replacing the Oracle Logo on page 37. Once another logo is added, it appears in

this dropdown list.

### *Icon Motif*

These are Agile's "Mechanical", "Electrical", and "High Tech" symbols that appear on this page.

### *Color Palette*

On the Select Theme Options page, the **Color Palette** list box lists all available style schemes. Those with an asterisk (*) are the out-of-box System Styles, which cannot be modified or deleted.

You can select a visual style theme from the dropdown list and begin modifying it with one of the following actions:

- New themes can be developed using **Save As**. Clicking the **Save As** button brings up the Specify Palette Colors page.

  At first, you must enter a new name in the Name field. If you alter the elements of a System Style without assigning a new name, the system does not save your new scheme; as soon as you enter a distinguishing name (regardless of whether you have altered any of the fields), the system can save it.

  This is the way you can "create" a new style scheme from the default System Styles (and the only way to create a new scheme at first, while there are no non-default color-palette Styles).

  For more details, see Modifying Styles on page 38.

- Click the **Delete** button to delete a selected style scheme.

  This button is disabled when a System Style is in the field. You cannot delete a System Style (denoted by the “*” in the name).

- Click the **Edit** button to modify a selected style scheme.

  Clicking the **Edit** button brings up the Specify Palette Colors page.

  For more details, see Modifying Styles on page 38.

  | **Note** | The button is disabled when a System Style is in the field. As noted, you cannot modify the System Styles, except by working through the SaveAs feature. |
  |---|---|

## Replacing the Oracle Logo

If you want your company’s logo (or any other co-branding image) to appear in place of Oracle’s logo, or to alter its height or width, the easiest way is to alter the file **web.xml**.

**To replace the default company logo with a custom logo:**

1. Log in to server side.
2. Find **web.xml** in the Oracle home directory.
3. The bullet-items below give some samples of changing specific parameters in the web.xml file:
   - These lines of code allow you to enter the filename of your custom logo file:

```
<context-param>
   <param-name>cobranding.image</param-name>
   <param-value>logo.gif</param-value>
</context-param>
```

Change the filename (including extension) where "`logo.gif`" appears.

- These lines of code allow you to change the height of the logo:

```
<context-param>
   <param-name>cobranding.image.height</param-name>
   <param-value>25</param-value>
</context-param>
```

Change the parameter value "`25`" to another value.

- These lines of code allow you to change the width of the logo:

```
<context-param>
   <param-name>cobranding.image.width</param-name>
   <param-value>114</param-value>
</context-param>
```

Change the parameter value "`114`" to another value.

- These lines of code allow you to change the title of the Oracle Agile window or page:

```
<context-param>
   <param-name>application.window.title</param-name>
   <param-value>Product Lifecycle Solution</param-value>
</context-param>
```

Change the parameter value "`Product Lifecycle Solution`" to another value.

- The next few lines of code allow you to update the tooltip by changing the text:

```
<context-param>
   <param-name>cobranding.image.altext</param-name>
   <param-value>Oracle|Agile</param-value>
</context-param>
```

Insert the new text where "`Oracle|Agile`" appears in the example.

4. Restart the application server.
5. Log into Web Client to verify or evaluate your change(s).

# Modifying Style Schemes

Once you have modified a style theme, you can use these buttons for further actions:

- Click **Preview** to preview the modified style scheme. This displays the Web Client interface in the selected scheme on your machine only. (All other users still see the user interface in the currently applied scheme.)

**Note** The Preview button displays only when the user creates a new theme.

- If you want to save the current scheme, click **Save**. This does nothing more than simply save the settings under the name in the Name field, as well as add that scheme to the list in the Color Palette field. (Clicking **Save** does not propagate the scheme to your Web Client users.)

  Whether or not you have saved it, if you wish to make further modifications to the scheme you are modifying, you may do so. You can save over the scheme you saved, or enter a new name and thereby save a new scheme.

- To cancel the preview of the selected scheme, click **Cancel**. You are returned to the Select Theme Options page.

- With a named color-palette scheme selected, when you click the **Apply** button, the selected scheme becomes current throughout your system. This changed scheme is seen by active users as soon as their screen is refreshed, or when a user next logs in to Web Client.
- If you do not want to keep the new style in the Style Theme list, you can simply **Delete** it.

The following table lists the fields on the Specify Palette Colors page, with descriptions of what each property modifies in the Web Client user interface (UI).

| Field | Area or Element in UI |
|---|---|
| Name | Name of specified style scheme. The name of each scheme that is saved appears in the Color Palette dropdown list. |
| Gradient | "Gradient" or "Flat" applies a filter to certain areas of the UI, so the chosen color has no shading (Flat) or appears to have a light-to-dark quality (Gradient). |
| [ **Background** column ] | "Hex" value that specifies the main color of the field |
| [ **Text Color** column ] | "Light" results in white text lettering, or "Dark" results in black text lettering. |
| [ **Sample** column ] | This color swatch displays how the color you specified in Background and the Text Color value will look together. |
| Global Toolbar | Toolbar along the top of the Web Client UI |
| Dialog Bars | Top and bottom color bars in action dialogs |
| Widget Title Bar | On the Dashboard, background color where the widget's name is displayed |
| Selected Controls | Areas like tabs (in objects) or the drawers (on the Navigation Pane) when selected |
| Non-Selected Controls | Same elements named in "Selected Controls," only when they are not selected |
| Control Highlight | Drawer (in Navigation Pane) & Dropdown list item when mouse-over; Table Row when selected |
| Message Area | Shading in the Information line (near top of an object) |
| Backgrounds | Various backgrounds not assigned by above fields |
| Slider Separator | Small band of "outline" to give the slider bars definition |

## Specifying Colors in Hexadecimal Format

The PLM application has provided a few default style schemes, but it is likely that your company will prefer that Web Client have a corporate "look and feel." You have probably noted that most of the fields in the Specify Palette Colors page are in Hexadecimal format. Web designers use a variety of tools to specify colors for elements of web documents, such as RGB triplets expressed in hexadecimal (a "hex triplet").

While the Specify Palette Colors page can accept any color in hex format, you will need to access such color tools to generate coordinated color schemes and their corresponding "Hex" format values. For example, the Adobe web site http://kuler.adobe.com/#themes/rating?time=30 can be useful for this purpose.

# Report Templates

Agile PLM integrates existing Standard Reports with Custom Reports, which are modifiable through Oracle BI Publisher.

There is not a separate installation of Oracle BI Publisher required because Agile PLM includes a BI Publisher in order to:

- Run a standard Agile report
- Modify a standard Agile report layout, including adding flex fields
- Create new reports
- Get data through Agile PLM SDK and generate reports using Agile Java PX

**Note** The version that is bundled with Oracle Agile PLM is a restricted use version only.

Further information and documentation about the BI Publisher tool is located at the following web site: http://xdo.us.oracle.com/?tab=doc (http://xdo.us.oracle.com/?tab=doc).

End-user documentation is found in the chapter on Reports in *Getting Started with Agile PLM*.

This section details how the administrator prepares the layout template for the end-user reports that are available through the BI Publisher interface.

## Manage Report Template (AppliedTo Choice)

"Manage Report Templates" is a choice in the Administrator-specific AppliedTo property (that is, found in **Administrator** and **Admin Access for User Admin** privilege masks).

When it is moved to the Selected table, Manage Report Templates enables you or an assistant administrator to upload templates for reports in the PLM Report Template utility. Choose **Web Client > Tools and Settings > Administration > Report Template** to bring up the Manage Report Template page.

Users see these templates, per assigned privileges, when they use the Reports function in Web Client.

## Dynamic and Static Templates

There are two template modes: Static Template or Dynamic Template.

#### Static Template

The only two reports that support **Static Template** and generate files in RTF are the BOM Explosion Report and the Manufacturer BOM Report. This is accessed when the user selects both “Use BI Publisher” and “Use Template For Layout” options. The Static Template is the only mode that supports XML format.

#### Dynamic Template

All custom reports and most of the standard reports use the **Dynamic Template** to define and generate

reports in multiple output formats such as PDF, HTML, Excel, and Word.

# Creating Report Templates

The administrator can define layouts by creating RTF templates in addition to the out-of-box layout already provided.

In Java Client, select **Admin > Server Settings > Preferences**. **Enable BI Publisher for Reports** has two values of Yes or No. When preference is set to No, reports are generated using the existing reporting layout. When preference is set to Yes, reports can be generated in multiple output formats using BI Publisher. For information on the system preference **Enable BI Publisher for Reports** see Preferences on page 371.

**Note** If the file server is down, templates will not generate reports as they are stored in the file server.

1. In Web Client, navigate to **Tools and Settings > Administration > Report Template**.
2. Click **Create** and the Create Report Template window appears.
3. Choose a name and enter a description for the template.
4. Browse to the location of your template RTF files.
5. Then click **OK** and your template is added to the list of options for templates. You can **Get** the template file and now modify it. You have the option to use Microsoft Office (Word) or Adobe Acrobat to design the layout of the report. There is also the option to use BI Publisher's Template Builder plug-in for Microsoft Word to employ drag-and-drop designing.

# Reports Unsupported by BI Publisher

The following reports are not supported with the BI Publisher framework template:

- Change Package Report
- Change Metrics
- Change Backlog
- Change Cycle Time
- Quality Metrics
- Quality Backlog
- Quality Cycle Time
- Failure Trend
- RFQ Export
- IP Transfer
- Project Actual vs Budgeted Cost Report
- Baseline Comparison Report
- Portfolio Cost Report

- Resource Pool Consumption Report
- All PCM Reports

# Logging Configuration

## – For use by Agile Support ONLY! –

The **Logging Configuration** monitor has been added to permit Agile Support to perform quick changes to the logging configuration without stopping and restarting your company's Agile Application Server (AAS).

| | |
|---|---|
| **Important** | Because the Logging Configuration utility is intended for use by Agile personnel in troubleshooting a customer's system, and is not intended for "self-treatment" of any problem, the various configuration scripts and Level menu choices are not documented herein. |

| | |
|---|---|
| **Caution** | You should never change the logging configuration without first receiving a recommendation from Agile Support. |

**Use of the Logging Configuration node:**

1. When you, the PLM customer, encounter a problem and call Agile Support, you provide a URL for Web Client access to your PLM system. This "window" to your system permits data gathering.
2. If the Support person can identify the problem and a quick remedy, he will enter the logging change, attempt to reproduce the problem, and turn off Logging Configuration. The monitor enables resetting your "Log4J logging level" dynamically without stopping and restarting your system's AAS.
   - Use the Quick Search field at the bottom of the page. The debugging information appears in a console window.
   - After collecting the debugging information, Support can exit the **Logging Configuration** page without altering the PLM system being examined.
   - Click the **Refresh** button to update the view of logging data at any time.
   - In the Level column, modify the configuration by selecting from the dropdown list for appropriate classes.
   - Click the **Update Logger** button to commit the configuration change without stopping the system's AAS.
3. If Support does not find a remedy or workaround, the problem can be escalated to Agile Development.
   - "Agile Development" may ask Support to request that the customer turn on certain logs, based on which area of PLM the problem is originating from.
   - The customer can turn on logs without restarting AAS, the information can be retrieved by Support, and the logs can be turned off without restarting AAS.
4. When "Agile Development" has debugged the problem, Support can:

- return to the customer's system to put the change into effect; or,
- communicate the proposed fix to the customer.

# Cache Health Monitor

Agile PLM caches in memory all data related to application configuration in each and every clustered server. This is called the Admin Cache. Any change to the Admin Cache data – on any server in the cluster – triggers a message to be delivered to all the cluster servers, to update the Cache for the data item that changed. Agile PLM uses Java Messaging Service (JMS) to broadcast this message.

It has been found that some changes are not propagated across all servers in the cluster. This happens randomly, that is, the occurrence of non-propagation is not possible to predict. Once it occurs, the solution is to restart all the servers in the cluster. An additional solution to this issue is the utility called **Cache Health Monitor**.

| **Note** | Cache Health Monitor utility is designed as a backup system. The current method of using the JMS server remains as the primary data synchronization mechanism. |
|---|---|

The Cache Health Monitor performs the following actions:

- Lists all modified Administrator nodes and the current synchronization state;
- Commences the synchronization manually or automatically (see Automatic Synchronization using Java Client Administrator on page 44);
- Turns logging of cache synchronization operations on or off;
- Views the log file containing cache synchronization operation messages;
- Purges old entries.

## How the Cache Health Monitor Works

All Administrator (configuration-related) changes are entered to a new database table. At the server startup time, Cache Health Monitor logs the server time as "last update time" (LUT) and monitors all changes made from LUT onwards. If a manual or automatic triggering is initiated, all changes made from the LUT are synchronized (if not already synchronized) and the LUT is updated to the current time.

To access the servers in the cluster, the administrator must log in to each application server directly, using the application server name and port. For example, suppose there is a cluster with three servers:

- app1.abc.com
- app2.abc.com
- app3.abc.com

To monitor and synchronize manually, the administrator would log in to each server using these URLs:

- http://app1.abc.com:7777/Agile/PLMServlet
- http://app2.abc.com:7777/Agile/PLMServlet
- http://app3.abc.com:7777/Agile/PLMServlet

The default Oracle Application HTTP port-"7777"-is shown in the examples; you are advised to change this as appropriate.

### Buttons on Cache Health Monitor Page

The specific application server name is listed on the **Cache Health Monitor** page. This helps identify the server that the user is connected to, especially when the user may want to have one browser session for each server in the cluster.

The buttons on the **Cache Health Monitor** page activate specific events as follows:

**Synchronize** – Synchronizes the data on that particular application server; use when one or more objects are reported as "Not synchronized" in the **Status** column. After the Synchronize button is clicked, synchronization is triggered and the status changes to "Synchronized."

**Refresh** – Refreshes the data on that page.

**Purge** – Purges old entries in the CACHE_SYNC_LOG table in database; the time interval to select entries to purge is configured through the property cache. syncevent.obsoletetime (in minutes) in the file **agile_home**/agileDomain /config/agile.properties. The default value is 144000 minutes, or 100 days. Changing this property requires server restart.

Purge removes all entries in the table that are older than the Purge value. In above case, all entries older than 144000 minutes will be removed.

**Turn Log on / Turn Log off** – Enables or disables logging, works as a toggle switch.

**View Log** – Displays current synchronization log; this log contains information about synchronization process, objects that have been synchronized and failures, if any.

## Automatic Synchronization using Java Client Administrator

To configure automatic cache synchronization, you must enable a background task called "Admin Cache Synchronization Task." This task is configured in **Java Client** > **Admin** > **Server Settings** > **Task Configuration**; it is documented in Task Configuration on page 384.

When you double-click a task in the Task Configuration page, the Task Configuration **General Info** window for that task appears. Admin Cache Synchronization Task has the following properties, including the defaults specific to Cache Health Monitor.

| Property | Description |
|---|---|
| Name | Name of the task being configured. |
| Task Interval | Periodic time interval in minutes after which the server updates the status of the task. Default is 5, meaning all enabled tasks in the node run every 5 minutes. |
| Task Delay Time | How many minutes the event is delayed before it starts to run. Default = 0. |

| Property | Description |
|---|---|
| Task Lookback Window | How far back in minutes the task manager is to look to retrieve the events that can be run now. Default = 0. |
| Task Max Event Number | How many events can be running at the same time on one server. Default = 1. |
| Task Restart Upon Failure | Indicates whether the event should be restarted after it failed. Yes or No. Default = No. |
| Task Load Across Server | Indicates whether the event can be executed at the same time in different servers of a cluster. Yes or No. Default = No. |
| Task Disabled | Indicates whether the task is disabled (Yes) or enabled (No). You may want to set this property to Yes while you are configuring and testing it; when you are ready to enable the task, you must re-set this property to No. Default = No. |

If you do not make frequent changes in Administrator, or if Admin changes are updated on a specific or scheduled time, manual synchronization of the cluster will likely meet your needs. If, however, you frequently change settings in Administrator, and these are entered at many intervals throughout the day, this automatic Admin Cache Synchronization Task is a more structured backup should synchronization fail. In the event of synchronization failure, use Cache Health Monitor (either manual or automatic) until you are able to restart the cluster's servers.

# Logging and Debugging Errors

To provide a mechanism to track the events related to Admin Cache synchronization, logging can be enabled to log all debugging messages. When logging is turned off, only errors and synchronization issues seen by the Cache Health Monitor are logged.

When an Administrator object or a business object is modified on any server in the cluster, a JMS server receives the event details and is triggered to broadcast the message to the other servers.

**Note** While the Cache Health Monitor monitors and synchronizes Administrator objects exclusively (for example, Changes Base Class or Change Orders Class), the logging mechanism logs events related both to Administrator objects and to user-modified business objects (for example, a modified change order). This is helpful for tracking the propagation of changes to business objects across the servers.

The following two messages (each message starts with a time stamp) will be recorded in the log file (when log is turned on).

```
2006-09-20 18:14:08,881 HttpRequestHandler-3096409  <Messenger:DEBUG>
Sending message <class:com.agile.pc.cmserver.event.CMBatchEvent,
content:Batch events:

event class: com.agile.pc.cmserver.event.CMObjectEvent, content:
object: class: 10000 subclass:10141 id:6010147 version:1 action: 4

event class: com.agile.pc.cmserver.event.CMFTSSyncEvent, content: FTS
sync object: class: 10000 subclass:10141 id:6010147 version:0

event class: com.agile.pc.cmserver.event.CMObjectEvent, content:
object: class: 10000 subclass:10141 id:6010147 version:1 action: 1

> to <topic:sync_topic> with property <name=APP_SRV_ID, value=6010141>

2006-09-20 18:14:10,222 HttpRequestHandler-3096409 <Messenger:DEBUG>
```

```
Message was delivered to JMS topic:sync_topic
```

The first message captures the details of the object modified; the second message indicates that the JMS server received the message. JMS then propagates this message and the following three messages are recorded on all servers other than the originating server.

```
2006-09-20 18:14:27,257 ApplicationServerThread-2 <SyncMDBean:DEBUG>
Received message from JMS server.

2006-09-20 18:14:27,257 ApplicationServerThread-2 <SyncMDBean:DEBUG>
Message was sent by server with ID: 6010141, current server id is:
6010121

2006-09-20 18:14:27,257 ApplicationServerThread-2 <SyncMDBean:DEBUG>
Process event: class name -->
com.agile.pc.cmserver.event.CMBatchEvent, type --> 1, content -->
Batch events:

event class: com.agile.pc.cmserver.event.CMObjectEvent, content:
object: class: 10000 subclass:10141 id:6010147 version:1 action: 4

event class: com.agile.pc.cmserver.event.CMFTSSyncEvent, content: FTS
sync object: class: 10000 subclass:10141 id:6010147 version:0

event class: com.agile.pc.cmserver.event.CMObjectEvent, content:
object: class: 10000 subclass:10141 id:6010147 version:1 action: 1
```

Using the above log messages, any unsynchronized data could be traced out.

When manual / automatic (background thread) triggering occurs, even with logging turned off, the following messages are displayed for all Administrator nodes being synchronized.

## Manual Triggering:

```
2006-09-20 13:17:39,512 HttpRequestHandler-394365 <ADictionary:INFO>
Admin cache sync > invalidate node: admin objid: 7000 class id: -1
version: -1
```

## Automatic Triggering:

```
2006-09-20 13:17:39,512 ApplicationServerThread-149 <ADictionary:INFO>
Admin cache sync > invalidate node: admin objid: 7000 class id: -1
version: -1
```

**Note** The difference in the two messages is the source of triggering, in the manual case since we are sending a HTTP request the source is displayed as HttpRequestHandler, in the second case, the source is application itself (background thread) therefore the source is displayed as ApplicationServerThread.

Logging is configured through the log.xml file in agile_home/agileDomain/config directory.

The following element configures the log for Cache Synchronization.

```
<appender name="CacheSyncLog"  class="org.apache.log4j.RollingFile
Appender">
        <param name="File" value="${agile.log.dir}/cachesync.log" />
        <param name="Append" value="true" />
        <param name="MaxFileSize" value="1MB"/>
```

```
        <param name="MaxBackupIndex" value="10"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d\t%t\t&lt;
%c{1}:%p&gt;\t%m\n"/>
            </layout>
    </appender>
```

| Parameter | Description |
|---|---|
| File | Name and path of the log file |
| Append | Append or overwrite to existing file at server startup |
| MaxFileSize | Maximum file size after the current file is archived and new file is created |
| MaxBackupIndex | Maximum number of files that would be archived |

# Deleting Objects in Agile

Java Client and Web Client support “soft” and “hard” deletes. When you soft-delete an object in the database, it is not permanently deleted. A soft-deleted object is marked “Deleted” in the database, and its object number or name is reserved. You cannot create another object with the same type and number or name as a soft-deleted object.

**Note** You *can* undelete a soft-deleted object, if you have the appropriate privileges set by the Agile administrator.

When you run a search, soft-deleted objects do not appear in the search results table. To locate soft-deleted objects, you must run the predefined deleted object searches in the **Recycle Bin Searches** folder.

**Note** You have access to the **Recycle Bin Searches** folder if **Recycle Bin Searches** is included in the **Searches** field of your user profile. If you have the appropriate privileges, you can edit your user profile and add **Recycle Bin Searches** to the list of searches in the **Searches** field.

A hard-deleted object is permanently removed from the database.

**Caution** You *cannot* undelete a hard-deleted object.

To soft-delete, hard-delete, or undelete an Agile object:

- You must have the appropriate Delete or Undelete privileges for that object.
- The object must meet the conditions that allow it to be deleted or undeleted.

  In general, if deleting or undeleting an object would cause the compromise of data integrity, the delete or undelete action is disabled.

You cannot delete an object unless all of its relationships and subscriptions are removed.

For detailed information about deleting specific *object* types see the *Deleting Agile Objects* section

of *Getting Started with Agile PLM*.

**To soft-delete an object in Web Client:**

1. Select and open the object you want to delete.
2. Choose **Actions > Delete**, and respond **OK** to the confirmation prompt.

The object is soft-deleted.

**To hard-delete a soft-deleted object (if you have the appropriate privileges) in Web Client:**

1. Run the appropriate Deleted <*objects*> search from the **Recycle Bin Searches** folder.
2. Select the one or more object rows in the search results table.
3. Click the **Delete** button.
4. In response to the warning prompt, choose **Continue** and then click **Finish**.
5. Or, you can hard-delete one object at a time:
   - Open the object.
   - Choose **Actions > Delete**, and respond **OK** to the confirmation prompt.

**Note** You can click the **Navigator** button in the search results table to display the search results object links in the left pane **NAVIGATOR** drawer.

**To undelete an object in Web Client:**

1. Run the appropriate Deleted <*objects*> search from the **Recycle Bin Searches** folder.
2. Open the deleted object you want to restore.
3. Choose **Actions > Undelete**.

**Note** You can click the **Navigator** button in the search results table to display the search result object links in the left pane **NAVIGATOR** drawer.

# Chapter 5

# Classes

**This chapter includes the following:**


- Configuring Business Objects with Agile PLM Classes ..... 49
- Agile PLM Class Structure ..... 50
- Agile PLM's Installed Routable Object Types ..... 52
- Agile PLM's Installed Nonroutable Object Types ..... 54
- Customizing Lifecycle Phases ..... 56
- Configuring Subclasses ..... 58
- Configuring Tabs ..... 63
- Modifying the Format of Tabs ..... 64
- About Attributes, Flex Fields, and Read-Through Fields ..... 70
- Defining Attributes ..... 75
- Configuring Attributes ..... 82
- Using Custom Icons for Subclasses ..... 84
- Quick Access to Objects and Files ..... 85


This chapter describes the predefined business classes delivered with Agile PLM, and how to create subclasses, assign autonumber sources, and tailor the arrangement of tabbed windows and formatted fields for your product data.

# Configuring Business Objects with Agile PLM Classes

The **Classes** node provides a framework for classifying objects in Agile PLM. Everything that an Agile PLM user creates is an object. As administrator, you can define new types of business objects, or modify the default classes and subclasses.

| **Note** | The default classes that are available in Agile PLM depend on which licenses your company purchased. Unavailable classes appear in red type on the Classes page. |
|---|---|

Each class represents a kind of action (business process) or kind of entity (business object). Every class has at least one default subclass. Every object created in Agile PLM is an instance of a subclass.

Before you start configuring Agile PLM, you should consider what types of business objects are required by your company. You can then tailor Agile PLM to your requirements. For example, you might want to create several new subclasses in the Parts class to organize the kinds of parts that your company manufactures or purchases.

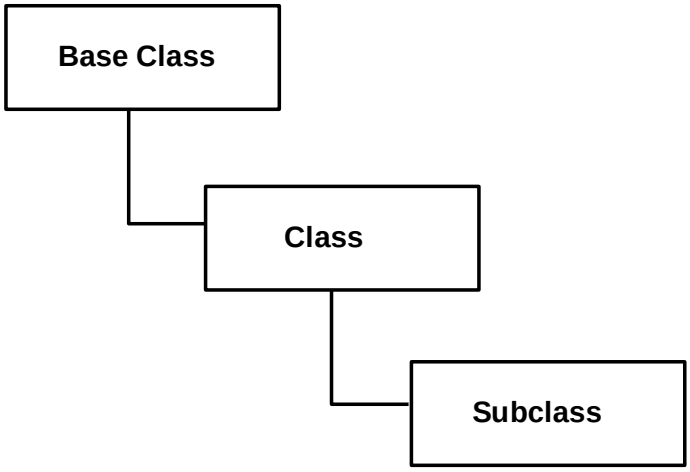
The Agile PLM object framework is ready to use out-of-box, or you can tailor it as follows:

- Create additional subclasses
- Modify classes and subclasses as needed

- Create new attributes for classes (Page Two) and subclasses (Page Three) as needed
- Modify the properties of attributes
- Display or hide attributes or tabs
- Reorder table columns or fields

# Agile PLM Class Structure

Agile PLM classes have the following three levels:




- **Base Class** — The highest level object. Base class is the parent object of classes. You cannot create or delete base classes, but you can rename them. Examples of provided base classes are Users, Items, and Changes.
- **Class** — The primary level of organization in Agile PLM. Also called *main class*. Inherits the general properties and process extensions of its base class. Classes are the organizing framework for the creation of subclasses. You cannot create or delete classes, but you can rename or disable them. Examples of provided classes are Parts, Change Orders, and Change Requests classes.
- **Subclass** — Inherits all properties and attributes of its parent class. Subclasses arrange the information and data that describes a specific business object in Agile PLM. You can rename subclasses. You can also create a subclass in any class except Users, Automated Transfer Orders, Requests For Quote, RFQ Responses, and Standard Reports. You can delete or disable any subclass except those descended from Items, Changes, and File Folders base classes. Examples of provided subclasses are User, Part, Document, ECO, and ECR.

| | |
|---|---|
| **Caution** | Before renaming any classes, particularly the base classes, make sure you become familiar with the default classes provided with Agile PLM. Class names must be unique. |

# Routable and Nonroutable Objects

Each class represents a specific process or kind of entity:

- **Routable objects** are created from classes that represent *processes*, such as a change in the change control process, or a transfer order, or a product service request. These objects can be routed to Agile PLM users for approval or other input via workflows (see Workflows on page 129).
- **Nonroutable objects** are created from classes that represent *entities* or things, such as parts, sites, RFQs, users, prices, or reports. These objects can be "flagged" for progress via lifecycle phases (see Customizing Lifecycle Phases on page 56). The nonroutable objects—especially parts and documents—can be changed by the process of information-gathering via workflows, which are created through such routable objects as change orders or change requests.

**Note** The File Folders class has a default, non-editable workflow with a single Review status and no other statuses. This provides file folder objects with a **Routing Slip** tab, which allows approvers to sign off and comment on a file folder. However, file folder objects also have lifecycle phases and are considered nonroutable.

# Object Types in Agile PLM

In PLM, Object Types are each and every one of the rows in the **Classes** node. Object types are organized by the hierarchy of Base Class (a kind of template), Class (more detailed, generally specific to a PLM solution), and Subclass (most specific organizing group).

So, for example, let `ECO987` be an individual business object: it is a single instance of the object type *ECO* (a subclass), which is derived from the object type *Change Orders* (a class), which in turn is derived from the object type *Changes* (a base class).

Therefore, each new subclass that you create is also an object type. So, *Items*, *Parts*, and *Part* are each an object type, and if you add subclasses called *Capacitor* and *Resistor*, each of these are also object types.

"Object Type" is a significant field in such areas of PLM administration as **Criteria**, **Privileges**, **Notifications**, and **Event Management**. For these and many other functions, it is a crucial element to inform the PLM system precisely what kind of object is to be accessed and subject to user actions.

# Viewing Agile PLM Classes

**To view an Agile PLM class or subclass:**

1. On the Admin tab, under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the class or subclass you want to view.
3. Now click the **User Interface Tabs** tab. Double-click any row to bring up that tab's information:

   The **General Information** tab displays the name and description of the UI tab and whether it is visible.

   Once you have opened one tab, a dropdown list allows you to switch to any of the other tabs for that object.

# Agile PLM's Installed Routable Object Types

Your company may be licensed to a subset of the list of object types in **Classes** node.

The following tables list all the possible installed object types in Agile PLM. The first table lists the routable objects, the second table lists the nonroutable objects.

Routable objects direct or recommend changes to nonroutable objects. These classes have a default workflow for changes to seek approval from other users. Objects from these classes can be changed without approval.

Nonroutable Objects are not routed to Agile PLM users with workflows; objects from some of these classes, however, are changed by a user submitting a workflow (from routable classes) for approval from other users.

| Base class | Class | Subclass | Description |
|---|---|---|---|
| **Changes** | Change Orders | ECO | Directives to change an item; can advance the revision (“rev”) of an item |
| | Change Requests | ECR | Requests for a change to an item |
| | Deviations | Deviation | Directives to temporarily substitute one item for another |
| | Manufacturer Orders | MCO | Changes to AML data, such as information about manufacturers or manufacturer part numbers |
| | Price Change Orders | PCO | Directives to change a published price; can advance the revision of a published price |
| | Site Change Orders | SCO | Changes to BOM and AML information for a specific site |
| | Stop Ships | Stop Ship | Directives to stop shipping/using an item |
| **Declarations** | Substance Declarations | Substance Declaration | Seeks compliance information for each substance within the specification |
| | Part Declarations | Part Declaration | Seeks part-level compliance information and other composition header-level information (manufacturing parameters) |
| | JGPSSI Declarations | JGPSSI Declaration | Seeks compliance information (weights) according to the JGP standard. |
| | Homogeneous Material Declarations | Homogeneous Material Declaration | Seeks complete breakdown of parts on the Bill Of Substances and compliance information at the |

| Base class | Class | Subclass | Description |
|---|---|---|---|
| | | | homogeneous material level |
| | Supplier Declarations of Conformance | Supplier Declaration of Conformance | Seeks compliance with specifications from customers and government agencies |
| | IPC 1752-1 Declarations | IPC 1752-1 Declaration | A Joint Industry Guide (JIG) material composition declaration for electronic products |
| | IPC 1752-2 Declarations | IPC 1752-2 Declaration | A homogeneous material composition declaration for electronic products |
| **Packages** | packages | Package | Packages of data to share with partners |
| **Product Service Requests** | Problem Reports | Problem Report | Quality incidents with items or products |
| | Non-Conformance Reports | NCR | Quality conformance issues with items or products |
| **Projects** | Activities | Phase; Program; Project; Task | Components of project planning in Product Portfolio Management; activities are time-based objects to which resources can be assigned |
| | Gates | DG1, DG2, DG3, DG4, DG5, DG6; Gate; Milestone | Project management milestones in Product Portfolio Management; gates identify cross-PLM deliverables of the product development process to enable executive reviews of projects |
| **Quality Change Requests** | Corrective and Preventive Actions | CAPA | Requests for corrective actions and preventive actions |
| | Audits | Audit | Proactive reviews of business processes |
| **Transfer Orders** | Automated Transfer Orders | ATO | Transfer or publication of a product record that is automatically triggered by a workflow |
| | Content Transfer Orders | CTO | Transfer or publication of a product record that is manually triggered |

# Agile PLM's Installed Nonroutable Object Types

| Base class | Class | Subclass | Description |
|---|---|---|---|
| **Non-Routable Objects**: objects in these classes are not routed to Agile PLM users with workflows; objects from some of these classes, however, are changed by a user submitting a workflow (from routable classes) for approval from other users. | | | |
| **Customers** | customers | Customer | Clients of the company |
| **Discussions** | discussions | Discussion | Informal, threaded dialog |
| **File Folders** | File folders | File Folder<br>Markup | Objects that include files or URLs; this class includes all file folder objects except historical report file folders |
| | Designs | Design | Objects that permit building design structures in the CAD environment |
| **Items** | Documents | Document | Specifications, blueprints, manufacturing data, and so forth |
| | Parts | Part | Parts manufactured within the company, or provided by manufacturers or suppliers and given internal part numbers |
| **Manufacturers** | manufacturers | Manufacturer | Qualified manufacturers |
| **Manufacturer Parts** | Manufacturer parts | Manufacturer Part | Parts provided by manufacturers |
| **Part Groups** | Part groups | Commodity<br>Part Family<br>Item Group | Containers of other parts (items or manufacturer parts) that share such properties as mass or composition information.<br>**Note** This object was named "Commodities" in previous releases of Agile PLM. |
| **Prices** | Quote Histories | Quote History | Organizes bid prices from RFQ responses; cannot be revised by a PCO (see grouping of Changes in preceding table) |
| | Published Prices | Published Price; Contract | Organizes prices of the company's products; can be revised by a PCO (see grouping of Changes in preceding table) |
| **Reports** | Standard Reports | Administrator Report; Standard Report | The out-of-box reports for administrators (Administrator Reports) and users (Standard Reports include Products, Sourcing, |

| Base class | Class | Subclass | Description |
|---|---|---|---|
| | | | Quality, Process, Personal, and Global reports) |
| | Custom Reports | Custom Report | Reports created and used within company |
| | External Reports | External Report | Reports created outside Agile PLM |
| **Requests For Quote** | Requests for quote | RFQ | Requests for quote, which are assembled from sourcing projects and sent to suppliers for formal bids |
| **RFQ Responses** | RFQ responses | RFQ Response | Bids, that is, responses from suppliers to your company's RFQs |
| **Sites** | sites | Site | Manufacturing locations within the company, or closely partnered with the company |
| **Sourcing Projects** | Sourcing projects | Sourcing Project | Work preparatory to creating RFQs and capability for analysis across multiple RFQs |
| **Specifications** | specifications | Specification | Lists of banned substances (or substances of concern) and their threshold values |
| **Substances** | substances | Substance | A single chemical element used in composition of items, manufacturer parts, and part families |
| | Materials | Material | A compound chemical, a substance consisting of multiple substances |
| | Subparts | Subpart | A subunit of a component, used to get to the homogeneous material level to collect compliance information |
| | Substance Groups | Substance Group | A group of multiple substances, with a base substance that is what legislation is interested in, for example, "Lead and Lead Compounds" |
| **Suppliers** | suppliers | Broker; Distributor; Component Mfr.; Contract Mfr.; Mfr. Representative | Qualified suppliers of manufacturer parts; used by PCM (which uses the out-of-box subclasses) and PG&C solutions |
| **Users** | users | User | Individuals using the Agile PLM system |
| **User Groups** | User groups | User Group | Groups of people using the Agile PLM system, e.g., departments, |

| Base class | Class | Subclass | Description |
|---|---|---|---|
| | | | teams, site-specific groups |

## Disabling Object Types

When a base class is disabled, the child classes and subclasses are automatically disabled. When a class is disabled, the child subclasses are disabled. When a base class, class, or subclass is disabled, it is “grayed out” in the user interface for Java Client or Web Client. Examples of where the disabled class would be unavailable are the **Create** menu and search options. Existing searches and criteria are not deleted or modified if a class is disabled; existing objects of disabled object types remain in the Agile PLM database.

**Note** Item, Change, User, and File Folder base classes cannot be disabled, nor can their child classes or out-of-box subclasses.

# Customizing Lifecycle Phases

Lifecycle phases describe the valid phases for classes whose objects are nonroutable, that is, they do not follow workflows. Items, sites, manufacturers, manufacturer parts, prices, sourcing projects, RFQs, RFQ responses, suppliers, customers, prices, commodities, specifications, substances, and file folders all have lifecycle phases.

Lifecycle phases at the base class level are inherited by classes and subclasses. The lifecycle stamp is automatically created for you using the name and color you specified for each lifecycle phase.

## Modifying a Lifecycle Phase

You can modify lifecycle phases for the following base classes:

- Commodities
- Customers
- File Folders
- Items
- Manufacturer Parts
- Manufacturers
- Prices
- Specifications
- Substances
- Suppliers

**Note** You cannot delete or rename the Preliminary lifecycle phase for items. Objects in the Sites, Sourcing Projects, Requests for Quote, and RFQ Responses classes have read-only lifecycle phases that cannot be renamed or deleted.

**To modify** lifecycle phase **settings:**

1. Select the base class you want to work with, and select the **Lifecycle Phases** tab. The Lifecycle Phases window appears.
2. Double-click the lifecycle phase you want to modify.
3. In any editable field, enter a new value or select from the dropdown list.
4. When you have finished, click **Save**.

# Adding a Lifecycle Phase

You can add lifecycle phases to base classes, classes, or subclasses.

**To add a lifecycle phase:**

1. Select the class you want to work with, and select the **Lifecycle Phases** tab. The Lifecycle Phases window appears.
2. Click the **New** button. The Define Lifecycle Phase dialog box appears.
3. Enter a name, API name and description for the lifecycle phase. The name can be up to 29 characters.
4. In the **Enable** field, select Yes or No from the dropdown list.
5. For items, there is an additional field, **AddLifeCyclePhaseOnBomRule**. Select Allow, Disallow, or Warning (the default).

   If you select Disallow, users will be prevented from adding items currently in this lifecycle phase to an item's BOM table.
6. When you are done, click **OK**.

**Note** To select a different color for the lifecycle phase, modify it on the appropriate lifecycle phases tab.

# Removing a Lifecycle Phase

You can remove a lifecycle phase only from the level in which it was created. When you remove a lifecycle phase from a class, it is also removed from all subclasses that inherit from the class. For example, if you create a new lifecycle phase in the Items base class, you can open the Items base class later to remove it. The lifecycle phase is also removed from the Parts and Documents classes, and the Part and Document subclasses.

**To remove a lifecycle phase:**

1. Select the class you want to work with, and select the **Lifecycle Phases** tab. The Lifecycle Phases window appears.
2. Select the lifecycle phase you want to delete.

3. Click the **Delete** button.
4. Click **OK** in the confirmation message to delete the lifecycle phase.

   The lifecycle phase is deleted and is no longer available as a selection in Agile PLM.

If you delete a lifecycle phase that has already been used to release an item in Agile PLM, it still indicates that phase, but users cannot assign that phase in the future.

**Note** You cannot delete the Preliminary lifecycle phase for items.

# Configuring Subclasses

When users create an object in Agile PLM, they first specify its subclass, such as ECO, Part, Capacitor, and so on. These subclasses are created and defined in the **Classes** node. The defined Agile PLM classes are each equipped with at least one subclass. You can create more subclasses in any class. For example, the Parts class can have subclasses called Capacitors, Resistors, Diodes, and so on.

You can decide to use just the default subclasses, define new subclasses, or rename subclasses.

You can delete a subclass as long as it has not been used to create any of these objects from the administrative nodes:

- Search criteria
- An object in Java Client or Web Client
- A privilege mask

**Note** You cannot create new subclasses for the following classes:

- Users class
- Automated Transfer Orders class
- Requests For Quote class
- RFQ Responses class
- Standard Reports class

**Important** Setting up new subclasses in Agile PLM requires a lot of planning. If you set up new subclasses without an overall plan, you may find in a few weeks that you want to "start over," only now new subclasses are in use and will not be easy to remove from the system. It is better to become proficient with Agile PLM objects using the out-of-box classes and subclasses while you develop an overall plan that covers your company's needs.

## Before You Begin

Consider the following details before creating a new subclass:

- For which Agile PLM class will you create a subclass?
- What name will you give the new subclass? Remember that a subclass name must be unique, and cannot be used more than once in the same Agile PLM system.
- Do you need to track product data specific to this subclass? If so, you may also want to set up a **Page Three** tab for this subclass.
- If you plan to create a Page Three for the new subclass, you will need to adjust some users' roles and privileges to allow Page Three modification.

For new subclasses that have autonumbering, you should consider the following:

- Will you permit users to assign numbers manually for the new subclass?
- Will you be creating new autonumber sources for the subclass, or will you use existing autonumber sources?
- If you plan to use existing autonumber sources, which ones?

## Viewing Subclass Properties

You can see how a subclass is configured by viewing its properties. For example, to see what number sources are assigned to a Change Orders subclass, you can view the subclass properties in Administrator.

**To view the properties of a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the name of the subclass you want to view.

   The properties of the subclass are listed on the **General Info** tab.

## Changing the AutoNumber Source Property

Subclasses that can be autonumbered have an AutoNumber Source property. Since subclasses in the Manufacturers, Manufacturer Parts, and Sites classes do not use autonumbering, they do not have an AutoNumber Source property.

The Agile PLM administrator can change the AutoNumber Source property for a subclass to do the following:

- Require automatic number assignment for the subclass, so that manual number assignment is prevented.
- Assign one or more different autonumber sources for the subclass.
- Remove autonumber source assignments from the subclass.

**To change the autonumber sources used for a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the name of the subclass you want. The subclass properties window appears.
3. Next to the **AutoNumber Source** field, click . The List Chooser dialog box appears.
4. To move a number source from the **Choices** list to the **Selected** list, select it, and click the right arrow. You can also move the number source by double-clicking it.

   You can select more than one number source.
5. Click **OK** to close the dialog box.
6. If you want to create a new autonumber—more customized than those offered by the automatic sequencing of autonumbers—click the **New AutoNumber** button and follow the instructions in the dialog box. For more information, see Creating a New Autonumber Source on page 114.
7. If you want to prevent users from entering an object number manually, select No in the **Autonumber Required** dropdown list. For more information, see the next section.
8. Select Yes or No in the **AutoGenerate** dropdown list.
9. Click **Save**.

# Preventing Manual Number Assignment

If you decide to require autonumbering for a specific subclass, you can prevent users from entering numbers manually. You can require autonumbering for any subclass that can be autonumbered.

| | |
|---|---|
| **Note** | Several Agile PLM subclasses, such as Manufacturer and User, do not allow autonumbering. |

**To prevent manual number assignment for a specific subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the name of the subclass you want. The subclass properties window appears.
3. In the **Autonumber Required** dropdown list, select **Yes**.
4. Click **Save**.

# Changing a Subclass Name or Description

You can change the name of any subclass. If you change the name of a subclass that has already been used to create objects in Agile PLM clients, the subclass name for the existing objects is changed to reflect the new name. You can also enter a different description for the subclass; a description is optional.

| | |
|---|---|
| **Caution** | Although all characters are supported within Agile PLM, and should not create issues within Agile's clients, it is recommended that you avoid using the following characters when renaming or creating subclasses, due to issues that could potentially arise in integrations or custom applications written on the SDK. |

, comma . period (dot) : colon ; semicolon = equal sign

**To change the name of a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the name of the subclass you want. The subclass properties window appears.
3. Enter a new name and description in the **Name** and **Description** fields.
4. Click **Save**.

   The subclass name is changed for all objects in the subclass. The new name appears as a selection in the **Type** dropdown list in the New Object dialog box in Java Client, and the new name appears as a selection in the **Type** dropdown list in the Create New Change Order window in Web Client.

# Changing the Site-Specific BOM Property

On the subclass properties window, there is a property called **Site-Specific BOM**. The value for the property is either **Allow** or **Disallow**.

- **Allow –** The subclass may be added to a site-specific portion of a BOM.
- **Disallow –** The subclass may not be added to a site-specific portion of a BOM.

This property determines whether a Parts or Documents subclass may be part of a site-specific BOM.

For example, let the Manufacturing Item subclass have the Site-Specific BOM property set to Allow, and subclass Part have it set to Disallow. A user at the Singapore site can add any manufacturing item to the site-specific section of the BOM for Singapore, but not add a part to the site-specific section of the BOM for Singapore. A Global user can add either a manufacturing item or a part to the Global section of the BOM.

The default value for all newly created Parts and Documents subclasses is Allow.

**To change the Site-Specific BOM property of a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the name of the subclass you want. The subclass properties window appears.
3. In the **Site-Specific BOM** field, select Allow or Disallow. Click **Save**.

# Creating a New Subclass

This is the procedure for creating a new subclass. Please review the points in Before You Begin on page 59. Remember you can keep a subclass disabled while you run some tests before releasing it to your users.

**To create a new subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Click the **New Subclass** button. The New Subclass dialog box appears.
3. Select a class from the dropdown list.

4. Fill in the **Name**, **API Name** and **Description** fields, and select **Yes** or **No** in the **Enabled** list. You must use a unique subclass name.
5. Next to the **AutoNumber Source** field, click . The selection dialog box appears.
6. To move a number source from the **Choices** list to the **Selected** list, select it, and click the right arrow. You can also move the number source by double-clicking it.

   You can select more than one number source.
7. Click **OK** to close the dialog box.
8. If you want to create a new autonumber—more customized than those offered by the automatic sequencing of autonumbers—click the **New AutoNumber** button and follow the instructions in the dialog box. (This task is detailed in Creating a New Autonumber Source on page 114.)
9. If you want to prevent users from entering an object number manually, select **No** in the **Autonumber Required** dropdown list. (For more information, see Preventing Manual Number Assignment on page 60.)
10. Select **Yes** or **No** in the **AutoGenerate** dropdown list.
11. If you selected the Parts or Documents class in step 3 above, set **Site-specific BOM** to Allow or Disallow. (For more information, see Changing the Site-Specific BOM Property on page 61.)
12. When you have specified all the settings, click **OK**.

    The new subclass appears under the class name in the Classes window.

| | |
|---|---|
| **Note** | Number sources are created and configured in the **AutoNumbers** node. For instructions, see About Autonumber Sources. |

# Deleting a Subclass

You can delete any subclass that has not been used to create a saved search, an object, or reusable criteria.

**To delete a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click the subclass you want to delete. The subclass properties window appears.
3. Click the **Delete** button.
4. In the confirmation dialog box, click **OK**.

   The subclass no longer appears under the class name on the Classes window.

| | |
|---|---|
| **Note** | If you attempt to delete a subclass that has already been used to create a search, an object, or a reusable criteria, an error message states that the object is already in use. |

You can also delete a subclass by displaying it and then choosing **Delete** from the Actions menu.

# Configuring Tabs

When users open an object in Agile PLM clients, they see a tabbed page, or window of tabs. From the **Classes** node, you can enable (make visible) or disable (hide) most object tabs. You can also rename any object tab. You configure optional **Page Two** and **Page Three** tabs by enabling the tabs to make them visible.

Although you can hide most object tabs, you cannot disable or hide the following tabs:

- The first tab of any object, specifically:
  - **Cover Page** tab of any change, package, PSR, QCR, transfer order, or project
  - **General Info** tab of any manufacturer, manufacturer part, user, user group, site, customer, discussion, price, or report
  - **Title Block** of any item or file folder
- **Affected Items** tab of any change and the **Affected Prices** tab of PCOs
- **Action Items** tab of discussions
- **Selected Content** tab and **Where Sent** tab of any transfer order—in fact, no tab of ATOs can be disabled or hidden
- **Files** tab of file folders (not historical report file folders)
- **Response List** tab of RFQs and RFQ responses
- **Replies** tab of discussions
- **Team** tab and **Workflows** tab of projects

If you want to configure the attributes that appear on a tab, you must go to the setup window for that tab. See Defining Attributes on page 75.

## About Page Two Tabs

If you need to document additional information for change orders, for example, you can enable an extra tab called a **Page Two** tab by default. This tab then appears next to the **Cover Page** tab for all subclasses in the Change Orders class. You can use the default name **Page Two** for this tab, or you can give it a more specific name.

**Note** In Web Client, Page Two attributes are displayed on the **Cover Page** tab, under the heading **Page Two** (any custom tab name is not displayed). The user can scroll down to see the additional “Page Two” information.

## About Page Three Tabs

If you need to record additional information that is specific to a subclass, you can enable an extra tab called a **Page Three** tab by default. For example, you can set up a **Page Three** tab for the subclass Capacitors and create a field called Insulation (or multiple fields as needed), where users record information about insulation for each capacitor object created. You can use the default name **Page Three**, or you can give the tab a more specific name.

| Note | In Web Client, Page Three attributes are displayed on the **Cover Page** tab, below the Page Two attributes, under the heading **Page Three** (any custom tab name is not displayed). The user can scroll down to see the additional “Page Three” information. |
|---|---|

## Showing, Hiding, or Renaming a Tab

You can customize the Agile PLM client user interface by showing, hiding, or renaming the tabs in classes.

| Note | If you hide the **BOM** tab or the **Manufacturers** tab, the buttons under the **Redline BOM** and the **Redline Manufacturer** tabs are disabled. |
|---|---|

The name and visibility of any tab depends on the settings at the *subclass* level. For example, in the Change Orders class, if you set **Page Two** tab to Name = P2, Visible = No, and in ECO subclass, if you set **Page Two** tab to Name = P2 of ECO, Visible = Yes, then in Java Client or Web Client, the **Page Two** tab will be visible and appear as **P2 of ECO**.

**To show, hide, or rename a tab:**

1. In the Classes window, locate the tab you want and double-click it to display the setup window for that tab.

   For example, click **Change Orders Class**, display the **User Interface Tabs** tab, then double-click **Workflow** to display the setup window for the change order **Workflow** tab.

2. If you want, on the **General Information** setup tab, enter a new name for the tab.

   The tab, with its new name, appears in Java Client. In Web Client, the tab data appears on the first object tab, for example, the **Cover Page** tab.

   | Note | When you change the name of **Page Three** tabs for Content Transfer Order (CTO) subclasses, the modified **Page Three** tab name does not appear in the Content Service filters. The name **Page Three** always appears in the filters. |
   |---|---|

3. In the **Visible** dropdown list, select **Yes** to make the tab visible in Agile PLM clients.
4. When you have finished your modifications, click **Save**.

   | Note | If you hide a **Page Two** or **Page Three** tab, any attributes configured for the tab are still visible as search selections in Java Client or Web Client unless you also hide them. See Removing a Page Two or Page Three Tab on page 70. |
   |---|---|

# Modifying the Format of Tabs

Each object tab uses one of two different formats to display data in Agile PLM clients:

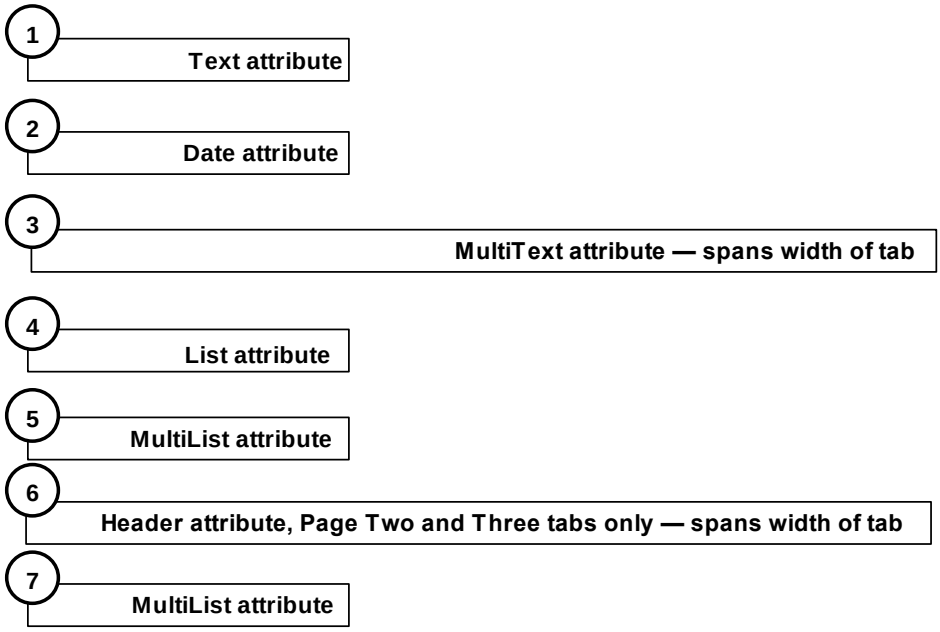
- **Form format** — Attributes appear on the tab in rows and columns, according to the attribute order, lowest number first, left to right, then top to bottom. Each attribute field is labeled. Web Client has two columns. In Java Client, the number of columns varies depending on the size of the object window.
- **Table format** — Attributes appear on the tab in a table. The attribute names appear as column

headings in the table.

You can modify the order in which object attributes appear on object tabs in both forms and tables.

# Modifying Attribute Order on a Form Tab

In Web Client, on a form type tab, the attributes appear in one column. You can determine the order in which they appear. Starting with the lowest numbered attribute, the attributes appear from top to bottom. MultiText attributes span the width of the tab. Header attributes, which appear on **Page Two** and **Page Three** tabs only, also span the width of the tab. The following figure shows an example of how attributes appear on a form tab in Web Client.




In Java Client, attributes appear on the tab in rows and columns, according to the attribute order, lowest number first, left to right, then top to bottom. The number of columns changes as the user resizes the object window. No matter how many columns are in an object window, the attributes are always displayed by attribute order. Header attributes (Page Two and Page Three) span the width of the object window.

**To modify the order of attributes on a form tab:**

1. On the **User Interface Tabs** tab of the setup window for the class, locate the tab you want, and double-click it to display the setup window for that tab.

   For example, double-click Change Orders, then double-click Cover Page to display the setup window for the change order **Cover Page** tab.

2. Click the **Attributes** tab.
3. Click the **Order visible attributes** button. The Define Attribute Order dialog box appears. All the object tab attributes with Visible property set to Yes appear in the list of attributes.
4. Click an attribute name to highlight it.

5. Click the up arrow to move the highlighted attribute up one position in the list. Click the down arrow to move the highlighted attribute down one position in the list.
6. Continue highlighting attributes and moving them up or down in the list until they are in the order you want.
7. When you are finished, click **Save**.

   The attributes appear on the **Attributes** setup tab in the order you specified.

   To verify that the order was changed, open an object in Java Client or Web Client where the tab appears. Click the tab to view the form or table and confirm the change.

# Modifying Attribute Order on a Table Tab

In both Web and Java Clients, on a table-type tab, the attributes appear in a table; the attribute names are table headings for each column. You modify the order of attributes on a table tab the same way you do on a form tab.

# Designing a Page Two or Page Three Tab

You can create extra tabs for Java Client. When enabled, these tabs are always displayed behind the first or second tab, and are named Page Two or Page Three by default. You can change these tab names to something more specific if necessary.

| | |
|---|---|
| **Note** | In Web Client, Page Two and Page Three information is displayed below the Cover Page information on the Cover Page tab. Users scroll down to see this information. |

While Page Two tabs record supplemental information for an entire class of objects, such as the Change Orders class, Page Three tabs record information specific to individual subclasses, such as ECOs or Capacitors.

The Page Two tab's built-in Create User feature is described next.

## About the Page Two Create User Attribute

Each **Page Two** tab has a Create User attribute that allows users with a Create privilege, but not a Read privilege, to create an object. Users with their name in the Create User list can then also open and read any pending or preliminary objects they have created.

To activate the Create User feature for a class of objects, such as Change Orders, you must first make the Create User attribute visible on the **Page Two** tab for that class. Then the attribute can be used both to display the name of the Create User on the tab, and to create criteria in the **Criteria** node.

For example, if you set the subclass criteria for a read privilege to Create User Equal To $USER, then the privilege mask will be enabled when the current user is also the user who created the object. For more information about Create User criteria, see Applying Create User Criteria on page 246.

# Guidelines for Designing Page Two and Page Three Tabs

There are two stages for designing a **Page Two** or **Page Three** tab. This section outlines the process

for these two stages, and then gives you detailed instructions for each step in the process. Also, see Before You Begin on page 114.

The following is a summary of the two stages in designing a **Page Two** or **Page Three** tab.

**Stage I: Preparing Page Two or Page Three Attributes**

1. Make the tab visible and name the tab.
2. Make the attributes visible, and name them.
3. Assign length properties to text attributes.
4. Assign other attribute properties, such as drop-down lists or defaults.

**Stage II: Setting Up the Tab**

Set the attribute order.

The following sections provide detailed instructions for creating a **Page Two** or **Page Three** tab.

## Stage I: Preparing Page Two or Page Three Attributes

The first stage in designing a **Page Two** or **Page Three** tab is to:

- Make the tab visible and name the tab. This work is done from the Page Two or Page Three **General Information** setup tab. For instructions, see Showing, Hiding, or Renaming a Tab on page 64.
- Select the attributes that you want to appear on the tab, make them visible, and give them names that describe the fields and their use. This work is done from Page Two or Page Three **Attributes** setup tab.

**Note** Attribute names are the criteria for Java Client or Web Client searches. By making attribute names both specific and unique, you increase the efficiency of Agile PLM client searches. If you modify the name of an attribute once a system is in production, you may affect existing searches.

## Before You Begin

Consider the following details before designing a new tab.

- Do you want to create an extra tab for an entire Agile PLM class (**Page Two**), or a tab that's specific to a particular subclass (**Page Three**)?

  If you are creating a **Page Three** tab, remember that you also need to create privilege masks that enable users to modify attributes for the specific subclass.

- Do you plan to use Agile ChangeCAST to transfer information to an ERP system? If so, the information must be on a **Page Two** tab. Page Three data cannot be transferred using ChangeCAST.
- What do you want to name the tab?
- Which attributes will appear on the tab?
- What are the properties for each attribute, including name, dropdown list values, default values, the format for text fields, and so on. See Defining Attribute Properties on page 77.
- Where will each field be placed (located under a heading) on the tab in Web Client?
- What are the length requirements (in characters) for each text field?
- What will the tab order be, that is, the sequence in which fields are activated when a user presses the Tab key? (This is determined by the attribute order.)

For more information about attributes, see Recognizing Attribute Types on page 76.

**To make the tab visible, and configure the attributes you want to use:**

1. Make the tab visible and name the tab. For instructions, see Showing, Hiding, or Renaming a Tab on page 64.
2. Locate the attributes you want, and double-click them to open the setup window.

   For detailed instructions, see Modifying Attributes on page 82. For **Page Two** tab attributes, double-click the class, for example, Change Orders. For **Page Three** tab attributes, double-click the subclass, for example, **ECO**.
3. Enter a new name for each attribute.
4. Make the attributes visible; select Yes in the **Visible** dropdown list.
5. Set the appropriate properties for each attribute type (see Defining Attribute Properties on page 77).
   - For List and MuliList attributes, define the list selections (see Configuring Lists on page 97).
   - For Text and MultiText attributes, set the **MaxLength** and **Include Characters** properties.
   - For Numeric attributes, set the **MinValue**, **MaxValue**, and **Scale** properties.
   - For Heading attributes, enter a heading **Name** (note: headings are used in Web Client only).
   - For Date attributes, set the **Default Value** date using the calendar utility.
   - For Money attributes, set the **Default Value**.

6. If appropriate, set a default value for each attribute. (See Setting Default Values on page 83.)
7. When you are finished configuring the attributes, click **Save**.

Once you have completely defined the attributes for your **Page Two** or **Page Three** tab on the **Attributes** tab, you define the attribute order to set up the physical appearance of the tab.

## Stage II: Setting Up the Tab

After selecting the attributes that will appear on the **Page Two** or **Page Three** tab and establishing their properties, you need to arrange the fields on the tab. You do this by using the **Order visible attributes** button on the **Attributes** setup tab.

**Note** For detailed information about setting attribute order, see Modifying Attribute Order on a Form Tab on page 65.

**To set up the order the attributes appear on the tab:**

1. Click the **Order visible attributes** button. The Define Attribute Order dialog box appears.
2. Click an attribute name to highlight it.
3. Click the up arrow to move the highlighted attribute up one position in the list. Click the down arrow to move the highlighted attribute down one position in the list.
4. Continue highlighting attributes and moving them up or down in the list until they are in the order you want.
5. When you are finished, click **Save**.

### *Field Tab Order*

The *tab order* refers to the sequence in which the fields in an Agile PLM client **Page Two** or **Page Three** tab is activated when a user presses the Tab key.

The field tab order is determined by the attribute order, as described above.

## Using Heading Attributes

Java Client displays actual tabs for **Page Two** and **Page Three** tabs. However, Web Client displays the Page Two and Page Three information on the first object tab (for example, **Cover Page**, **Title Block**, or **General Info** tab). When viewing an object with Page Two and Page Three information, the user scrolls down to see that information. For example, the Page Two information is displayed below the Cover Page information. The Page Three information is displayed below the Page Two information.

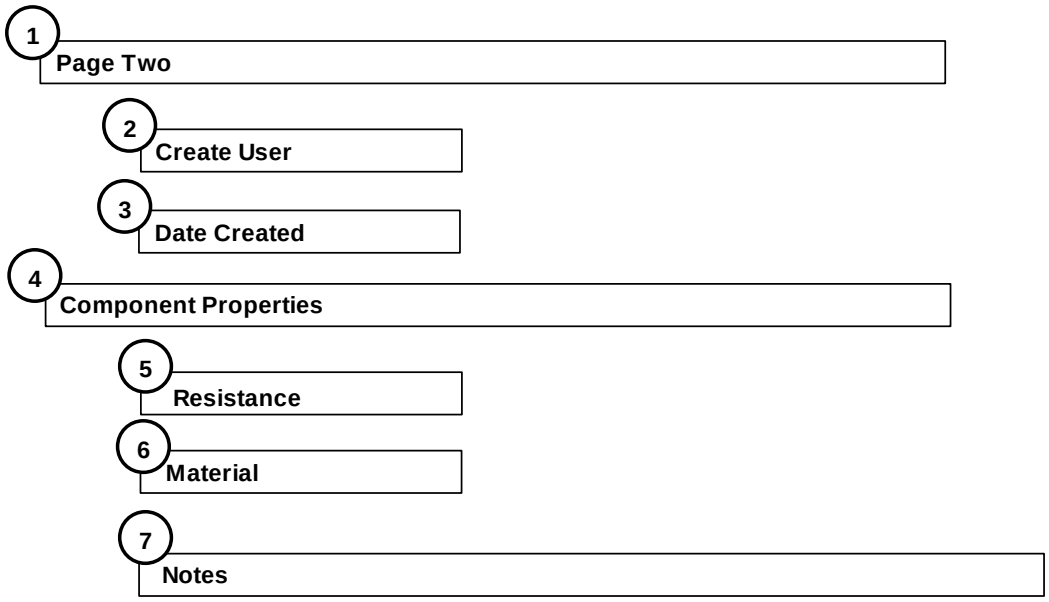
To make the additional information on the first tab easier to understand in Web Client, each Page Two and Page Three tab has 10 Heading attributes. Make Heading attributes visible and give them a meaningful name. Then, position them on the attribute order list so they describe the group of attributes that follow them.

For example, a Page Two attribute order list might be set as follows:

| Attribute order list | Type of attribute |
|---|---|
| Page Two | Heading01 |

| Attribute order list | Type of attribute |
|---|---|
| Create User | CreateUser |
| Date Created | Date01 |
| Component Properties | Heading02 |
| Resistance | Text01 |
| Material | List01 |
| Notes | Multitext10 |

The attributes appear in Web Client, following the Cover Page information, in the format shown in the figure below. In Java Client, the Page Two and Page Three attributes (including Heading attributes) are displayed on the **Page Two** and **Page Three** tabs.




## Removing a Page Two or Page Three Tab

You may effectively remove a **Page Two** or **Page Three** tab from Java Client or Web Client by changing its Visible property to No. You must make this change in the **Tabs** node as well as in each of the Page Two attributes that were defined for the tab. If you do not change an attribute's Visible property to No, it continues to appear in Java Client or Web Client Search folders.

# About Attributes, Flex Fields, and Read-Through Fields

If you have been reading this chapter sequentially, the term "attributes" is already familiar. Before

going into the detailed discussion of Defining Attributes on page 75, this section introduces two special kinds of attributes: user-defined flex fields and read-through attributes.

When users open an object in Agile PLM clients, they see a tabbed window with fields. These fields are called attributes. From the **Classes** node, you define how these attributes appear for each Agile PLM class.

The individual fields that you define for each Agile PLM class appear on the tabs for all subclasses within that class. For example, the part categories that you define for the Parts class become the part categories for its subclasses, such as Capacitors, Resistors, and Diodes. In addition, if you enable an extra tab called a **Page Two** tab for an Agile PLM class, the tab and its attributes also appear for each subclass within the class.

Each subclass also has its own setup window for the **Page Three** tab where you can define the fields that are specific to the subclass, and will appear only on the **Page Three** tab for that subclass.

# User-Defined Flex Fields

You can create an infinite number of attributes in any class or subclass. These are called “user-defined flex fields.” They are distinct from the predefined flex fields installed with Agile PLM.

User-defined flex fields can be created only on **Page Two** and **Page Three** tabs. As such, any user-defined flex field is specific to a class (**Page Two**) or to a subclass (**Page Three**).

**Note** User-defined flex fields are specific to a class and therefore require class-specific privileges. Base class privileges (such as Read Items or Modify Preliminary Items) do not apply to class attributes. The subject of roles and privileges in Agile PLM is broad, but you may want to refer to a small tutorial, Building and Testing Attributes, Read/Modify Privileges, and Roles on page 202.

**To create a user-defined flex field in a class:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. As an example, double-click the **Change Orders** class, then click the **User Interface Tabs** tab.
3. Double-click Page Two, then click the **Attributes: Page Two** tab.
4. Click the **New** icon. The New Attribute dialog appears.
5. Type in a name and API name for the new attribute. Also select an attribute type from the dropdown list. Click **OK**.

   **Note** An attribute name cannot contain a colon ( : ), a semicolon ( ; ), or an equal sign ( = ).

6. The new attribute object appears. This user-defined flex field will be present in all objects created in the **Change Orders** class, for this example.

   You may now fill in the various supporting fields, such as Description, Visible, Default Value, Required, and Available for Subscribe.

**To create a user-defined flex field in a subclass:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. As an example, double-click the **ECO** subclass, then click the **User Interface Tabs** tab.

3. Double-click Page Three, then click the **Attributes: Page Three** tab.
4. Click the **New** icon. The New Attribute dialog appears.
5. Type in a name and API name for the new attribute. Also select an attribute type from the dropdown list. Click **OK**.

   **Note** An attribute name cannot contain a colon ( : ), a semicolon ( ; ), or an equal sign ( = ).

6. The new attribute object appears. This user-defined flex field will be present in all objects created in the **ECO** subclass, for this example.

   You may now fill in the various supporting fields, such as Description, Visible, Default Value, Required, and Available for Subscribe.

## Where User-Defined Flex Fields can be Used

User-defined flex fields are more limited than predefined flex fields. They are supported in the following places in Agile PLM:

## Where user-defined flex fields are fully supported:

- **Advanced Search**
  - User-defined flex fields can be used as a query condition
  - ... can be used in display/output of search results
  - ... can be used as an attribute for sorting of search results
- **Roles and Privileges**
  - User-defined flex fields can be used as an AppliedTo attribute in Read privilege
  - ... can be used as an AppliedTo attribute in Modify privilege
- **Workflow**
  - User-defined flex fields can be used in entry required fields
  - ... can be used in exit required fields
- **Subscription**
  - User-defined flex fields can be made available for subscription when this field changes or is updated
- **Out-of-box Standard Reports**
  - For reports where user can optionally select **Page Two** attributes to display in report results, user-defined flex fields can also be selected.
- **Custom Reports**
  - User-defined flex fields can be used as a query condition in building a custom report
  - ... can be used in display/output of report results
  - ... can be used as an attribute for sorting and grouping of report results
- **Agile Integration Service**
  - User-defined flex fields can be used as part of AIS

- **Agile Content Service**
  - User-defined flex fields can be used as part of CTO.
- **Export**
  - User-defined flex fields can be exported into supported output formats

### Where user-defined flex fields are partially supported:

- **SDK**
  - User-defined flex fields support Set (update) the property and Get (retrieve) the property, but does not support Create.

### Where User-Defined Flex Fields cannot be Used

User-defined flex fields are not supported in the following places within the Agile system:

- **Criteria**
  - User-defined flex fields cannot be used as a value in defining Criteria
- **DataLoad**
  - User-defined flex fields cannot be loaded via DataLoad
- **Notification**
  - User-defined flex fields cannot be used as a data tag that can be sent in subject or body of the notification
- **ACS**
  - User-defined flex fields cannot be used for ATO, since they are autogenerated and modification is not allowed in class settings

## Read-Through Fields

Some fields on some tabs are able to display pertinent information about a related object. These fields are referred to as “read-though” fields. The data that is displayed in a read-through field is a read-only copy of attribute data from a different object that is related to the currently viewed object. If a tab has read-through fields, you can enable or disable them as needed.

Some tabs with read-through fields are not editable (for example, the item object **Changes** tab). However, a few editable tabs include both types of attributes: editable and read-through (for example, the item object **Manufacturers** tab). Read-through fields cannot be edited on the tab because they are read-only copies. However, if the original attribute is modified, the read-only copy in a read-through field will reflect those modifications.

In Administrator, on the **<Tab Name>**:**Attributes** tab of the setup window of object tabs that include both editable fields and read-through fields, read-through fields include the name of the related object in their names. For example, on the item object **Manufacturers** tab, attributes with names that begin with “Mfr” or “Mfr. Part” are read-through fields from the Manufacturer and Manufacturer Parts objects, for example, Mfr. Name and Mfr. Part Notes.

## Read-Through Fields Made Consistent in Items and Changes

Oracle Agile PLM makes it also possible for the administrator to consistently define **Page One** and **Page Two** fields as read-through fields, as follows:

- All Item **P1** / **P2** fields can be defined as read-through fields on the **Item > BOM** table (excluding **Page Two** heading fields)
- All Item **P1** / **P2** fields can be defined as read-through fields on the **Item > Where Used** table (excluding **Page Two** heading fields)
- Most Item **P1** / **P2** fields can be defined as read-through fields on the associated **Change > Affected Items** table (excluding **Page Two** heading fields and these special handling of Rev-related fields: REV.COMPLIANCY_CALC_DATE [only on ECO and MCO], REV.COMPLIANCY [already on ECO and MCO], and REV.INCORP_DATE [ECO only for the future revision, MCO for the revision it applies to]).
- This does *not* include any new (user-generated) **P2** multilist fields.

## Read-Through Fields on the BOM and Manufacturers Tabs

The Item **BOM** tab and the Item **Manufacturers** tab include both editable fields and read-through fields.

Editable fields on an item's **BOM** tab describe the relationship between it and the items on its BOM, for example, Quantity, Find Number, Reference Designators. Read-through fields on the **BOM** tab allow you to display additional information about the items listed on the **BOM** tab, for example, Page Two data of the BOM items.

Editable fields on an item's **Manufacturers** tab define the relationship between it and its manufacturer parts, for example, Preferred Status and Reference Notes. Read-through fields on the **Manufacturers** tab allow you to display additional information about the Manufacturer (for example, the manufacturer's address), or the Manufacturer Part (for example, Page Two data of the Manufacturer Parts).

**Important** Like all read-through fields, the read-through fields on the **BOM** tab and the **Manufacturers** tab cannot be edited on those tables. Because these read-through cannot be edited on the **BOM** tab and **Manufacturers** tab, they also cannot be redlined from the **Redline BOM** tab or the **Redline Manufacturers** tab.

For your reference, the item object **BOM** tab and item object **Manufacturers** tab attributes which can be edited and, therefore, redlined are listed in the table below.

**Note** A user's assigned roles and privileges also determine which specific fields he can edit or redline.

| Item object BOM table fields which can be edited or redlined | Item object Manufacturers table fields which can be edited or redlined |
|---|---|
| BOM Date01 – BOM Date05<br>BOM Description<br>BOM List01 – BOM List05<br>BOM MultiText30, BOM MultiText31<br>BOM Notes<br>BOM Numeric01 – BOM Numeric05<br>BOM Text01 – BOM Text05<br>Find Num<br>Item Number<br>Qty<br>Ref Des | AML Split (%)<br>Mfr. Name<br>Mfr. Part Number<br>Mfr. Tab Date01 – Mfr. Tab Date05<br>Mfr. Tab List01 – Mfr. Tab List05<br>Mfr. Tab MultiList01 – Mfr. Tab MultiList03<br>Mfr. Tab MultiText30, Mfr. Tab MultiText31<br>Mfr. Tab Numeric01 – Mfr. Tab Numeric05<br>Mfr. Tab Text01 – Mfr. Tab Text05<br>Preferred Status<br>Reference Notes |

# Defining Attributes

Attributes are the fields that appear on Agile PLM client tabs. You can define these attributes by configuring their properties.

Each object tab has its own setup window. To display the tab setup window you want, use the guidelines in the following table.

| Type of attribute or property | Setup activity | Do this |
|---|---|---|
| Class-specific attribute<br><br>(all tabs except **Page Three**) | Define attributes for:<br>**Cover Page**<br>**Title Block**<br>**General Information**<br>**Page Two**<br>**Affected Items**<br>**Workflow**<br>**Relationships**<br>**Attachments**<br>**History**<br>**Changes**<br>**BOM**<br>**Manufacturers**<br>**Where Used**<br>**Sites**<br>**Escalations**<br>**Selected Objects**<br>**Where Sent** | In the **Classes** window:<br>▫ Double-click a class name.<br>▫ Click **User Interface Tabs.**<br>▫ Double-click the tab you want to configure. |

| Type of attribute or property | Setup activity | Do this |
|---|---|---|
| Subclass properties | Define properties, for example, for Name, Description, Number Source, AutoNumber Required, Site-specific BOM, and so forth | In the **Classes** window:<br>▫ Double-click a subclass name.<br>▫ Set properties on **General Information** tab. |
| Subclass-specific attribute<br><br>(**Page Three** tab only) | Define attributes for: **Page Three** | In the **Classes** window:<br>▫ Double-click a subclass name.<br>▫ Choose **User Interface Tabs > Page Three** tab. |

**For example, to define attributes for the** Affected Items **tab of the Change Orders class:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Double-click **Change Orders Class**.
3. Click the **User Interface Tabs** tab.
4. Double-click **Affected Items**. The Affected Items tabbed setup window appears.

# Recognizing Attribute Types

An attribute's type indicates how it is used in Agile PLM. For example, a Text attribute is used for entering text into a field on a Java Client or Web Client tab. The types of attributes used in Agile PLM are listed in the table below.

| Type | What to Enter in the Field |
|---|---|
| **Date** | Enter date and time, or select date and time from a pop-up calendar. |
| **Heading** | Displays a heading above a group of attributes. Allows the user to identify a group of associated attributes on these tabs. Heading attributes are available on **Page Two** and **Page Three**. They are also available on the General Information and Preferences tabs of User classes. |
| **Icon** | Read-only attribute type. Used to display icons in tables (for example, BOM table, search results table) that indicate, for example, the object type, that the object has attachments, that the object has a manufacturer, or that the object has a pending change. |
| **List** | Select one item from a list.<br>Select one item from a cascading list. |
| **Money** | Enter a monetary amount. |
| **MultiList** | Select one or more items from a multiple-selection list. The list can be a cascading list.<br><br>**Note**: There is no limit to the number of list entries in a multi-list field and the number of possible multi-list fields on Items **P2** is 15. |
| **MultiText** | Enter a block of text (up to 4,000 characters) that wraps across multiple lines. |
| **Numeric** | Enter a numeric value. |

| Type | What to Enter in the Field |
|---|---|
| **Text** | Enter a single line of text; the allowable length varies, but generally it is up to 50 characters. |
| The Agile PLM administrator can set default values (such as the current date, or current user) to automatically appear in a field when new objects are created. See Setting Default Values on page 83. | |

# Defining Attribute Properties

The following table lists the attribute properties, the settings available for these properties, and how the settings are applied in Agile PLM. It also indicates the attribute types each property can be applied to.

**Note** The "Required" property (found near the end of this table) is important to understand, as it provides a visual cue for users to attend to that attribute when creating a business object in that class or subclass. Also, there is a corner case involving required fields detailed in An Exception involving Required Fields in Object Modification on page 81.

**Note** Please note that a field that is "required to fulfill creation" of an object, this idea is not the same thing as "required fields" that are found in Agile Workflows.

| Property | Attribute type | Setting | Application in Agile PLM |
|---|---|---|---|
| **Name** | All | Supplied by administrator | The name of the field as you want it to be displayed for users, limited to 40 characters.<br>Although all characters are supported within Agile PLM, and should not create issues within Agile's clients, it is recommended that you avoid using the following characters when renaming or creating attributes, due to issues that could potentially arise in integrations or custom applications written on the SDK.<br>, comma . period (dot) : colon ; semicolon = equal sign |
| **API Name** | All | System-generated at object creation | Unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. |
| ***Type** | All | N/A | *Not configurable by the Agile PLM administrator.* Indicates the type of attribute: date, list, multilist, text, multitext, numeric, image, or heading. |
| **Visible** | All | Yes | Makes the attribute visible in Agile PLM clients. Users will see this attribute when they create objects in this class or subclass. |

| Property | Attribute type | Setting | Application in Agile PLM |
|---|---|---|---|
| | | No | Keeps the attribute hidden in Agile PLM clients. Users will not see this attribute when they create objects in this class or subclass.<br>Note that all Item **Title Block** fields can be disabled. |
| **List** | List<br>MultiList | Supplied by administrator | Selections to appear in the dropdown list for users to choose from. The default value must appear among the selections entered here.<br>For list attributes whose selections are defined elsewhere, this property reads **(List)** and the property is not editable on the attribute setup window. For example, the Lifecycle Phase attribute selections are those that are defined on the **Lifecycle Phases** node.<br>List entries are limited to 255 characters each. |
| **Default Value** | All | Supplied by administrator | The default value to display when a new object is created. For lists and multilists, the default value must be one of the values in the selection list (see Setting Default Values for Lists on page 99). For some fields, you can use a default variable.<br>For MultiText attributes, the administrator is constrained to 510 characters of default text, despite the **Max (System) Length** value of 4,000. Web and Java client users are not constrained in this field in business objects. |
| **Enable for Search Criteria** | All | Generally "No" but can be set to Yes by administrator | Permits (Yes) or prevents (No) the attribute from being added as a criteria for a Parametric search. (Parametric searches are documented in *Getting Started with Agile PLM*.) |
| **MaxLength** | Text<br>MultiText | Supplied by administrator | Sets the maximum number of bytes that can be entered in the field (spaces = 1 byte; carriage returns = 2 bytes). Valid settings range from 0 to the maximum allowed for that attribute (see the attribute's **Max (System) Length** property).<br>Generally speaking, 1 character = 1 byte; however, some Agile installations are localized with Chinese or Japanese characters that can use 2 or 3 bytes per character. This must be factored in when setting the value of this property in relation to **Max (System) Length** value.<br>Because many attributes appear in many areas of Agile PLM, and the **MaxLength** can be set independently for specific attributes, issues can develop if there is too much range of **MaxLength**s for similar attributes. For example, a part number may appear on the Title Block of the part, the BOM of (same or other) parts, and the |

| Property | Attribute type | Setting | Application in Agile PLM |
|---|---|---|---|
| | | | Affected Items of changes. It is best to set the **MaxLength**s of such similar attributes to a standard value.<br>When setting **MaxLength**s, consider field length limitations of other applications used by your company to which Agile PLM data is exported. |
| ***Max (System) Length** | Text<br>MultiText | N/A | *Not configurable by the Agile PLM administrator.* Designates the maximum length of the data field in bytes, and represents the maximum number of bytes allowed in the field by Agile PLM clients. An attribute's **Max (System) Length** value cannot be exceeded by its **MaxLength** value. |
| **Include Characters** | Text<br>Multitext | Select from list | This setting determines whether spaces are allowed in the field and which types of alphabetic and numeric characters are allowed. Select from the list of character sets defined on your Agile PLM system. For more information, see Character Sets on page 89. |
| **Min Value** | Numeric | Supplied by administrator | The minimum value for the numeric field. |
| **Max Value** | Numeric | Supplied by administrator | The maximum value for the numeric field; must be greater than or equal to zero (0). |
| ***Order** | All | N/A | This number indicates the order in which the attribute field appears on the table or form. It is an Agile system-generated number greater than 0. The Agile system generates this number when you use the **Order visible attributes** button to arrange the attributes. |

| Property | Attribute type | Setting | Application in Agile PLM |
|---|---|---|---|
| **Site-Specific Field** | | Select from list | Common = this attribute inherits its value from the global value.<br>Site-Specific = the value of this attribute is independent from the global value.<br>This property does not apply to all attributes. If it does not apply, it is set to N/A and it is not editable.<br>This property is not editable for all attributes where it does apply. In those cases, the property has a default value of either Common or Site-Specific and it cannot be edited.<br>For examples, view the attribute setup for the **Affected Items** tab of the Change Orders class. |
| **Height** | MultiText | Supplied by administrator | Determines the number of lines in the text area that are displayed when in edit mode. (In Web Client, the text area displays 4 lines.) |
| **Scale** | Numeric<br>Money | Supplied by administrator | The number of digits after the decimal point in a numeric field. This setting must be greater than or equal to zero (0). |
| ***Attribute** | All | N/A | *Not configurable by the Agile PLM administrator.*<br>Lists the database attribute name for the selected node. Since the Name property gives the attribute name as it will appear to the Agile PLM user, the Attribute and Name properties are often different. |
| ***Base ID** | All | N/A | *Not configurable by the Agile administrator.* Lists the original system ID for the attribute before it was modified in any way. |
| **Input Width** | MultiText<br>List (Web Client)<br>MultiList (Web Client) | | Controls the display width of attributes. Settings are Medium and Long.<br>**Note** For List attributes there is an additional setting of Small available. |

| Property | Attribute type | Setting | Application in Agile PLM |
|---|---|---|---|
| **Input Height** | MultiText | | Controls the display height of MultiText attributes. Settings are numeric integers ranging from 3 to 15. |
| **Required** | All | Yes | The field appears to the user in **bold font**, which is a visual cue that the field should be populated. If an attribute that has Required set to Yes is not populated during the Create New process (in Web Client), or it is emptied during an Edit and the user tries to Save (in both clients), the user will be prompted to fill the field, and the procedure will not complete.<br>You can set Heading attributes to be Required. (There was an issue in **Classes > User Interface Tabs > Page Two > Attributes:Page Two > Heading01** attribute that the field Required was missing.)<br>See An Exception involving Required Fields in Object Modification on page 81. |
| | | No | The field does not appear to the user in **bold font**, and it does not stop the Create New or Edit/Save process if it is not populated. |
| **Available for Subscribe** | All | Yes/No | Determines whether or not this attribute can be subscribed to by a user (who has sufficient privileges). Subscription is documented in *Getting Started with Agile PLM*, Chapter 5, Working with Objects. |
| **The Agile PLM administrator cannot change these properties*. | | | |
| The file type of an attached file is determined by the characters following the last period in a file name. Therefore, users should not use filenames ending with a period (.). | | | |

## An Exception involving Required Fields in Object Modification

If you modify a business object and you select a different subclass as part of the modification, you receive this warning: "**You are changing the subclass of the object. Do you want to continue using the same object name or number?**" This warning does not warn you of another risk, that the object's original subclass and the subclass to which the object is being assigned may have differing required fields. A field that is set to be required in the "new" subclass may not be required in the "original" subclass, but the system does not check for "new" required fields, nor does it warn you of this possible

discrepancy.

### Image Attributes

To include an image attribute on a table, set its Visible property to Yes. The image attribute's Name property identifies which type of icon the image attribute displays. The name does not appear to the end user; you do not need to modify it.

To determine in which table column the image will appear, see Modifying Attribute Order on a Form Tab on page 65.

# Configuring Attributes

A property may be configurable for one attribute and not for another. For example, you cannot edit the Name property of the lifecycle phase attribute on the **Title Block** tab, but you can edit the Name property of the item lifecycle phase on the **BOM** tab.

As the Agile PLM administrator, you can easily change a configurable property. For example, if you decide the maximum allowable length for the change order number should be 15 characters instead of 30 characters, you can change the MaxLength property setting on the Change Orders **Cover Page** tab.

**Note** Because many attributes appear in many areas of Agile PLM, and the MaxLength can be set independently for specific attributes, issues can develop if there is too much range of MaxLengths for similar attributes. For example, a part number may appear on the Title Block of the part, the BOM of (same or other) parts, and the Affected Items of changes. It is best to set the MaxLengths of such similar attributes to a standard value.

## Modifying Attributes

**To modify an attribute property:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. To modify any class tab attribute, double-click the class name.
   a. For example, to modify the **Affected Items** tab attributes for change orders, double-click Change Orders, then select the **User Interface Tabs** tab. All the Change Orders class tabs are listed.
   b. Click **Affected Items**. The Classes Tab setup window appears.
   c. Select the **Attributes:Affected Items** tab. All the fields for that tab are listed.
3. To modify a **Page Three** tab attribute for a specific subclass, double-click the name of the subclass listed under the class name.
   a. For example, double-click ECO under Change Orders class, then select the **User Interface Tabs** tab. All the ECO subclass class tabs are listed.
   b. Double-click **Page Three**. The Class Tabs setup window appears.
   c. Select the **Attributes:Page Three** tab. All the fields for that tab are listed.
4. Double-click the attributes you want to modify.
5. Modify the attribute properties. Depending on the type of property, use one of the following

methods:

- Enter text.
- Select a value from a list.
- Click to open a dialog box.
- For date attribute default values, click to select a specific date, or use a variable. See Default Value Variables on page 15.

6. When you have finished your modifications, click **Save**.

# Matching the Properties of Attributes Common to Several Tabs

When the same attribute appears on several tabs, you must establish the same properties for it everywhere it appears.

For example, an item's Number attribute appears on the item's **Title Block** (that is, its "**Page One**" tab), the item's **BOM** tab (Item Number attribute), and the **Affected Items** tab of any associated changes (Item Number attribute). You must establish identical properties for this attribute in all three locations.

For example, in an item's **Title Block.Number** field, if you set its Include Characters property to All, you should also set the item's **BOM.Item Number.Include Characters** property to All, and set [associated change].**Affected Items.Item Number .Include Characters** property to All. If you do not, when you try to add an affected item, for example, Agile PLM clients prevent you from adding the affected item, and return an error message.

Setting proper attributes allows **Page Two** and **Page Three** data to be copied properly when using **SaveAs** to create a new object (for instance, a project number field on **Page Two** of both ECOs and MCOs), and allows you to display information from an item's **Page Two** on its **BOM** table or an associated change's **Affected Items** tab (for instance, a Unit of Measure field).

Regarding the **SaveAs** method of object creation: if a subclass has Required fields set up (by the administrator) and the user does a **SaveAs** but there is a type violation in the source object on one of the required fields, **SaveAs** will complete with the value still copied over to the source. Previously in this case, the user would encounter an error during **SaveAs** saying that a violation occurred on certain attributes, and **SaveAs** cannot be completed.

# Setting Default Values

You can set default values to appear for attributes.

**To change a default value setting:**

1. Locate the attribute you want, and double-click it to open the setup window.

   (For detailed instructions, see Modifying Attributes on page 82.)

2. In the **DefaultValue** field, enter the default value you want. Depending on the type of attribute, enter a variable (see Default Value Variables on page 15), enter text, select from a drop-down list, or select a date.
3. When you have finished your modifications, click **Save**.

The new default value appears in the Agile PLM client field when a new object is created.

## Showing or Hiding an Attribute on a Tab

Many attributes can be either hidden or displayed on a tab. For example, you can either show or hide the fields on the **Pending Changes** tab, and the attributes on the **Page Two** tab are hidden until you need to use them.

**To show or hide an attribute on a tab:**

1. Locate the attribute you want and double-click it.

   (For detailed instructions, see Modifying Attributes on page 82.)

2. In the **Visible** field for the attribute, select Yes to make the attribute visible. To hide the attribute, select No.
3. When you have finished your modifications, click **Save**.

   The attribute is now either visible or hidden on the tab. To view the modifications, open the object in Java Client or Web Client where the attribute appears.

## Editing a Field Label or Table Column Name

You can change the field label or the heading of a table column by editing the attribute's Name property.

For example, attributes on the **Title Block** tab of Parts class objects are displayed as labeled fields. You can modify the labels that appear by editing the attributes' Name properties. Fields on the **Affected Items** tab of Change Orders class objects are displayed in a table; the table column heading, or field, is the attribute name. You can modify the table column heading by editing the attribute's Name property.

| | |
|---|---|
| **Note** | The Lifecycle Phase attribute name on the Parts and Documents class **Title Block** tabs cannot be edited. |

**To edit a field label or a column name in a table:**

1. Locate the attribute you want and double-click it.

   (For detailed instructions, see Modifying Attributes on page 82.)

2. In the **Name** field, highlight the text you want to edit and enter a new name.
3. When you finish all the attribute modifications, click **Save**.

   To verify that the name was changed, open an object in Java Client or Web Client where the table appears. Click the tab to view the field or table and confirm the modification.

# Using Custom Icons for Subclasses

Each of the predefined subclasses provided with Agile PLM has a default icon associated with it. The subclass icons appear (in Java Client and Web Client) to visually differentiate the various types of objects. You can replace some or all of these icons at any time.

**To replace the default subclass icon with a custom icon:**

1. Open the subclass and, on the **General Information** tab, enter a filename in the **Icon** property for that subclass.

   For Documents and Parts subclasses, there is an additional property called **Icon for Assembly**. This is the icon image that is used when the object represents an assembly.

   For both the **Icon** and **Icon for Assembly** properties, fill in only the graphic filename (including extension), for example, **electrical.gif**.

2. You must place the custom graphic in two locations on the proxy server, one for Web Client and one for Java Client. The system finds the graphic depending on the client being used.

- **OAS:**
  - Stop OAS
  - For **Web Client**, place the custom graphic in the location *\OAS_HOME\j2ee\home\applications\Agile\application\images*
  - For **Java Client**, place the custom graphic in the location *\OAS_HOME\j2ee\home\applications\Agile\JavaClient\lib\images\custom.jar*
  - Restart OAS
- **WebLogic**:
  - Stop weblogic
  - For **Web Client**, place the custom graphic in the location *\AGILE_HOME\agileDomain\applications\application.ear\application.war\images*
  - For **Java Client**, place the custom graphic in the location *\AGILE_HOME\agileDomain\applications\application.ear\JavaClient.war\lib\images\custom.jar*
  - Delete *\AGILE_HOME\agileDomain\servers\xxxx*
  - Restart weblogic

  **Note** For Java Client, custom icons must be added to the custom.jar file. For instructions on how to update a JAR file, refer to Sun's documentation on the JAR utility. An online tutorial on the JAR utility is available at http://java.sun.com/docs/books/tutorial/jar/. You can also use many Zip-compatible utilities, such as WinZip, to update a JAR file.

1. The Agile Application Server must be restarted for the new icons to take effect.

# Quick Access to Objects and Files

You can generate quick access URLs that provide direct pointers to either Agile objects or to Agile attachment files.

These quick access URLs can be pasted into other applications or files such as spreadsheet files, word processing files, a company Intranet web page or WIKI page, or into an email.

For information about quick access to Agile objects using quick access URLs, see:

- Agile-Generated Quick Access Object URLs

- Formats for User-Generated Smart Object URLs
- Using Quick Access Object URLs

For information about quick access to attachment files using attachment quick access URLs, see:

- Generating Attachment Quick Access URLs on page 399

# Agile-Generated Quick Access Object URLs

You can generate Agile PLM object quick access URLs from the Web Client **Actions** menu.

**To generate an Agile PLM object quick access URL:**

1. Login to Agile PLM Web Client and open the desired object.
2. In the **Actions** menu, choose **Copy URL to Clipboard**.
3. The URL is copied to your clip board. You can then paste it into another application or file, such as a word processing file, a spreadsheet file, or an email.

# Formats for User-Generated Smart Object URLs

The Agile Smart Object URL feature is designed to enable effective collaboration through email, spreadsheets, documents, WIKIs, and so forth. Users or automated tools can generate smart readable URLs that point to a specific object in Agile. By using the following format, a user does not need to log into Agile to generate the object URL. Automated tools can also use this format to generate object URLs.

Smart Object URLs can be used for objects available from Web Client.

**Smart Object URL Format**

URLs for pointing to Agile objects use the following format:

http://server:port/VirtualPath/object/ObjectType/ObjectName

The following table defines the Smart Object URL parameters:

| Parameter | Description |
|---|---|
| http://server:port/<VirtualPath>/object/<ObjectType>/<ObjectName> | |
| server<br>port<br>VirtualPath | Server, port, and VirtualPath identify the Agile PLM instance. |
| **object** | The text "object" indicates to Agile that the URL is an object URL. |
| ObjectType | ObjectType refers to the subclass (type) of object, for example, ECO, Part, Manufacturer, File Folder, Program, Project, Activity, or Discussion. |
| ObjectName | ObjectName refers to the unique identifier for the object. Depending on the |

| Parameter | Description |
|---|---|
| | object class, the unique identifier is either an object number or a name.<br>**Note** Manufacturer Parts are a special class of objects that are uniquely identified by two parameters: a Manufacturer Name and a Manufacturer Part Number. Therefore, a URL referring to a specific Manufacturer Part uses the following format:<br>http://server:port/VirtualPath/object/ObjectType/MfrName/MfrPartNumber |

Agile PLM identifies and handles special tags (such as %20) added to URLs to handle special characters (such as spaces and other special characters such as asterisk "*") in an object's unique identifier. For example, the URL

http://server:port/VirtualPath/object/Manufacturer/Philips%20Semiconductor

resolves to the manufacturer "Philips Semiconductor".

# Using Quick Access Object URLs

When a user clicks a Quick Access Object URL, the user actions are the same whether the URL is an Agile-generated quick access URL or a user-generated Smart Object URL:

1. The user clicks the URL and the Agile PLM login screen is displayed in a browser window.
2. The user enters the login information.
3. Upon successful login, Agile PLM searches for the object specified in the link.

   (If the object is not found, Agile PLM displays an error and returns to the user's Home page.)

When the object is found, Agile PLM verifies that the user has the appropriate privileges to read the object and displays the object's first tab, for example, the **Title Page**.

# Chapter 6

# Character Sets

**This chapter includes the following:**


- Available Character Sets ..... 89
- Creating a Character Set ..... 90
- Modifying a Character Set ..... 91


This chapter provides information about how to customize and configure character sets.

## Available Character Sets

The **Character Sets** node lets you view and create sets of valid characters in Agile PLM.

Character sets are lists of valid characters that are used to manage the Include Characters property of Text and MultiText attributes. You can create and delete character sets. Any language supported by Agile PLM can be supported by character sets.

The following table lists character sets provided with Agile PLM. They are all enabled by default and cannot be modified.

| Name | Description |
|---|---|
| **All** | All valid characters |
| **All Characters with space and Lower Case** | Only lowercase characters, but including the space character |
| **All Characters with space and Mixed Case** | All upper- and lowercase characters, and including the space character |
| **All Characters with space and Upper Case** | Only uppercase characters, but including the space character |
| **All Characters without space and Lower Case** | Only lowercase characters and no space character |
| **All Characters without space and Mixed Case** | All upper- and lowercase characters and no space character |
| **All Characters without space and Upper Case** | Only uppercase characters and no space character |
| **Alpha All** | Upper- and lowercase alphabet only |
| **Alpha All & Space** | Same as Alpha All plus space character |
| **Alpha Lower Case** | Lowercase alphabet only |
| **Alpha Lower Case & Space** | Same as Alpha Lower Case plus space character |
| **Alpha Upper Case** | Uppercase alphabet only |

| Name | Description |
|---|---|
| **Alpha Upper Case & Space** | Same as Alpha Upper Case plus space character |
| **AlphaNumeric All** | Upper- and lowercase alphabet and numbers |
| **AlphaNumeric All& Space** | Same as AlphaNumeric All plus space character |
| **AlphaNumeric Lower Case** | Lowercase alphabet and numbers |
| **AlphaNumeric Lower Case & Space** | Same as AlphaNumeric Lower Case plus space character |
| **AlphaNumeric Upper Case** | Uppercase alphabet and numbers |
| **AlphaNumeric Upper Case & Space** | Same as AlphaNumeric Upper Case plus space character |
| **Numeric** | Numbers only |
| **Numeric & Space** | Numbers only plus space character |

# Creating a Character Set

You can create a character set that is tailored to your company's requirements. Each character set is limited to 100 characters.

**Caution** Although all characters are supported within Agile PLM, and should not create issues within Agile's clients, it is recommended that you avoid using the following characters when naming or renaming objects (for instance, subclasses or attributes), due to issues that could potentially arise in integrations or custom applications written on the SDK.

, comma . period (dot)

Previously, Agile experienced issues with certain Unicode characters, specifically katakana characters, as they were not valid XML name characters. Agile PLM has addressed these issues within aXML.

Regarding the "slash" character (/), the SDK makes use of certain characters within its query language, but the recommended practice is to refer to classes and fields by ID, rather than by name. For easy access to folders (e.g., search folders), the SDK allows indication of folders like this: "folder/folder/folder/object" — thus, slashes in folder names could cause issues; however, you can use the escape sequence (with "\").

**To create a new character set:**

1. Under **Data Settings,** double-click **Character Sets**. The Character Sets window appears.
2. Click the **New** button. The Define Character Set dialog box appears.
3. In the **Name**, **API Name** and **Description** fields, type a unique name and a description, respectively.
4. In the **Enabled** list, select Yes.
5. In the **Valid Character Set** field, enter up to 100 characters to include in the set. Duplicate

characters are not allowed.

6. To add special characters, click the **Special Character** button and select space (^s), new line/carriage return (^p), or caret (^^).
7. When you have finished, click **OK**.

**Note** Custom character sets can affect the performance of your system if very large character sets are defined and many attributes use them.

# Modifying a Character Set

You can modify the contents, name, and description of a character set that you have created from the **Character Sets** window. The default character sets provided with Agile PLM are read-only.

**To modify a character set:**

1. Under **Data Settings,** double-click the **Character Sets** node. The Character Sets window appears.
2. Double-click a character set to open it.
3. In the **Name** and **Description** fields, type a unique name and a description, respectively.
4. In the **Valid Character Set** field, add or remove characters.
5. When you have finished, click **Save**.

**To delete a character set:**

1. Under **Data Settings,** double-click the **Character Sets** node. The Character Sets window appears.
2. Select a character set.
3. Click the **Delete** button, and then click **Yes** to confirm.

**Note** If the character set is currently assigned to any Text or MultiText attributes in the Agile PLM system, it cannot be deleted.

Make sure you choose appropriate character sets where required. There is a potential problem when adding files or URLs from a business object to a file folder object. If the file folder's File Description attribute has a Character Set = Numeric, but the business object's File Description attribute has a Character Set = Alpha or AlphaNumeric, the latter will override the former, meaning whatever control you wanted to impose in the file folder's File Description (being numeric) will be lost, the system will save the original character set.

# Chapter 7

# Lists

**This chapter includes the following:**


- Overview ..... 93
- Routing Manager Lists ..... 93
- More Kinds of Lists ..... 95
- Configuring Lists ..... 97
- Creating Cascading Lists ..... 100


This chapter provides information about how to customize and configure lists.

## Overview

The **List** node accesses a collection or library of lists. This is used to manage lists that are used across multiple classes. You can define custom lists that can be used for **Page Two** and **Page Three** list attributes. Many attributes in the Agile PLM system are configured to be lists. Attributes, properties, flex fields on tables, and cells can all be lists.

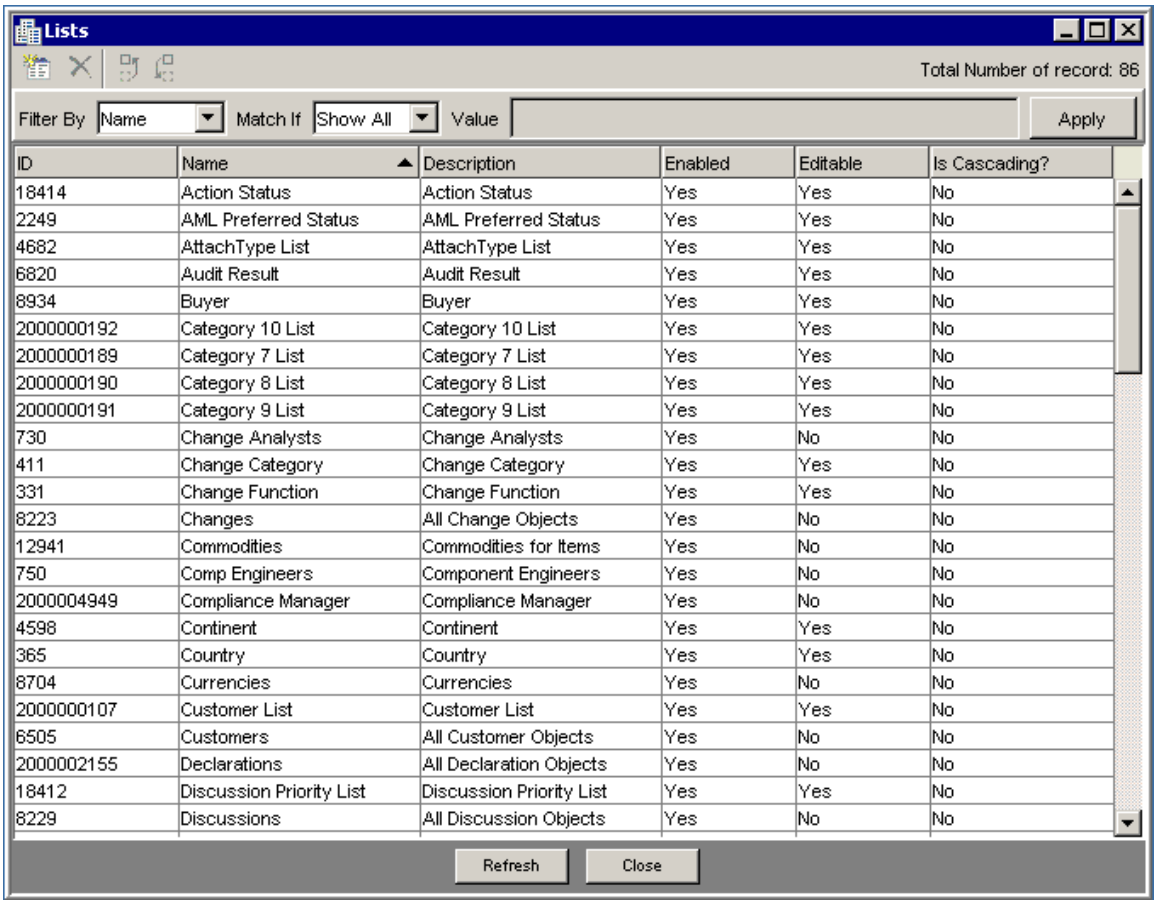
## Routing Manager Lists

There are six lists that pertain to the routing managers of workflows: **Change Analyst** list, **Component Engineer** list, **Price Administrator** list, **Compliance Manager** list, **Quality Analyst** list, and **Quality Administrator** list. They are not editable from the **Lists** node. Names of users are added and removed from these lists according to what you set for a given user's Lists property (**Users >** <user> **> General Info** tab (**Profile** area) > **Lists**).

On the **Lists** node, the single editable property for these six lists is called **Display Type**. This property lets you choose whether the "routing manager" list field in routable objects appears to the originating user as a **List** or as a **Search** format. For companies sized so only a few people are on each list, the dropdown list may be fine. Larger companies may have, for example, one or two dozen change analysts, and they will find the Search format useful; you can select the appropriate change analyst for a change order and all the other analysts will not receive notifications about that change order.

**Note** The default for the **Display Type** field is List.

# Creating Custom Lists and Adding to Lists




| ID | Name | Description | Enabled | Editable | Is Cascading? |
|---|---|---|---|---|---|
| 18414 | Action Status | Action Status | Yes | Yes | No |
| 2249 | AML Preferred Status | AML Preferred Status | Yes | Yes | No |
| 4682 | AttachType List | AttachType List | Yes | Yes | No |
| 6820 | Audit Result | Audit Result | Yes | Yes | No |
| 8934 | Buyer | Buyer | Yes | Yes | No |
| 2000000192 | Category 10 List | Category 10 List | Yes | Yes | No |
| 2000000189 | Category 7 List | Category 7 List | Yes | Yes | No |
| 2000000190 | Category 8 List | Category 8 List | Yes | Yes | No |
| 2000000191 | Category 9 List | Category 9 List | Yes | Yes | No |
| 730 | Change Analysts | Change Analysts | Yes | No | No |
| 411 | Change Category | Change Category | Yes | Yes | No |
| 331 | Change Function | Change Function | Yes | Yes | No |
| 8223 | Changes | All Change Objects | Yes | No | No |
| 12941 | Commodities | Commodities for Items | Yes | No | No |
| 750 | Comp Engineers | Component Engineers | Yes | No | No |
| 2000004949 | Compliance Manager | Compliance Manager | Yes | No | No |
| 4598 | Continent | Continent | Yes | Yes | No |
| 365 | Country | Country | Yes | Yes | No |
| 8704 | Currencies | Currencies | Yes | No | No |
| 2000000107 | Customer List | Customer List | Yes | Yes | No |
| 6505 | Customers | All Customer Objects | Yes | No | No |
| 2000002155 | Declarations | All Declaration Objects | Yes | No | No |
| 18412 | Discussion Priority List | Discussion Priority List | Yes | Yes | No |
| 8229 | Discussions | All Discussion Objects | Yes | No | No |

**To create a custom list for the list library:**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. Click the **New** button. The Create List dialog box appears.
3. Fill in the **Name**, **API Name** and **Description** fields. They are both limited to 255 characters.
4. Select Enabled or Disabled in the **Enabled** list.
5. Select Yes or No in the **Cascade** list.
6. Click **OK**. The new List:<*name*> window appears. The new list also appears in the list library.
7. The Create a new list value window automatically appears. Fill in the **Name**, **API Name** and **Description** fields.
8. Select Yes or No in the **Active** list.
9. Click **Add** or **Add Another**.

**Note** You can create a list without populating it with list items.

**To add values to a list:**

1. Under **Data Settings**, double-click **Lists**. The Lists window appears.
2. If necessary, use the filter bar to display the list you want to populate. Double-click a list to open it.
3. Click the **List** tab. The List:<*name*> List window appears.
4. Click the **New** button. A dialog box appears.
5. Type the item name, type two colons (::), the API name, type two colons (::), and then type a description of the item. The name and description can each be up to 255 characters. The format will be List Value (name)::API Name::List Description.

   | **Note** | Do **not** use commas when adding values to a list. |
   |---|---|

   On separate lines, enter additional list items as needed. You can type names and descriptions of as many list items as you want, separating them with a carriage return. You can also paste list items into the dialog box from a text editor, such as Notepad.

   | **Note** | This dialog box is limited to 4000 bytes. The “carriage return” for each item uses 1 byte. If the average list item uses 20 bytes (including spaces and the carriage return), the list can contain approximately 200 items. Additional list items can be added by selecting **New** again. |
   |---|---|

6. After you finish typing new list items, click **OK**. The list items are added to the list.

   By default, the **Active** property of a list item is set to Yes. If you want to make it inactive, double-click the row and choose No for the **Active** property.

This process of adding multiple list items at a time is called “batch add.”

# More Kinds of Lists

Given that users see Custom and Object lists, let’s further distinguish what kinds of lists users see in their work. You are familiar with the “List” (SingleList) and “MultiList” attributes from Classes on page 49, on page 52.

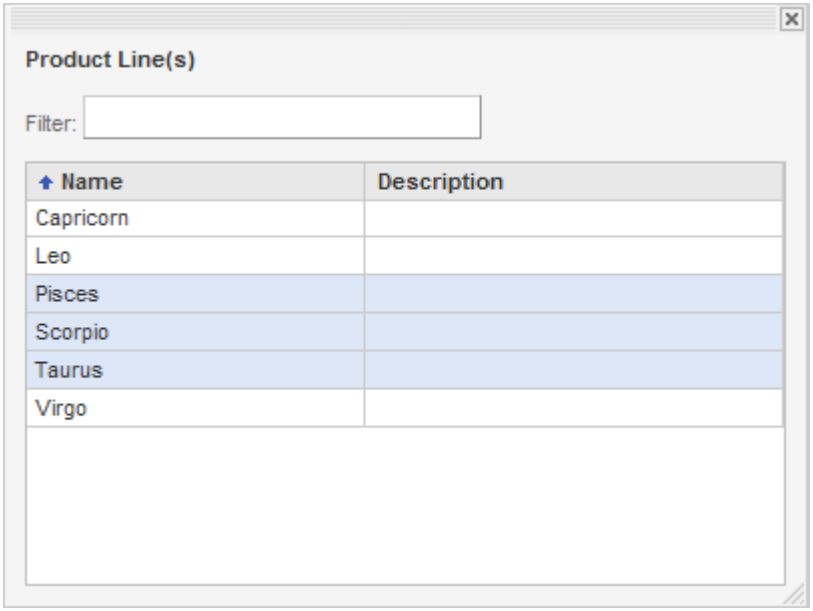
- Simple Lists
- Cascading Lists
- Dynamic Lists

## SingleList Lists

A SingleList attribute or cell presents a list from which only one value can be selected. The following figure shows the **Time Format** field, a SingleList cell in Web Client.

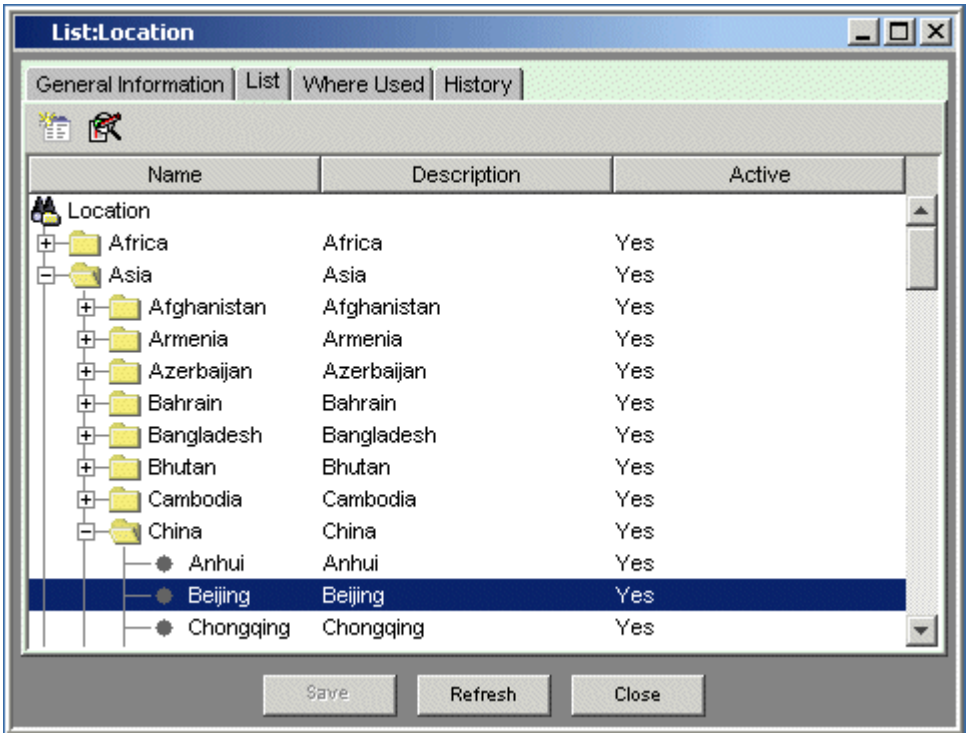
## MultiList Lists

A MultiList attribute or cell is a list from which multiple values can be selected. In Web Client, you can select values for a MultiList cell using the Multiple Value Selection window, shown in the following figure.




## Cascading Lists

A SingleList or MultiList attribute can be configured in Java Client to have multiple hierarchical levels. A list with multiple hierarchical levels is called a cascading list. The following figure shows the setup window (in Java Client) for the Location list, a cascading list. The list has separate levels for continent, country, and city.




**Note** The Location list is the only cascading list that ships with Agile PLM. However, you can define your own cascading lists.

For more information about cascading lists, see Creating Cascading Lists on page 100.

## Dynamic Lists

The Agile PLM server has both static lists and dynamic lists. Static lists are lists that contain an unchanging selection of values. Dynamic lists contain a list of values that are updated at run time. Static lists can be modified; new values can be added and current values can be made obsolete. Dynamic lists cannot be modified; consequently, the Editable property of dynamic lists is set to No.

# Configuring Lists

This section provides instructions for configuring dropdown lists in Java Client. These instructions apply to ordinary lists and cascading lists.

**To modify a list:**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. If necessary, use the filter bar to display the list you want to modify.
3. Double-click a list to open it.

4. On the **General Information** tab, edit the **Name** or **Description** fields.
5. To disable the list, change the **Enabled** field to No.

   **Note** If the list is currently being used in the Agile PLM system, it cannot be disabled.

6. To modify the list values (that is, the list entries), click the **List** tab. Add, delete, or modify values as necessary.

   **Note** If the list is currently being used in the Agile PLM system, its list entry values can be deleted, but there will be a warning message. It is recommended that you deactivate a list entry value so that it no longer appears in Agile PLM clients.

7. Click **Save** to save your changes.

**To delete a list:**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. If necessary, use the filter bar to display the list you want to delete.
3. Select a list.
4. Click the **Delete** button, and then click **Yes** to confirm.

   **Note** If the list is currently being used in the Agile PLM system, it cannot be deleted or disabled.

**To modify a list value (list entry):**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. If necessary, use the filter bar to display the list you want to modify.
3. Double-click a list to open it.
4. Click the **List** tab.
5. Double-click a list value to open it.
6. Fill in the **Name** and **Description** fields. They are both limited to 255 characters.
7. To deactivate the value, set the **Active** field to No.

   **Note** By making a list value inactive, you effectively hide it from Agile PLM clients. However, users can still search for inactive values.

8. Click **OK** to save your changes.

**To delete a list value (list entry):**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. If necessary, use the filter bar to display the list you want to modify.

3. Double-click a list to open it.
4. Click the **List** tab.
5. Select a list value.
6. Click the **Delete** button: you may see the Confirm Dialog with a warning (and the prompt "Do you want to continue to delete?"), click Yes to continue the delete.

   You may see the Delete List Item dialog, which indicates the system's assessment that the list item (list entry) is in use, and that you must first remove all references to the list entry before trying again to delete it.

| **Note** | If the list is currently being used in the Agile PLM system, its values cannot be deleted. However, you can deactivate a list value so that it no longer appears in Agile PLM clients. |
|---|---|

**To change the list used by a List or MultiList attribute:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Open the class or subclass whose attributes you want to modify.
3. Click the **User Interface Tabs** tab.
4. Double-click a tab to open it.
5. Click the **Attributes:<Table>** tab.
6. Double-click the List or MultiList attribute you want to modify.
7. For the **List** property, select a list. You can also click **New List** to create a new list.
8. Click **Save** to save your changes.

| **Note** | If any Agile PLM objects have the attribute set to use the currently selected list, you cannot select a different list. |
|---|---|

# Setting Default Values for Lists

Once you have created the list for a List attribute, you can establish its default value if necessary. For more information, see Setting Default Values on page 83. A default value is one that is selected automatically for a field when you create a new Agile PLM object. The default value for a List attribute must be one of the values in its associated list.

| **Note** | Only the administrator has the ability to set a multi-value default for a multilist attribute. |
|---|---|

## Display Type for Lists in Web Client

The Display Type setting available for Routing Manager Lists is also an available setting for Lists and Multilists for Web Client. In Java Client the administrator chooses whether the object list attribute is displayed as a List or Search format in Web Client. If the selection is List mode then the entries display in a dropdown list. If the selection is Search mode then the entries display in a search palette.

**Note** For list attributes that contain more than 250 entries, the Web Client automatically displays the entries in Search mode.

## Failure Mode Lists

Failure mode is used by product service requests (PSRs) to indicate how the item failed when quality incidents are reported. The failure mode is a list of values that are specific to each item subclass. For instance, if a customer created a subclass of Items called Optics, then a Failure Mode list would be created for Optics that contains the types of defects specific to optical components. Failure mode lists have the following special characteristics:

- When you create a new subclass of the Documents or Parts classes, the list library automatically creates a new Failure Mode list for that subclass.
- If you change the subclass name after it has been created, the name of the related failure mode list is *not* updated. This list will be updated after refresh.
- The **Where Used** tab of failure mode lists is disabled due to the special handling of failure modes in the Agile PLM system.
- A failure mode list cannot be deleted unless its corresponding subclass has been deleted.

Because of their special characteristics, failure mode lists should not be used for other types of list attributes.

# Creating Cascading Lists

Cascading lists have multiple hierarchical levels, presenting several lists within lists.

Configuring cascading lists is very similar to configuring single-level lists. See Configuring Lists on page 97 for more information about modifying lists. Cascading lists have the following additional restrictions:

- Once you create a cascading list, you cannot change it to an ordinary list. After the list is saved, the **Is Cascading?** property is read-only.
- Each folder within a cascading list must contain unique values. However, different folders within a cascading list can share the same values.
- Cascading list values must be added one at a time. There isn't a batch method for adding cascading list values.
- A cascading list can have any number of levels. However, in practice you should try to limit a cascading list to no more than four levels.
- Ability to choose a branch value of a cascading list; both the nodes and the final list values.

**To create a new cascading list:**

1. Under **Data Settings,** double-click **Lists**. The Lists window appears.
2. Click the **New** button. The Create List dialog box appears.
3. Fill in the **Name, API Name** and **Description** fields.

4. To enable the list, make sure the **Enabled** list is set to Yes.
5. In the **Is Cascading?** list, select Yes.
6. Click **OK**. The new List:<*name*> window appears.
7. To add values to the list, click the **List** tab.
8. Click the **New** button. A dialog box appears.
9. Type in the list item name, the API name, and then type a description of the list item.
10. Click **Add** to add the value and close the dialog box. If you wish to save the value and add other values to the same level, click **Add Another**.

    **Note** List values are always added to the level *below* the currently selected value in the cascading list.

11. To add more values, repeat steps 8 through 10.

# Chapter 8

# Process Extensions

**This chapter includes the following:**


- Overview to Process Extensions ..... 103
- Using the Process Extension Library ..... 104
- Assigning Process Extensions to Classes ..... 106
- Assigning Process Extensions to Workflow Statuses ..... 106


This chapter provides information about how to customize and configure process extensions.

## Overview to Process Extensions

The **Process Extensions** node lets you define custom actions that extend the functionality of the Agile PLM system. The custom actions you define can be used to create custom reports, user-driven and workflow-triggered custom actions, and custom tools accessible through Agile PLM clients.

A Process Extension (PX) is either a Java class deployed on the Agile Application Server or a link to a URL. The URL can be a simple Web site or the location of a Web-based application.

There are two types of process extensions available in Agile PLM:

- *Custom process extensions* (Custom PX) is a general name for PXs created in **Data Settings > Process Extensions** node. These PXs have limited initiation capability. This node continues as the site to store and manage your custom PXs.
- *Event-based process extensions* (Event PX) are created in the **System Settings > Event Management > Event Handlers** node. For more information about Event-based process extensions, see Handler Types on page 307 of Event Management on page 301.

There are several integration points for process extensions available in Agile PLM clients. You can invoke process extensions from the following areas:

- Actions menu
- Tools menu
- Workflow Status
- External reports
- Dashboard — URL-based process extensions that are created with Dashboard as the integration point are listed in the Dashboard Extension field, while a Dashboard table with View Type List is created. The output of the URL-based process extension is displayed in the Dashboard table.

**Note** Technically, a custom AutoNumber source is another type of process extension because it uses the same server-side framework as custom actions. For more information about custom AutoNumber sources, see Autonumbers on page 109.

What tools do you need to develop process extensions? For URL-based PXs, you don't need any additional tools. You can define any number of URL-based PXs in the Process Extension Library and specify where those URLs can be accessed from in the client.

**Note** A server property called agile.sso.expirationTime controls the expiration of the cookie used by a URL-based PX. The default value of the property is 86400 (seconds), which means the cookie will expire after one day. The administrator can increase or decrease the expiration time by adding this property to **agile.properties** under **<AgileHome>\agileDomain\config**. Remember the value is in seconds.

To create process extensions that are Java classes deployed on the Agile Application Server, you need to use the Agile SDK, work with Oracle Consulting – Agile Practice, or work with an Agile partner to develop process extensions that suit your company.

For more information about how to use the process extensions framework to develop and deploy custom AutoNumber sources and custom actions, see the *Agile SDK Developer Guide*.

# Using the Process Extension Library

The Process Extension Library is where you define the custom actions that can be used in Agile PLM clients. When you add a custom action to the Process Extension Library, you specify how to initiate that action from the client.

Agile PLM provides two default process extensions, as shown in the following table.

| Name | Description |
|---|---|
| **Accept Package Response** | Sends an Accept response to the Agile PLM system that submitted a Package. |
| **Reject Package Response** | Sends a Reject response to the Agile PLM system that submitted a Package. |

These process extensions are used by Agile Content Service (ACS) to automatically send Accept or Reject responses during Agile-to-Agile communication. For more information, see *Agile Content Service User Guide*.

There are several out-of-the-box process extensions used by Product Governance & Compliance. For more information, see *Product Governance & Compliance User Guide*.

Of course, you can add any number of additional process extensions to the Process Extension Library.

**To add a custom action to the Process Extension Library:**

1. Under **Data Settings**, double-click **Process Extensions**. The Process Extension Library window opens.

2. In the Process Extension Library window, click . The Add Process Extension dialog box appears.
3. Enter the following information:
   - **Name**: Enter the name of the custom action.
   - **API Name**: autogenerated
   - **Description**: Enter a brief description of the custom action.
   - **Type**: Select URL or Internal Custom Action.
   - **Address**: Specify a Web address (if Type is set to URL).
   - **Internal Custom Action**: Select a custom action from the list (if **Type** is set to Internal Custom Action). The list reflects the custom actions that have been deployed on the Agile Application Server using the process extensions framework. If no custom actions have been deployed, the list is empty.
   - **Initiate From**: Select one or more locations from which the custom action can be initiated. Select from the following options:
     - **Actions Menu**: Allows users to select the custom action from the **Actions** menu of a properly configured class.
     - **External Report**: Allows users to generate a report by accessing an external resource or URL. If the process extension is an internal custom action, the External Report option is unavailable.
     - **Tools Menu**: Allows users to select the custom action from the Tools menu.
     - **Workflow State (status)**: Invokes the custom action whenever a properly configured workflow enters a particular status. If the process extension is a URL, the Workflow State option is unavailable.

     If you specify that a custom action is initiated from the **Actions** menu or a workflow status, you can configure subclasses or workflows to use that custom action. If you specify that a URL is used to generate an external report, you can use the Agile Web Client to create the report. If you specify that an action is initiated from the **Tools** menu, it's available at all times in the Agile client.
   - **Roles**: Select one or more roles to use for the custom action. To use the roles and privileges of the current user, leave this field blank. To temporarily override roles and privileges of the current user, select one or more roles. Once the custom action is completed, the client reverts to the current user's roles and privileges.
   - **Enabled**: Select Yes.
4. Click **OK** to save the process extension.

**To modify a process extension:**

1. Under **Data Settings**, double-click **Process Extensions**. The Process Extension Library window opens.
2. Open the process extension you want to modify.
3. Change the settings, and then click **OK**.

**To delete a process extension:**

1. Under **Data Settings**, double-click **Process Extensions**. The Process Extension Library window opens.

2. Select the process extension you want to delete, and then click ✕.

| Note | Custom actions currently being used by a class, workflow, or external report cannot be deleted. If you disable a custom action, it is inactive. Removing a custom action from the Process Extension Library does not affect custom code that has been deployed on the Agile Application Server. |
|---|---|

# Assigning Process Extensions to Classes

To add custom actions to the **Actions** menu of an Agile PLM object (such as a part or an ECO), you configure the object's class. Each base class, class, and subclass has a **Process Extensions** tab. The custom actions that you assign to a class must be previously defined in the Process Extension Library.

Process extensions are inherited from classes and base classes. Consequently, if you assign a process extension to a base class, it is also assigned to classes and subclasses beneath the base class.

| Note | Process extensions can be assigned to only one level in a class hierarchy. For example, if a process extension is assigned to the Part subclass it can't be assigned to the Item base class.<br><br>However, in the case of Agile Portlets, using the process extension Configurator, process extensions can be assigned only to base classes, that is, Items base class, not Parts or Documents classes, nor Part or Document subclasses. |
|---|---|

**To assign process extensions to a class:**

1. Under **Data Settings**, double-click **Classes**. The Classes window opens.
2. Double-click the base class, class, or subclass you want to work with.
3. Click the **Process Extensions** tab.
4. In the toolbar, click . The Assign Process Extension dialog box appears.
5. Select custom actions in the **Choice List**, and then click to move them into the **Selected List**. When you are finished, click **OK**.
6. Click **OK** to save settings.

# Assigning Process Extensions to Workflow Statuses

For each workflow status except the Pending status, you can assign one or more custom actions that are initiated when the workflow enters that status. The custom actions you assign to a workflow status must be previously defined in the Process Extension Library.

| Note | The Automated Transfer Orders (ATOs) class and its subclasses do not support workflow-initiated process extensions. |
|---|---|

**To assign process extensions to a workflow status:**

1. Under **Workflow Settings**, double-click **Workflows**. The Workflows window opens.
2. Double-click the workflow you want to work with.
3. Click the **Status** tab.
4. Select a status other than Pending. The Criteria properties table for the selected status appears below the status table.
5. Double-click the selected status in the Criteria properties table.
6. In the **Process Extensions** list, click . A popup window appears.
7. Select custom actions in the **Choices List**, and then click the **>** button to move them into the **Selected List**. When you are finished, click **OK**.
8. Click **Save** to save settings.

# Chapter 9

# Autonumbers

**This chapter includes the following:**


- How Autonumbers Are Used ..... 109
- About Autonumber Sources ..... 110
- Configuring Autonumber Sources ..... 112
- Creating a New Autonumber Source ..... 114
- Deleting an Autonumber Source ..... 115


This chapter describes how to manage the automatic numbering of Agile PLM objects.

# How Autonumbers Are Used

From the **AutoNumbers** node, you can configure sources for automatic numbering in Agile PLM. For example, you can change the name of an autonumber source or its subclass assignments, or expand the sequence when necessary.

You can also specify your own prefix or suffix for an autonumber source, or create new autonumber sources. You can delete an autonumber source that is no longer needed as long as it has not been used to create an object in Agile PLM clients.

There are two types of autonumbers that can be used; a standard and a custom method. The standard method is to define a sequence by specifying the prefix, suffix, character set, and number of characters allowed. The custom method is more flexible and uses the process extensions framework to define any numbering sequence.

To use process extensions to create custom autonumber sources, you need to either purchase the Agile SDK or work with Oracle Consulting – Agile Practice or an Agile partner to develop process extensions that suit your company's needs. For more information about process extensions, see the *Agile SDK Developer Guide*.

## Modifying an Autonumber Source

You can modify any number source from the **AutoNumbers** node.

**To modify an autonumber source:**

1. Under **Data Settings,** double-click **AutoNumbers**. The AutoNumbers window appears.
2. Double-click a row to open that source's window (for example, the AutoNumber: Deviation Number window).
3. In any editable field, enter a new value or select from the dropdown list.
4. When you have finished, click **Save**.
5. If you wish to revert to the original settings or start over, click **Cancel**. You will be prompted whether you want to cancel the new settings in the window; click **OK** if you do or **Cancel** if you

do not.

The autonumber properties are described in Configuring Autonumber Sources on page 112.

# About Autonumber Sources

An autonumber source is a predefined, consecutive number series used for the automatic numbering of objects created in Agile PLM clients. Each object in Agile PLM is assigned a number when it is created. You can allow users to choose between assigning numbers manually and letting Agile PLM assign numbers automatically, or you can prevent manual number assignment.

The Agile PLM administrator sets up the number assignment convention for the Agile PLM system from the following two administrative nodes:

- From the **Classes** node, you assign specific autonumber sources to specific subclasses. See Changing the AutoNumber Source Property on page 59.
- From the **AutoNumbers** node, you configure the number sources themselves, including their names, whether they have a prefix or suffix, and how many numbers are in each sequence. You can view which subclasses are assigned to a specific number source, and you can configure this arrangement. You can also create new number sources or delete number sources that are not needed.

Autonumber sources are assigned to subclasses so that Agile PLM will know which number source to use when automatically assigning numbers to objects. A separate autonumber source is provided and assigned to most subclasses with your Agile PLM installation.

**Note** Several Agile PLM subclasses, such as Manufacturer and User, do not allow autonumbering.

To view the existing autonumber sources, double-click the **AutoNumbers** node. You can use the autonumber sources provided, or you can make any of the following changes to the configuration:

- Create new autonumber sources using the **New** button.
- Enable or disable an autonumber source using the **Enable** or **Disable** buttons.
- Assign one or more autonumber sources to a subclass.
- Assign one or more subclasses to an autonumber source.
- Add a prefix or suffix to an autonumber source using the **Prefix** or **Suffix** field in the setup window.
- Change the name of an autonumber source using the **Name** field in the setup window.
- When all numbers in an existing autonumber source have been used, expand the number series using the **Next Number** field in the setup window.
- Delete an autonumber source that is not needed using the **Delete** button.

**Note** For information about requiring automatic numbering, see Preventing Manual Number Assignment on page 60.

If you create a new subclass, you can assign it any existing number sources for autonumbering, or create a new number source just for that subclass.

The following table lists the autonumber sources as configured at installation.

| Name | Where Used | Prefix | Number of Characters |
|---|---|---|---|
| ATO Number | ATO | ATO | 7 |
| Audit Number | Audit | Audit | 5 |
| Broker Number | Broker | BROKER | 5 |
| CAPA Number | CAPA | CAPA | 5 |
| Component Manufacturer Number | Component Manufacturer | COMMR | 5 |
| Contract Manufacturer Number | Contract Manufacturer | CTRMR | 5 |
| Contracts Number | Contract | CONTRACT | 1 |
| CTO Number | CTO | CTO | 7 |
| Customer Number | Customer | CUST | 5 |
| Declaration Number | Declaration | MD | 5 |
| Designs Number | Design | DESIGN | 5 |
| Deviation Number | Deviation | DEV- | 6 |
| Discussion AutoNumber | Discussion | D | 5 |
| Distributor Number | Distributor | DISTRIBUTOR | 5 |
| Document Number | Document | DOC | 6 |
| ECO Number | ECO | ECO- | 6 |
| ECR Number | ECR | ECR- | 6 |
| File Folder Number | File Folder | FOLDER | 7 |
| Gate AutoNumber | Gate | G | 5 |
| Historical Report File Folder Number | Historical Report File Folder | HFILE | 5 |
| Manufacturer Rep Number | Manufacturer Representative | MR | 5 |
| Markup Number | Markup | MARKUP | 7 |
| Mfr. Orders Number | MCO | MCO- | 6 |
| NCR Number | NCR | NCR | 5 |
| Package Number | Package | PKG | 5 |
| Part Number | Part | P | 6 |
| PCO Number | PCO | PCO | 5 |

| Name | Where Used | Prefix | Number of Characters |
|---|---|---|---|
| Phase AutoNumber | Phase | PH | 5 |
| PR Number | Problem Report | PR | 5 |
| Program AutoNumber | Program | PGM | 5 |
| Project AutoNumber | | PROJECT | 7 |
| Project Number | Sourcing Project | PRJ | 5 |
| Project Number - Cost Management | | | |
| Project Number - New Products | | | |
| Publish Price Number | Published Price | PRICE | 5 |
| Quote History Number | Quote History | QUOTE | 5 |
| RFQ Number | Request for Quote | RFQ | 5 |
| RFQ Number - Cost Management | | | |
| RFQ Number - New Product | | | |
| Site Change Orders Number | SCO | SCO | 5 |
| StopShip Number | Stop Ship | STOP- | 6 |
| Task AutoNumber | Task | T | 5 |

# Configuring Autonumber Sources

Autonumber sources have the properties listed in the following table.

| Property | Description |
|---|---|
| **Name** | This is the name of the autonumber source. It might simply be the same as the subclass, or it might represent a further distinction, for example, Fasteners or Fasteners–Red. You can change this name. |
| **API Name** | Unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. |
| **Description** | A brief description about the autonumber source. This is an optional field. |
| **Enabled** | Whether this autonumber source is enabled (Yes) or not (No). The default is Yes. |
| **Type** | Standard or Custom. The default is Standard. Custom autonumbering is available to customers with SDK capabilities, which allow you to add a process extension that incorporates existing numbering schema. See Process Extensions on page 103, and *Agile SDK Developer Guide*. |

| Property | Description |
|---|---|
| **Prefix** | Property of a standard autonumber. The characters that precede each number that Agile PLM automatically assigns for this autonumber source, for example, Acme000759 (Acme = prefix). You can change the prefix. |
| **Suffix** | Property of a standard autonumber. The characters that follows each number that Agile PLM automatically assigns for this autonumber source, for example, Acme000759SW5 (SW5 = suffix). You can change this suffix. |
| **Character Set** | Property of a standard autonumber. The characters that are entered in this property are used to create the autonumber. Appropriate for internal autonumber only. Character set can include only a–z, A–Z, 0–9; no special characters are supported. |
| **Number of Characters** | Property of a standard autonumber. Sets the number of characters to use or the length. So if "4" is entered, the numbering can be 0001–9999. Appropriate for internal autonumber only. Limit is 10, so this property can be populated only with 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10. |
| **Starting Number** | Property of a standard autonumber. The first number that Agile PLM automatically assigns in the sequence. This property can be thought of as an arbitrary "next number." |
| **Next Number** | Property of a standard autonumber. While you can manipulate the "Starting Number" value, the Next Number simply reports what the system will use for the next offered autonumber. |
| **Custom Autonumber** | This property of a *custom autonumber* is only activated (and visible) when Type is Custom. A dropdown list field that you populate with your company's customized autonumbering system. The list of custom autonumbers reflects the set of custom autonumbers deployed on the Agile Application Server. |

**Note** For out-of-box subclasses, the defaults of AutoNumber Required = No and Autogenerate = Yes means that a number will be automatically generated when you create an object in that subclass; however, users can enter a new object number.

# Changing an Autonumber's Prefix or Suffix

The Prefix and Suffix properties identify the fixed string of letters or numbers that precede or follow an autonumber. For example, the autonumber prefix for parts is P, so all parts created in Agile PLM clients have numbers preceded by the prefix P, such as P00678. If the autonumber source also had a suffix of 00, number assignments would be followed with 00.

The Agile PLM administrator can change the prefix and suffix properties for an autonumber source. New objects created for subclasses that use this autonumber source have the new prefix or suffix in their autonumbers.

**Note** It is not recommended to change any properties other than **Name**, **Description** and **Enabled** for an autonumber already in use.

# Changing Where an Autonumber Source Is Used

The Where Used property shows you what subclasses use a particular autonumber source in Agile PLM. You can modify this property to permit additional subclasses to use the number source, or to

prevent a subclass from using it.

**To view or change subclasses that use a particular autonumber source:**

1. Under **Data Settings,** double-click **AutoNumbers**. The AutoNumbers window appears.
2. Double-click a row, and that source's window appears (for example, the Deviation Number window).
3. On the **Where Used** tab, click the **Add** button .

   The "Add classes to where used tab" dialog box appears, showing all subclasses that exist in your Agile PLM system. The subclasses listed in the **Selected List** list use this number source for objects created in Agile PLM clients. The subclasses listed in the **Choice List** list do not use this number source.
4. To change the configuration, double-click a subclass name to move it from one list to the other.
5. To save your changes and close the dialog box, click **OK**.
6. When you have finished, click **Save**.
7. If you wish to revert to the original settings or start over, click **Cancel**. You are prompted to confirm whether you want to cancel new settings on the window; click **OK** if you do or **Cancel** if you do not.

## Custom Autonumbers

Select a custom autonumber source from the list. You can create custom autonumber sources only if they have been deployed on the Agile Application Server using the process extensions framework. When you create a custom autonumber source, you select from the **Custom Autonumber** drop-down list. If no custom autonumber sources have been deployed, the list is empty.

## Timeout

Enter the number of seconds after which a timeout error will occur if the custom autonumber source does not respond. The default is 30 seconds.

# Creating a New Autonumber Source

You can create a new autonumber source for user-defined subclasses if required.

## Before You Begin

Consider the following details before creating a new autonumber source.

- Which subclasses will use the new autonumber source?
- What name will you give the new autonumber source?
- Will the new autonumber source require a prefix? If so, what prefix?
- Will the new autonumber source require a suffix? If so, what suffix?
- What size number sequence is required?

**To create a new autonumber source:**

1. Under **Data Settings,** double-click **AutoNumbers**. The AutoNumbers window appears.
2. Click **New**. The Define the AutoNumber dialog box appears.
3. Fill in the **Name**, **API Name** and **Description** fields, and select Yes or No in the **Enabled** list.

   Set the value in the **Enabled** list to Yes if you are ready to introduce this new number source to the Agile PLM system. The default is Yes in this dialog box, but it may be prudent to set it to No until the subclass is entirely set up.
4. Select the type of new autonumber:

   If you selected Standard, fill in, as your company's needs require, the **Prefix**, **Suffix**, **Character Set**, **Number of Characters**, and **Starting Number** fields.

   To prevent duplicating a number source, see About Autonumber Sources for a list of the prefixes for autonumber sources already provided with your Agile PLM installation.

   The Number of Characters setting determines how many autonumbers can be assigned. For example, if you enter 4 in the **Number of Characters** field, the range of numbers available for this series is from 0001 to 9999 (or a total of 9,999 numbers). Choose a value that will prevent the sequence from being exhausted in a short time.
5. If you selected Custom, fill in **Custom AutoNumber** and **Timeout** (seconds) fields.
6. Click the down arrow for the **Where Used** list. A dialog box appears, showing all subclasses that exist in your Agile PLM system. Since you are creating a new number source, the **Selected** field will be empty.
7. If you have already created the subclass that will use this number source, double-click the subclass's name in the **Choices** list to move it to the **Selected** list. If you have not created the subclass that will use this number source, this step cannot be completed; in this case, leave the **Enabled** field set to No.
8. To save your changes and close the dialog box, click **OK**.

   The new name appears as a new object on the AutoNumbers window.

For information about how to prevent Agile PLM users from manually assigning numbers for this subclass, see Preventing Manual Number Assignment on page 60.

# Deleting an Autonumber Source

The Agile PLM administrator can delete an autonumber source that is not needed, as long as it has not been used to create an object in Agile PLM clients.

| **Note** | If an autonumber source has been used to create an object, disabling it makes it unavailable for further use. See Modifying an Autonumber Source on page 109. |
|---|---|

**To delete an autonumber source:**

1. Under **Data Settings,** double-click **AutoNumbers**. The AutoNumbers window appears.
2. Select the autonumber source you want to delete.

3. Click the **Delete** button.

   If the autonumber source has already been used to create an object in Agile PLM clients, you see an error message stating that the object is already in use, and the Agile PLM system does not delete it.

   If the autonumber source has not already been used to create an object in Agile PLM clients, a dialog box appears to confirm the deletion.

4. Click **OK** to confirm the deletion.

   The autonumber source is deleted, and is no longer available for assignment to a user-defined subclass.

# Chapter 10

# Criteria

**This chapter includes the following:**


- About Reusable Criteria ..... 117
- Exploring Reusable Criteria ..... 118
- Creating a New Criteria ..... 119
- Modifying an Existing Reusable Criteria ..... 121
- Using Affected Items Tab Fields in Reusable Criteria ..... 121
- Using $CURRENTREV and $LATESTREV in Criteria for Items ..... 123


This chapter examines the criteria used in Agile PLM workflows, and explains how to create reusable criteria.

# About Reusable Criteria

A reusable criteria is a database query, just like an advanced search. The attributes used to create reusable criteria are similar to the attributes used to create advanced searches. They are also like filters that qualify the objects moving through a workflow process or the conditions for which the privileges masks apply.

The **Criteria** node is where reusable criteria are stored and maintained. From this node, you can create, delete, or modify the reusable criteria that are a fundamental building block of Agile PLM workflows and privileges. Reusable criteria are also used to define designated escalation persons and signoff transfer authority.

Reusable criteria can be assigned to multiple workflows, and to multiple statuses within a single workflow. You can define criteria that:

- Specify or distinguish combinations of **approvers** or **observers**, whether they are selected users, existing or created global groups, or even personal groups.
- Specify each of the company's **product lines**.
- Correspond to **Agile PLM classes.**
- Correspond to any other combination of attributes of **Agile PLM objects**—items, changes, packages, manufacturers and manufacturer parts.

Since changes to the reusable criteria for these workflows are global (that is, a refinement to a reusable criteria affects every workflow in which it appears), you can complete systemwide changes to your workflows in one step.

By specifying a list of reusable criteria for each workflow, you limit which changes can use a specific workflow. For more information about reusable criteria applied as matching criteria, see Using Affected Items Tab Fields in Reusable Criteria on page 121.

# Exploring Reusable Criteria

**To view the available reusable criteria:**

1. Under **Data Settings,** double-click **Criteria**. The Criteria window appears.
2. Filter criteria records to narrow your search. For example, filter records by Description Contains Item to find all the reusable criteria for item objects. (See Filtering Data on page 9.)

   The filtered list of reusable criteria is displayed in the table. The Criteria table shows the name, description, and object type for each reusable criteria. Click a column header to sort the table by that column.

   Use the buttons at the top of the window to perform various reusable criteria management tasks.

| Button | Description |
|---|---|
| **Delete** | Deletes the selected reusable criteria. A reusable criteria cannot be deleted if it is already in use. |
| **Import** | Imports an archive file (.AGI) to create a new criteria. See Administrator Import and Export on page 27. |
| **Export** | Exports reusable criteria data for the selected reusable criteria. See Administrator Import and Export on page 27. |
| **Export All** | Exports all reusable criteria data. See Administrator Import and Export on page 27. |
| **New Criteria** | Create a new reusable criteria. See Creating a New Criteria on page 119. |

# Viewing Reusable Criteria

The reusable criteria can have any of the Agile PLM classes or subclasses as an object type. Many of the out-of-box reusable criteria have been used to define privilege masks.

**To view specific reusable criteria:**

1. Under **Data Settings,** double-click **Criteria**. The Criteria window appears.
2. Filter criteria records to narrow your search. For example, filter records by Description Contains Item to find all the reusable criteria for item objects. (See Filtering Data on page 9.)
3. In the Criteria window, click the name of the reusable criteria you want.

   The tabbed window for that reusable criteria appears.

The **General Information** tab of the Criteria setup window has button you can use to perform the available actions described in the following table:

| Action | Description |
|---|---|
| **Delete** | Deletes the reusable criteria if it is not in use. |
| **Export** | Exports the reusable criteria information. See Administrator Import and Export on page 27. |

| Action | Description |
|---|---|
| **Save As** | Allows you to copy the reusable criteria and give the copy a new name.<br>▫ Click **Save As**.<br>▫ Enter a new name in the dialog box that appears.<br>▫ Click **OK**. |

## Criteria Windows and Tabs

When you double-click an existing reusable criteria, its properties are displayed in a tabbed window in the right pane. The **General Information** tab displays the name and description of the criteria, and object type that criteria applies to. You can use the **Edit** button to edit the name and description.

The **Criteria** tab lists the type—class or subclass—that the criteria is associated with and the criteria parameters. These are the conditions that define how each reusable criteria works.

**Note** You can edit a reusable criteria that's already being used, but its **Name**, **Description**, and **Object Type** fields are read-only.

The **Where Used** tab tells where reusable criteria are used in criteria-specific properties, matching criteria, privilege masks, escalations, and transfers. The **History** tab lists the actions that have been performed in relation to this reusable criteria.

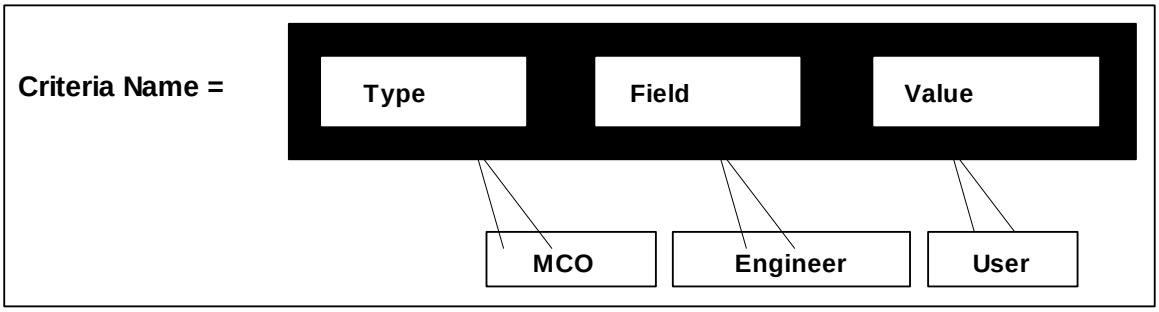
# Creating a New Criteria

You create reusable criteria according to the needs of your workflow or privilege mask. Because reusable criteria can be widely used in Agile PLM, their purpose and function need to be clear from the start.

It is recommended that you devise a system or convention for naming reusable criteria.

**Caution** Although you can change the name of reusable criteria at any time, doing so too often may confuse users, who do *not* see a name change until they have logged out and back into the Agile PLM client.

A possible naming scheme could be:




For example, a criteria name of “MCO_Acme_July2000” is a lot easier for your users to decipher than “MCO6.”

**Note** Reusable criteria created by the Agile PLM administrator appear in the Transfer Authority dialog as Global.*Criteria name*. Personal reusable criteria created by other users appear in the Transfer Authority dialog box as User.*Criteria name*.

**To create a new reusable criteria:**

1. Under **Data Settings,** double-click **Criteria**. The Criteria window appears.
2. Click the **New** button. The Create Criteria dialog box appears.
3. Fill in the **Name**, **API Name** and **Description** fields, and select the object type that your reusable criteria will apply to. This list comprises all the Agile PLM classes and subclasses.

   **Note** The reusable criteria name must be unique. If you specify a reusable criteria name that is already used, you get a duplicate name error message.

4. If you want the criteria to be case-sensitive, select the **Case Sensitive** checkbox. Case-sensitive searches improve system performance and can simplify how you define reusable criteria:
   - If you enter text in the **Value** field, the criteria will look for text that is an exact match.
   - If you define a numerical value, checking **Case Sensitive** allows the Agile PLM system to make use of internal database settings to find objects more quickly.
5. Click **Add** and select the Agile PLM attribute you want from the **Attribute** dropdown list. The values that are available change according to the Agile PLM class or subclass you selected in the previous step. Click **OK** in the small dialog box.
6. Click the **Match if** field and specify the search operator.
7. Click the **Value** field, and select a value for the field you selected from the **Attribute** list. Click **OK** in the small dialog box.
8. If you are going to add additional conditions, click the **Add** button again, and then in the **And/Or** field, select And or Or.
9. You can click the **Insert** button to add a condition above the currently highlighted row.
10. Repeat step 5 through step 9 until your reusable criteria is complete.
11. When the criteria is complete, Click **OK**.

| Note | Changing the criteria in the **Object Type** field resets the entire reusable criteria. A reusable criteria covers one entire Agile PLM class or subclass. You need to create a separate global criteria to cover a different class or subclass. |
|---|---|

Click the **( )** button to place parentheses around the specified condition or conditions, which changes the order in which the search conditions are evaluated. The **( )** button functions like a formula within parentheses in an algebraic equation, following the standard algebraic order of operations. The grouped criteria within parentheses are resolved before any others.

# Modifying an Existing Reusable Criteria

You cannot modify an existing reusable criteria if it is already in use. You can, however, remove the reusable criteria from its assigned objects, modify it, then reassign it again. Or, you can create a new reusable criteria, assign the new criteria to the objects and remove the old criteria from the objects. The **Where Used** tab lists all the objects where the reusable criteria is in use.

**To modify an existing reusable criteria:**

1. Under **Data Settings,** double-click **Criteria**. The Criteria window appears.
2. Filter criteria records to narrow your search. For example, filter records by Description Contains Item to find all the reusable criteria for item objects. (See Filtering Data on page 9.)
3. In the Criteria window, double-click the name of the reusable criteria object you want.
4. The criteria setup window appears containing the existing information about the criteria. Modify the criteria just as when creating it.
5. When you are finished, click **Save**.

| Note | You cannot delete a reusable criteria once it has been used to create a Transfer Authority privilege, even when you delete that transfer authority. You might rename the criteria you want to delete so it is, for instance, dropped to the bottom of the list of reusable criteria. |
|---|---|

# Using Affected Items Tab Fields in Reusable Criteria

There are many similarities between reusable criteria and other searches in the Agile PLM system, but there are a few important differences.

## How Changes Are Matched When Reusable Criteria Specify Affected Items Tab Fields

To select a specific routable object in Agile PLM clients, an affected items field condition must be true for all the objects on the **Affected Items** tab of that routable object. For example, if you specify Affected Items.Old Lifecycle Phase Equals Preliminary, then all the objects on the **Affected Items** tab must have the **Old Lifecycle** field equal to Preliminary. If you select Contains as the search operator, then every object on the **Affected Items** tab must contain the specified value in the specified field.

When you create a reusable criteria in Java Client, you can include criteria conditions against item fields (such as **Part Category**), provided the item fields are displayed on the **Affected Items** tab of the routable object.

**Note** Criteria that specify Affected Items attributes cannot be used to create privilege masks. If a criteria does not appear in a privilege mask's dropdown list, it cannot be used to define the privilege mask. See also Revising Item Descriptions.

In the Create Criteria dialog box, the **Attributes** list contains **Affected Items** tab fields, including those from the Items themselves. This allows you to create reusable criteria that return changes according to fields of the Items that appear on the **Affected Items** tab. For this kind of query to work, the **Affected Items** tab field must be visible.

**For example, to find changes with Items on their** Affected Items **tab that have a** Part Category **field equal to Engineering, do the following:**

1. Under **Data Settings,** double-click **Classes**. The Classes window appears.
2. Click the class you want (for example, Change Orders).
3. In the Class setup window, click the **User Interface Tabs** tab.
4. Double-click **Affected Items**. The Class Tabs setup window appears.
5. Click the **Attributes** tab.
6. Find **Item Category** in the Name column. double-click the row to display the Attributes setup window.
7. In the **Visible** dropdown list, select Yes. The Visible property is now set to Yes.

# Difference Between “Item” and “Part” in Affected Items Names

It is important to understand how the system interprets the naming of affected items when you create reusable criteria, as shown in the following table:

| Name (general) | Applies to... | Name (example) | Applies to... |
|---|---|---|---|
| **Affected Items.Item fieldname** | Parts and documents | **Affected Items.Items.P2 Text20** | **Text20** field for both parts and documents |
| **Affected Items.Part fieldname** | Parts only | **Affected Items.Part.P2 List03** | **List03** field for part |

Most list and multilist fields for Documents class objects cannot be used to create reusable criteria. For instance, the list fields on **Page Two** and **Page Three** tabs are not available. The two available **Documentation** list fields are **Documentation.Product Line** and **Documentation.Size**.

## More about Affected Items: Adding Approvers and Observers

Default change analysts (or default component engineers) may want to monitor changes assigned to them to make sure that the list of approvers and observers is complete. When workflows assign approvers and observers according to the attributes of the affected items of a routable object, you may want to add approvers that were not automatically assigned by the workflow.

For example, at Acme Inc., each product line has several projects in development at one time. Acme uses a field on **Page Two** (in Agile Java Client) to identify which project or projects each item (part or document) belongs to. The Libra Product Line workflow looks at the project assigned to the affected items on an ECO to determine which default approvers and observers to assign.

Mary Green creates ECO 333 in Agile Java Client. She adds six Orion project objects to the **Affected Items** tab: three parts and three documents. Mary finishes preparing ECO 333 and switches it to the next status, where the change is submitted to Bob Smith, the default change analyst.

To select approvers for the Orion project, the Libra Product Line workflow examines each item on the Affected Items table to determine if the **Page Two** field contains Orion. If *any* the items on the Affected Items table belong to the Orion project, then the list of Orion approvers, as defined in the workflow, are automatically added to the **Workflow** tab when ECO 333 is submitted to Bob.

However, when some of the objects on the Affected Items table are documents, Acme requires that Orion team members in the Publications department must also sign off the change. Since not all ECOs include documents to sign off, the Libra Product Line workflow does not automatically add approvers from the Publications department to every ECO.

Bob, the change analyst, examines ECO 333 and notices that some of the affected items are documents. Bob clicks the **Add Approvers** button on the **Workflow** tab of ECO 333 and adds all the members of the Orion Publications group as approvers. Bob can add approvers to any future Review or Released status type; the ECO does not need to be in the Review or Released status type for approvers or observers to be added.

# Using $CURRENTREV and $LATESTREV in Criteria for Items

$CURRENTREV and $LATESTREV are two powerful system variables available to the administrator to construct criteria that can target specific revisions of items (parts and documents) and specific workflow statuses (for those workflows that apply to rev-based changes, that is, Change Orders, Sites Change Orders, and Manufacturer Orders).

Be aware that this functionality is quite granular and will be best understood when you have become familiar with item revisions (in Production Collaboration) and workflows. It may be helpful to request assistance from your Oracle Consulting – Agile Practice representative.

These variables are also used to construct some criteria in Privilege Masks, which is another area of robust capability and granularity. Variables were introduced in Getting Started in Administrator, see Default Value Variables on page 15; $CURRENTREV in conjunction with the Modify privilege is detailed in Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV on page 243.

# About TitleBlock.Rev

Title Block.Rev is currently not available for use in criteria conditions. Instead, use $CURRENTREV criteria conditions to identify which revision the user has selected.

# Rules for Using the $CURRENTREV Variable

The $CURRENTREV variable is used to construct criteria based on the workflow status of the change (Change Order or Manufacturing Change Order) that corresponds to a selected item revision.

The following rules govern the usage of the $CURRENTREV variable:

- $CURRENTREV may only be used in defining criteria against objects belonging to the Items base class (including Parts and Documents classes and their subclasses).
- $CURRENTREV is available as a value in the Attribute list. $CURRENTREV itself does not evaluate to a value: the entire condition itself is evaluated as an expression.
- The **Match If** operator can be set to **Equal To** and **Not Equal To**.
- **Value** can be set to the following values.

  The list of values is comprised of $STATUSTYPE variables for all predefined Change Status Types, as well as all possible Workflow.StatusName values for all configured Workflows created for the Changes object type.

  **Note** The only Change types that are pertinent to this feature are the ones that appear in the Item's Rev dropdown list, that is, those belonging to Change Orders (ECOs), Manufacturer Change Orders (MCOs), and Site Change Orders (SCOs) classes. However, the other Change classes (Change Requests, Deviations, Stop Ships, and PCOs) do appear in this list. **For expressions pertaining to $CURRENTREV, only statuses pertaining to workflows developed for ECOs, MCOs, and SCOs should be used.**

  - The condition **$CURRENTREV Equal To $STATUSTYPE.PENDING** evaluates to True if the selected item revision corresponds to a Change whose current status type is Pending. Similar statements can be constructed for other status types as well. The supported status type variables include:

    $UNASSIGNED

    $STATUSTYPE.PENDING

    $STATUSTYPE.SUBMIT

    $STATUSTYPE.REVIEW

    $STATUSTYPE.RELEASED

    $STATUSTYPE.COMPLETE

    $STATUSTYPE.HOLD

    Note that **$STATUSTYPE.CANCEL** is not available, since revisions pertaining to Canceled changes do not appear in the Rev dropdown list for items, and therefore may not be selected. In any case, the expression **$CURRENTREV Equal To $STATUSTYPE.CANCEL** would always evaluate to False.

- The condition **$CURRENTREV Equal To Default Change Orders.Pending** evaluates to True if the selected item revision corresponds to a change that is assigned the Default Change Orders workflow, and whose current status is Pending. Similar statements can be constructed for other workflows and statuses as well. The general format is:

  <WorkflowName>.<StatusName>

  Note that <WorkflowName>.CANCEL is not available.

- While the workflow status values above cover all possible statuses for a pending or released revision, special criteria values are provided for the Introductory revision because there are no changes associated with that revision.

  Based on the possible use cases, one of the following conditions would hold true when an Introductory revision of an item is selected (the term Pending change below implies a change in its pre-Released state, for example, Submitted, Review):

  - **$INTRODUCTORY_NOCHANGE** – There are no Pending or Released changes against the item (the **Changes** tab is empty). This condition also covers the case where any past Pending changes against the item have been soft or hard deleted from the Agile system.
  - **$INTRODUCTORY_PENDINGCHANGE** – The item has never been released, but there are Pending changes against the item. This condition also covers the case where the item has been released in the past, but the released changes have all been unreleased
  - **$INTRODUCTORY_RELEASEDCHANGE** – The item has one or more released changes against it. It may also have one or more Pending changes against it.

  An expression involving any of these three values will always evaluate to False if the selected revision is not the Introductory revision.

- To support the use case where users are allowed to read, make changes to or get file attachments from only the latest released revision of an item (this is important in regulated industries (for example, medical devices) where it is important that users refer to only the latest design or spec files), another variable called $LATEST is introduced to the set of valid values for $CURRENTREV expressions. The expression **$CURRENTREV Equal To $LATEST** evaluates to True if the selected revision is the Latest Released revision for the item. This expression also evaluates to True if the selected revision is Introductory and the item does not have any Released ECOs, MCOs or SCOs against it.

  The expression **$CURRENTREV Equal To $LATEST** has the exact same functionality as the obsolete expression **Title Block.Rev Equal To $LATEST**.

The behavior of item Description on the Latest Released revision deserves special mention here. When a user has the privilege to modify the Latest Released revision only (**Title Block.Rev** = $LATEST) and not Pending revisions, and he modifies the description of the item, he is presented with a message asking if he'd like to update the description of all the pending revisions to this updated description. If the user chooses to do so, all the pending revision descriptions are also updated irrespective of whether or not the user has privilege to update descriptions on pending revisions.

# Rules for Using the $LATESTREV Variable

The $LATESTREV variable is another Administrator attribute used to construct criteria based on the workflow status of the Change Order (ECO/MCO/SCO) corresponding to the latest released revision of an item.

The following rules govern the usage of the $LATESTREV variable:

- $ LATESTREV may only be used in defining criteria against objects belonging to the Items base class, which includes Parts and Documents classes and their subclasses.
- $ LATESTREV is available as a value in the Attribute list. $LATESTREV itself does not evaluate to a value. The whole condition itself is evaluated as a boolean expression.
- Unlike $CURRENTREV, you must specify only the **Match If** field to construct a valid expression using $LATESTREV. The only operators available for **Match If** are **Is Released** and **Is Introductory**.
- The condition **$LATESTREV Is Released** evaluates to True if the selected item has been released on at least one Change of type ECO or MCO. Note that SCO is not included because an item has to be released on an ECO or MCO before it can be released on an SCO.
- The condition **$LATESTREV Is Introductory** evaluates to True if the selected item does not have any released ECO or MCO against it.

# Item Relationship Creation Using $CURRENTREV and $LATESTREV Criteria

To create a relationship and a relationship rule between any two Agile objects requires that the user has Modify privileges for both objects applied to <object>.Relationships.Name attribute (to create a relationship) and <object>.Relationships.Rule (to create a relationship rule). See Some 'Modify' Basics and Rules on page 242.

Modify Item privilege masks that use a $CURRENTREV criteria condition to limit item modification also limit which items can be selected for item-to-item relationships. For example, if a user has a Modify privilege that limits his Modify Item privilege mask with the criteria condition **$CURRENTREV Equal To $STATUSTYPE.PENDING**, that user can modify only items with a change in the Pending status. The items that can be selected for item-to-item relationships are also limited to items with a change in the Pending status.

To avoid this limitation to item-to-item relationships when using $CURRENTREV criteria conditions in Modify Item privilege masks, use the following criteria:

```
$CURRENTREV Equal to $LATEST
```

To create a Modify privilege mask with only one Applied To attribute, Item.Relationship.Name, use the following criteria:

```
Modify Items $CURRENTREV Equal to $LATEST Applied To
Item.Relationships.Name
```

To ensure that users can create item-to-item relationships between any two items, include this Modify privilege mask in any role that includes Modify Item privilege masks.

**Note** Relationships are not revision-specific.

# Using File Folder Attributes in Reusable Criteria

For file folder objects privilege checking is based on the latest checked in version of the file folder object. Privilege checking is not performed on checked out versions of file folders.

For example, if you create a Discover privilege mask using a criteria that requires the file folder lifecycle phase attribute to equal Pilot, then a checked out version of a file folder with the Pilot lifecycle phase will not be discovered. However, once the user checks in the file folder, then the checked in file folder will be discovered because the lifecycle phase of Pilot now meets the criteria.

# Chapter 11

# Workflows

**This chapter includes the following:**


- How Workflows Automate the Change Control Process ..... 129
- Workflow Basics ..... 130
- Default Status-Based Workflows ..... 133
- Status Types Defined ..... 133
- Creating New Workflows with Save As ..... 135
- General Workflow Properties ..... 136
- Creating and Using Custom Workflows ..... 139
- Status Properties Defined ..... 145
- Criteria-Specific Properties Defined ..... 151
- Workflow Functionality Testing ..... 156
- Migrating Workflows to Production ..... 157
- Changing Active Workflows ..... 158


The Workflows node stores the default workflows that Agile PLM client users select to create and send routable objects for approval. From this node, you can also define and create custom workflows.

# How Workflows Automate the Change Control Process

The **Workflows** node stores the default workflows that Agile PLM client users select to create and send routable objects for approval. From this node, you can also define and create custom workflows

A workflow is an organized sequence of stages involving people's decisions—their creation, modification, review, and approval or rejection of a routable object. An Agile PLM workflow is an automated sequence of statuses that a routable object follows as it goes through a company's change control process. (For a detailed description of a typical change control process, see the chapter about workflows in *Getting Started with Agile PLM*.)

Workflows automate the change control process in a variety of ways:

- The changes that are available for a particular routable object are completely configurable. When users create a routable object, they select a workflow from the list of workflows that are available for that routable object.
- When all required fields are filled in for a released routable object, and when all the approvers have signed off for a Review or Released status type, the routable object can be moved automatically—"autopromoted"—to the next status.
- Approval and rejection of the routable object occurs within defined permissions and other checks. You can preassign default approvers and observers.
- Email automatically notifies users of the creation and progress of a routable object. For

example, when an approver does not respond within the specified time period, a reminder email notification is triggered to be sent.

- If the approver does not respond within the specified period, the routable object is sent to that user's designated escalation person, who can approve or reject the routable object.
- By defining a workflow with multiple Review and Released statuses, routable objects can be sequentially routed to different lists of approvers and observers.
- A user can transfer authority to approve routable objects to another user for a specified period of time. With the appropriate privileges, a user can similarly transfer signoff authority for other users.

## Modifying a Workflow

The following sequence applies to modifying a workflow in the **Workflows** node.

**To modify the Workflow settings:**

1. Under **Workflow Settings,** double-click **Workflows**. The Workflows window appears.
2. Double-click the workflow you want to modify, for example, Default Change Orders. The Workflow: Default Change Orders window appears. The setup tabs are:
   - **General Information**
   - **Status**: the Status table defines status properties, and the Criteria table defines criteria-specific properties.
   - **History**
3. With the **General Information** setup tab selected, in the field you want to edit, enter a new value or select from the dropdown list. The two **Criteria Matching Type** fields on the **General Information** tab should be left alone at this point.
4. When you have finished, click **Save**.

To modify this workflow's status properties or criteria-specific properties, begin by clicking the **Status** tab.

For information about changing workflows that are active (that is, are enabled and may have routable objects moving through them), see Changing Active Workflows.

**Note** Alterations to workflow statuses have implications for process extensions (see Process Extensions on page 103) and transfer orders (see *Agile Content Service User Guide*).

# Workflow Basics

## Routing Managers

Any object that can be routed for approval is a **routable object**. The user who oversees the routing and approval process is the *routing manager*. The various routing managers (see the table below) are simply roles assigned to users by the administrator. Once a user has been assigned a role of, say, change analyst, his name appears on the Change Analyst List and can be selected manually by a Java Client or Web Client user. Routing managers evaluate and assign routable objects, and

they receive email notifications pertaining to the objects to which they are assigned.

The following table lists routable objects and the corresponding default routing manager.

| Routable object (with base class) | Default routing manager |
|---|---|
| Change: All changes except MCOs & PCOs: change orders (ECOs), change requests (ECRs), site change orders (SCOs), stop ships, deviations | Change analyst |
| Change: Manufacturer orders (MCOs) | Component engineer |
| Change: Price change orders (PCOs) | Price administrator |
| Declarations (all) | Compliance manager |
| Packages | Program manager |
| Product service requests (PSRs):<br>problem reports and nonconformance reports (NCRs) | Quality analyst |
| Project: activities and gates | PPM program manager |
| Quality change request (QCRs):<br>corrective and preventive actions (CAPAs) and audits | Quality administrator |
| Transfer Order:<br>content transfer orders (CTOs) and automated transfer orders (ATOs) | Content manager |

# Other Workflow Basics

Here is some other basic information about Agile PLM workflows:

- A workflow consists of a unique name, one or more *matching criteria*, and a *status list*, or sequence of statuses. An enabled workflow is visible and usable by Agile PLM client users.
- Each workflow must be qualified by at least one matching criteria; these are broad qualifiers that determine which workflow a routable object may follow. Agile PLM tests the routable object against all the matching criteria for all the workflows and determines which workflows match the routable object.
- A workflow can contain any number of statuses. *Status names* are editable, but their underlying *status types* are not. Only one status type, Pending, is required and this must be the first status in the workflow. Three status types—Submit, Review, and Released—may be used multiple times in a workflow, with unique names for each instance. Review and Released are the most versatile status types.
- Each and every status in a workflow's status list comprises *status properties* and *criteria-specific properties*.
- Status properties define some specific actions that can be automatically triggered or manually performed—or both. Status properties determine what happens to the routable object when assigned people approve, reject, or even forget to review the routable object.
- Criteria-specific properties are defined by filters called *reusable criteria*, which are created separately and stored in the **Criteria** node. Each individual status may have many criteria-specific properties. Criteria-specific properties describe specific actions that are triggered

(automatically, or they can be manually performed) if the routable object meets the conditions defined in the reusable criteria while the routable object is entering or exiting a status. Criteria-specific properties apply to individual statuses, not to the workflow as a whole.

- For example, criteria-specific properties can be used to define the following actions for a specific Review status of a workflow. When a routable object enters the status, if the routable object's Product Line field contains, for example, Libra, a specific set of approvers is assigned. If the Product Line field contains Scorpio, a different set of approvers is assigned. If the Product Line field contains both Libra and Scorpio, both sets of approvers are assigned.

## Keep It Simple!

Agile PLM workflows are a robust feature that offers a great deal of choice and flexibility for your company's change control process. For best results, plan thoroughly and gather all necessary data and information.

To prepare properly, follow these three steps before you attempt to customize Agile PLM workflows:

1. Compile information for tailored workflows. You must fully understand and define your company's required change control processes. This step is crucial to successful implementation of automated workflows.

   **Important** This process is not covered in this manual. Consulting services are available from Oracle Consulting – Agile Practice consultants, who can provide you with assistance in creating business process validation plans and guidance. During business process validation, you can propose workflows to simulate real scenarios and typical usage.

2. Learn about Agile PLM workflows in this chapter and elsewhere in this manual.
   - Be sure you understand the purposes and limitations of the status types. See Status Types Defined on page 133.
   - Use default workflows at first, which you can use immediately without modification. See Default Status-Based Workflows on page 133.
   - Create some original workflows (in your test environment—see the next step) that meet specific needs at your company. See Creating and Using Custom Workflows on page 139.
3. Establish a testing environment and complete test procedure. Test all new workflows! For example, create a routable object and send it through your enabled workflow. In any case, verify that new workflows work correctly, and alter them as necessary. For more information, see:
   - Workflow Functionality Testing
   - Migrating Workflows to Production
   - Changing Active Workflows

Although you can create as many workflows as your company needs, you are encouraged to explore the capacities of the out-of-box default workflows first. Creating too many workflows too quickly could create maintenance issues later.

# Default Status-Based Workflows

Agile PLM provides the following status-based workflows:

- Default Activities
- Default ATOs
- Default Audits
- Default CAPAs
- Default Change Orders
- Default Change Requests
- Default CTOs
- Default Declarations
- Default Deviations
- Default File Folders
- Default Gates
- Default Manufacturer Orders
- Default Non-Conformance Reports
- Default Packages
- Default Price Change Orders
- Default Problem Reports
- Default Site Change Orders
- Default Stop Ships

**Note** Default ATOs and Default File Folders are read-only workflows. They cannot be modified.

The default workflows can be used without modification. They ensure that all routable objects match criteria for a workflow. You can also tailor the default workflows to your requirements. If modifying the default workflows still does not meet your requirements, you can create custom workflows.

**Caution** As a best practices approach, it is recommended that you save a set of the default workflows, unmodified and disabled. You can do this easily using **Save As** (see Creating New Workflows with Save As on page 135).

# Status Types Defined

Agile PLM workflows are defined by their list of statuses, and each status is of a particular status type. Workflows are created from five basic status types—Pending, Submit, Review, Released, and

Complete. Two special status types, Cancel and Hold, handle routable objects that have failed to advance and must be stored.

The status types have sequence dependencies that determine where and how often you can use them in a workflow. A clear understanding of these dependencies is essential to constructing usable workflows. The following table defines the status types and provides details about these dependencies.

A workflow can have as many statuses as needed. Submit, Review, and Released are the status types that can be used more than once, with differently named and defined statuses. When you define a workflow with multiple Review and Released statuses, routable objects can be sequentially routed to different lists of approvers and observers.

| Status type | Definition |
|---|---|
| **Before a workflow is selected for a routable object** | |
| **Unassigned** | **Unassigned** is the default initial status name before a routable object has been matched with a workflow. |
| **Basic Status Types (Active)** | |
| **Pending** | ▫ An originator is developing the routable object.<br>▫ A workflow must begin with a **Pending** status type; it is the first status in every workflow. |
| **Submit** | The **Submit** status type sends the routable object to the specified routing manager, who determines whether or not to promote the routable object to the next status. |
| **Review** | The **Review** status type routes the routable object to CCB members (approvers and observers). When all approvers have signed off, and required fields have been completed, the routable object is automatically promoted to next status in the status list if AutoPromote is on, or it is manually moved by the routing manager if AutoPromote is off. |
| **Released** | ▫ When a routable object enters the *first* **Released** type status, the software performs a series of checks; it also creates new revisions of the affected items (ECOs) with the new BOM, incorporates redlines into a new "rev," or updates the manufacturer part information (MCOs).<br>▫ Like the Review status type, the Released status type may have approvers and observers and be routed for signoff and may be autopromoted.<br>▫ *Only the* first *Released status actually releases the routable object*. Subsequent Released status types—given distinguishing status names—provide additional review/approval cycles for the released routable object.<br>▫ Returning a released routable object to any of the first three status types *un*releases the routable object—removes approvals that were gathered. A Deviation can be unreleased if it does not have affected items.<br>**Note:** Although Agile PLM allows you to configure Change Order workflows that allow the release of an item without specifying a value for the New Lifecycle Phase, configuration management best-practices require that a New Lifecycle Phase always be specified prior to releasing an item on an ECO or MCO. In order to make sure there is a New Lifecycle Phase, make sure it is a required field in the Workflow Criteria. |

| Status type | Definition |
|---|---|
| **Complete** | ▫ If **Complete** status type is used, it must be the last *active* status in the workflow, and it may be used only once. |
| **Special Status Types (Inactive)** | |
| **Hold** | ▫ The **Hold** status type provides a way to temporarily remove a routable object from the workflow and reinsert it later. It is added to the *end* of the status list as it is *never* part of the active sequence of statuses.<br>▫ Hold does not unrelease a routable object.<br>▫ Given appropriate privileges, the user can manually move a routable object to Hold from any other status in the workflow *except* Complete (assuming it is listed as a valid next status under the Manual Valid Next Status property). It stays there until you manually move it back into the workflow. These moves require proper Change Status privileges. |
| **Cancel** | ▫ The **Cancel** status provides a way to remove a routable object from the workflow. It is not possible to re-activate the workflow once it is moved to Cancel status.<br>▫ Cancel unreleases a routable object.<br>▫ Given appropriate privileges, the user can manually move a routable object to Cancel from any other status in the workflow *except* Complete (assuming it is listed as a valid next status under the Manual Valid Next Status property).<br>**Note** In assigning privileges to move a workflow to the Cancel status, the administrator must recognize the potential for user-error. Because a Canceled workflow can **not** be re-activated by any means, the fact that it is relatively easy to change a workflow to Cancel must be recognized and allowed for. |

You can use the default workflows without modification. It is best to work with the default workflows with no modifications until you are familiar with their uses and capabilities.

**Caution** Agile recommends using **Save As** to create "archive" copies of all the default workflows. This ensures that you can always return to an unaltered version if you have customized the default workflows.

# Creating New Workflows with Save As

You can create a new workflow from an existing one and give it a new name. This makes it easy to create an archive of the default workflows or to create a custom workflow that starts from the foundation of a proven default workflow.

**To create a new (or "archive") workflow from an existing one:**

1. Under **Workflow Settings,** double-click **Workflows**. The Workflows window appears.
2. Double-click the workflow you want to base the new workflow on. That specific workflow's window appears.
3. Click the **Save As** button, and enter a descriptive, unique name. Autogenerate the API name

If you are creating an archive copy, use a name that will distinguish it as “off limits,” for example, “EXAMPLE Default Chg Orders.”

4. Click **OK**. The setup window displays the new name in the title bar.
5. Change the information as necessary, and click **Save**.

   The new workflow is disabled (the Enabled property is No), and can remain so for archived workflows, or will be enabled when you have finished constructing a customized workflow.

   **Note** Workflows created with **Save As** are disabled. A common mistake after creating a custom workflow is forgetting to enable it.

   Each Change Status privilege mask applies to one status of one specific workflow. You must create Change Status privilege masks for each status of every new workflow you create and enable, including workflows created by Save As. See Change Status on page 217.

# General Workflow Properties

Double-clicking the **Workflows** node opens the Workflows window, which lists the available default workflows and any custom workflows you have created.

The window has the following buttons: **New**, **Delete, Edit Workflow Init Settings**, **Import**, **Export**, **Export All**, **Enable**, and **Disable.**

The **Edit Workflow Init Settings** button opens a dialog box with the following fields:

| Field | Type | Default |
|---|---|---|
| **Default Initial Status Name** | Text | Unassigned |
| **Status Stamp Color** | List (of colors) | Blue |

## General Information Tab

When you double-click a default workflow in the Workflows window, the **General Information** tab of the setup window opens. The tab has the **Delete**, **SaveAs**, and **Export** buttons. It has the following fields:

| Field | Description |
|---|---|
| **Name** | The name of the workflow. Each workflow name must be unique |
| **API Name** | Unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. |
| **Enabled** | Enables or disables the workflow. An enabled workflow is visible and usable by Agile PLM client users. |
| **Object Type** | The type of routable object for which the workflow applies. Available for the following: changes, CTOs, QCRs, PSRs, projects, packages. Object types are listed in Agile PLM Class Structure on page 50.<br>Object Type is read-only after the workflow is created. |

| Field | Description |
|---|---|
| **Matching Criteria** | Determines which routable objects can follow the workflow. Can use any reusable criteria, for example, All Change Orders. |
| **Workflow Criteria Matching Type** | Refines how the matching criteria determines which changes can follow the workflow. It defines what happens when a routable object meets one or more reusable criteria that contain information about affected items.<br>Choose Same, All, or Some. See Workflow Criteria Matching Type on page 137. |
| **Status Criteria Matching Type** | Determines what happens when criteria-specific properties that contain information about affected items apply to a routable object.<br>Choose Same, All, or Some. See Status Criteria Matching Type on page 138. |

# About Workflow name

The workflow's name and whether or not it is enabled are defined at this level. The default workflows are already enabled. For information about what takes place when a workflow is enabled, see Status Properties Defined on page 145. For information about modifying or fixing problems in active workflows, see Changing Active Workflows.

When a workflow is created by the administrator, the **Object Type** field is limited to base class choices. The **Matching Criteria** property in a workflow selects an object type, and that is what the system picks up when a workflow is selected as the object type of an Event mask. For more information regarding Events and how they relate to Workflows, see Event Management on page 301.

# Workflow Criteria Matching Type

For each workflow, by specifying a list of reusable criteria in the Workflow Criteria Matching Type property, you limit which changes can use that workflow. Some examples of possible reusable criteria applied as matching criteria are:

- **All MECOs** — Finds all the changes that are MECOs (mechanical ECOs).
- **Scorpio ECOs** — Finds all the ECOs that include "Scorpio" in the **Product Line(s)** field of the routable object.
- **Libra Project** — Finds all the changes that have any items on the **Affected Items** tab that contain "Libra" in the **Product Line(s)** field of the item.

For example, you might create a workflow named "General Use," and select the three reusable criteria named above (All MECOs, Scorpio ECOs, and Libra Project) as the matching criteria for that workflow.

**Example 1:** If an Agile PLM client user creates a change that is a mechanical ECO (MECO), it successfully matches one of the matching criteria of the General Use workflow (All MECOs). The General Use workflow will appear in the **Workflow** dropdown list on the **Cover Page** tab of the change (Java Client) or the Cover Page page of the change (Web Client).

**Example 2:** If the user creates a change that is an ECO that includes “Scorpio” in the **Product Line(s)** field of the ECO, it successfully matches one of the matching criteria of the General Use workflow (Scorpio ECOs). The General Use workflow appears in the **Workflow** dropdown list on the **Cover Page** tab of the change (Java Client) or the Cover Page of the ECO (Web Client).

**Example 3:** If the user creates a change order and adds items to its **Affected Items** tab that contain “Libra” in the **Product Line(s)** field of the item, Agile PLM examines the **Product Lines** field on the **Affected Items** tab for every affected item. The General Use workflow will appear in the **Workflow** dropdown list on the **Cover Page** tab of the change (Java Client) or the Cover Page of the Change Order (Web Client) depending on the setting of the Criteria Matching Type property for the workflow:

- **Same** — All affected items must match the *same* affected item–based reusable criteria, in this case, Libra Project.
- **All** — When multiple affected item–based reusable criteria are used as matching criteria, *each* affected item must match at least one affected item–based reusable criteria; however, each affected item does not have to match the *same* reusable criteria.
- **Some** — One or more (but not all) affected items must match the affected item–based reusable criteria, in this case, Libra Project.

As the user adds items to the **Affected Items** tab and completes the fields on the tabs of the routable object, the **Workflow** dropdown list on the **Cover Page** tab may vary, depending on which matching criteria apply at the moment.

# Status Criteria Matching Type

Because reusable criteria appear on a workflow’s **Status** tab for each discrete status, Status Criteria Matching Type applies to the groups of reusable criteria in each status. The property is applied in Agile PLM clients when a status’s criteria-specific properties contain information about affected items that is pertinent to the routable object, for instance, which approvers or observers should be assigned to it.

The property values and their behavior are listed below, followed by an example.

- **Same** — All affected items must match the *same* affected item–based reusable criteria in a given status. If *all* the affected items don’t match the same reusable criteria, the ECO does not satisfy *any* exit criteria, and the “If No Criteria Apply at Exit, Notify” property is applied.
- **All** — When multiple affected item–based reusable criteria are used as criteria-specific properties in a given status, *each* affected item must match at least one affected item–based reusable criteria; however, each affected item does not have to match the *same* reusable criteria. “All” ensures that every affected item has an appropriate approver.

  For example, for each reusable criteria in a status that one of the affected items matches, the specified approvers are added to the **Workflow** tab of the ECO. If *each* of the affected items does not’t meet *at least* one of the reusable criteria, the ECO would not satisfy *any* exit criteria, and the “If No Criteria Apply at Exit, Notify” property is applied.
- **Some** — One or more (but not all) affected items must match at least one affected item–based reusable criteria in *each* status’s criteria-specific properties. For each reusable criteria in a status that one of the affected items matches, the approvers are added together for the ECO.

The general setting (that is, on a workflow’s **General Information** tab) for Status Criteria Matching Type

is validated against *both* the reusable criteria (the group of criteria in that status's Criteria table on the **Status** tab) for the present status *and* the reusable criteria for the next status in the workflow.

For example, Sarah has written a change called "ECO2" with the following affected items (on the **Affected Items** tab of the change in Java Client):

- Affected Item 1 (AI-1): **Product Line** = Aries, **Category** = Electrical
- Affected Item 2 (AI-2): **Product Line** = Aries, **Category** = Mechanical

For ECO2's Submitted status (Submit status type), on the Criteria table on the **Status** tab, is a reusable criteria that specifies:

- Reusable Criteria W (RC-W)**: Product Line** = Aries, **Affected Item.Category** = Electrical, **Required fields** = change description

Next, for ECO2's CCB status (Review status type), on the Criteria table on the **Status** tab, is a reusable criteria that specifies:

- Reusable Criteria X (RC-X)**: Product Line** = Aries, **Affected Item.Category** = Mechanical, **Approver** = Paul

If Sarah wanted to move the change from Submit to CCB, ECO2's Status Criteria Matching Type (SCMT) is set to All, AI-1 (the Electrical part) now matches RC-W, but AI-2 (the Mechanical part) does not match RC-W. If Sarah put RC-X under Submit as well, AI-2 now matches a criteria for Submit; however, because Criteria Matching Type is set to All, the change still cannot move to CCB because AI-1 does not match RC-X under CCB.

If SCMT is set to Some, Sarah can advance her change in both cases because one of the affected items matches a criteria for each status (a different AI for each status in this case, first Electrical, then Mechanical).

If SCMT is set to Same, even after Sarah adds RC-X to Submit, ECO2 is not able to advance because each affected item does not match each criteria: AI-1 matched RC-W and AI-2 matched RC-X.

# Creating and Using Custom Workflows

You will probably find it is much easier to create a new workflow by saving one of the default workflows with a new name (see Creating New Workflows with Save As on page 135), and then modifying the status properties and criteria-specific properties to suit your purposes.

The value of creating a custom workflow from scratch is that you go through all the elements of a workflow so that you are more familiar with the entire process, including the point where you will have to troubleshoot a workflow that is not doing what it is supposed to.

When creating a custom workflow, you must specify the following things:

- Name of the workflow.
- API name of the workflow.
- Matching criteria — at least one for any workflow.
- Status list — including how many statuses, status names, status types.

- Status properties — defined for each status in the workflow.
- Criteria-specific properties — defined for each status in the workflow.

## Before You Begin

It is important to plan as thoroughly as possible before you create an original workflow. To get the most out of this process, you need to do *all* of the following:

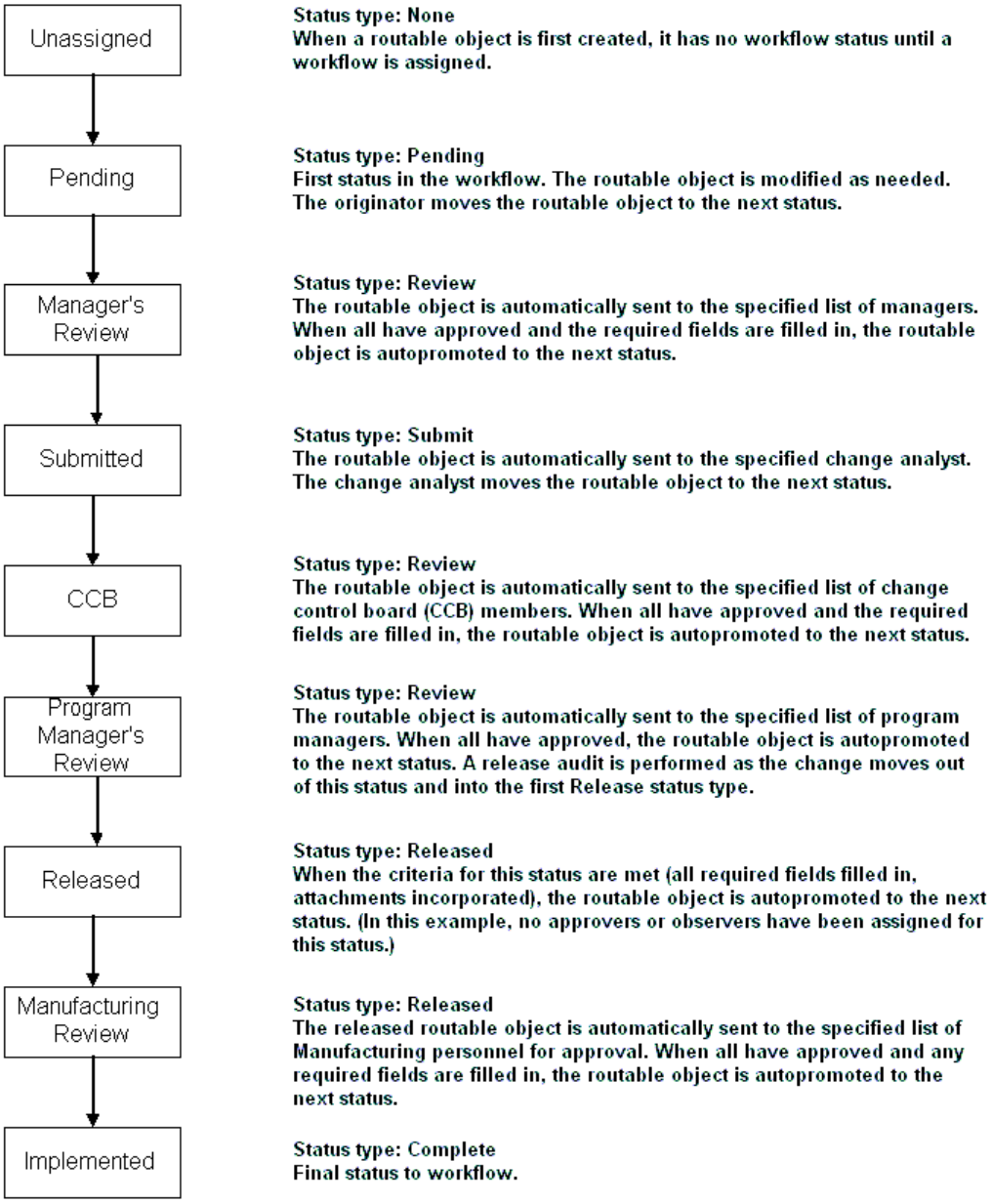
- Read this manual and become familiar with workflows documentation.
- Enable and use one or more of the default Agile PLM workflows in live situations (see Default Status-Based Workflows on page 133).
- Compile specific information and data that define what you need an original workflow to do. This includes a sequence of statuses that differ from those in the default workflows.

## Custom Workflow Example

The next several sections describe how to create the workflow in the figure below. This customized workflow demonstrates several departures from the “one status per status type” structure of the default workflows:

- The Submit status does not immediately follow the Pending.
- There are three Review statuses. The names in the example are less important than the idea that you can have as many Reviews as you need.
- Two statuses use the Released status type—but the second one is named Manufacturing Planning, which acts like another Review. Because the Review and Released status types share many attributes—approvers, autopromotion, and so on—this permits additional Review cycles *without* **un***releasing the routable object*. If a routable object enters a Review *following* a Released status type, it is *un*released, which means:
  - The routable object must qualify again for release (through the release audit: see Release Audit, or Audit Current Status for Conditions to Release).

Any revisions that were assigned to the affected items are revoked.




# Guidelines for Creating Workflows

The guidelines below describe the steps required to build a custom workflow. The steps are explained in detail in the following sections.

**To build a custom workflow:**

1. Define your change control process, including the kinds of routable objects you'll need to process with workflows.

   Contact your Agile representative or Agile Consulting Partner for help in defining your workflow

processes.

2. Create all required reusable criteria from the **Criteria** node first; these are necessary to describe (for the system) the distinguishing characteristics of the workflow.

   See Criteria on page 117.

3. Create a new workflow and give it a unique name, but don't enable it. Enabling is the final step.

   See Creating a New Workflow on page 142

4. Select at least one reusable criteria (defined from the **Criteria** node) for the matching criteria.
5. Create the Status list for the new workflow.
6. Modify the properties of each status on the **Status** tab, understanding that the available status properties are different for each status type.
7. Add criteria to the Criteria table on the **Status** tab and define the criteria-specific properties of the statuses, particularly required fields and approvers and observers.
8. Assign appropriate privileges that allow users to refine statuses in a new workflow. The privileges most pertinent to workflows are Add Approver/Observer, Remove Approver/Observer, Change Status, Override; Transfer Authority for Self, and Transfer Authority for Others.

   See Privileges and Privilege Masks on page 207. To move a routable object through the new workflow, users must have Change Status privileges for that specific workflow; see Change Status on page 217.

9. Set SmartRules for systemwide workflow functionality. See General System Settings.
10. Enable the modified workflow and test it. When it is ready, notify your Agile PLM users that it is available. They will be able to use it the next time they log in to Agile PLM.

# Creating a New Workflow

It will be helpful to create a list like this table with the statuses of your new workflow defined both by name and by status type.

| Status name | Status type |
|---|---|
| Originate | Pending |
| Review-PM | Review |
| Submit-CA | Submit |
| Review-CCB | Review |
| Review-Sr.PM | Review |
| Released | Released |
| Review-Mfg.Planning | Released |
| Implemented | Complete |

**To create a new workflow:**

1. Under **Workflow Settings,** double-click **Workflows**. The Workflows window appears.
2. Click **New**. The Create Workflow dialog box opens.
3. Enter a unique name, such as **Taurus ECOs** and an API name.

   A "best practice" may be to conceptualize a system of naming your original workflows. For some ideas about naming conventions, refer to Creating a New Criteria on page 119 and Naming Privilege Masks on page 223.

   The **Enabled** field on the **General Information** tab is not available at this point. No new workflow can be enabled until it is complete. When you have completed a new workflow, you must enable it to use it. Agile PLM client users will see the newly enabled workflow the next time they log in to Agile PLM.
4. In the **Workflow Criteria Matching Type** and **Status Criteria Matching Type** dropdown lists, allow (or select) Same.

   Matching criteria are used to find which workflows may be used for each routable object. Agile PLM checks all the matching criteria for all the workflows and determines which workflows match the routable object. For example, if a reusable criteria called All Taurus ECOs is a matching criteria for a workflow, then the user can select that workflow for an ECO in the Taurus project.

   The factors you need to consider to take full advantage of these three criteria properties are fairly complex. Detailed information is provided in Using Affected Items Tab Fields in Reusable Criteria on page 121.
5. From the **Object Type** dropdown list, select the appropriate base class.
6. In the **Matching Criteria** dropdown list, scroll down and select All Change Orders.
7. Click **OK**. The new workflow is added to the list on the Workflows window.

To complete setting up the new workflow, you need to double-click it in the Workflows window and create statuses with defined status properties and criteria-specific properties.

**To create statuses for a new workflow:**

1. Open the workflow, click the **Status** tab, and click **New** just above the Status table. The Add Workflow Status dialog box appears.
2. Type an appropriate name; in our example, it is **Originate** and API name.
3. In the **Status Type** dropdown list, select an appropriate status type, in this case Pending (the required first status for any workflow).
4. In the **Status Stamp Color** dropdown list, select a color for the type in the status stamp that will appear for that status in Agile PLM clients.
5. Click **OK**. Repeat this process for all statuses you wish the workflow to have.

   If you scroll to the right in the Statuses table, you see two status properties—"If No Criteria Apply At Exit, Notify" and "Manual Valid Next Status"—that need to be given values.

   Agile recommends creating all the statuses, then filling in the status properties before considering the criteria-specific properties. However, with so many factors to consider, the most important thing is to develop your own method or systematic approach to constructing an error-free workflow.

**To designate reusable criteria for a new workflow:**

1. Open the workflow, click the **Status** tab, select a status (row) in the Status table, then click the **New** button just above the Criteria table. The Add Workflow Criteria dialog box appears.
2. The **Status Name** field is grayed and populated with the name of the status you have selected. Click the down arrow for the **Select Criteria** field, and select one or more reusable criteria for that status.
3. When you have finished, click **Save**.

   Repeat this process for all statuses in the new workflow.
4. When you have completed the construction of your custom workflow, and have verified that it is ready for users to apply routable objects (see Workflow Functionality Testing), you need to enable it. In the setup window for the workflow, select Yes in the **Enabled** dropdown list.

   **Note** A new workflow is disabled as you create it. A common mistake after creating a custom workflow is forgetting to enable it.

   **Note** To move a routable object through the new workflow, users must have Change Status privileges for that specific workflow. See Change Status on page 217.

   Agile PLM client users will see the new workflow when they restart their client application *after* you have enabled the workflow.

# When a New Workflow is Enabled

When a workflow is enabled, the system checks the following items:

- The first status is the status type Pending.
- If there is a status of status type Complete, then it must be the final status of the workflow (excluding special statuses Cancel and Hold).
- The workflow's matching criteria must contain at least one reusable criteria.
- Each status in the workflow must be assigned at least one reusable criteria in the workflow's Criteria table.
- If there are Cancel or Hold statuses, they must appear at the end of the status list. If there is a Hold status, it must be the last status on the list.

If any of these requirements are not completed, an error message appears, and the workflow is not enabled.

# Routing to a User Group

A user can assign a user group (global group, not a personal group) as an approver or observer using valid approvers and observers. The administrator or Agile Web Client user who is assigning a user group can select whether one member or all members must sign off the routable object. See Adding User Groups as Approvers of a Change on page 181.

# Status Properties Defined

This section describes all of the Status Properties found in Agile PLM workflows. When you double-click a default workflow in the Workflows window, then click the Status tab, the status properties are found in the Status table: double-click a status in the table, the Workflow Status: [Status Name] dialog appears.

## If No Criteria Apply At Exit, Notify

**Answers the question**: Who should be notified if the routable object does not match its present criteria (that is, it is stuck)?
**General default (in default workflows)**: $CHANGEANALYST (see Routing Manager Variables on page 17)

This property indicates who is notified by automatic email if the routable object cannot advance to the next status because does not meet the conditions defined by the criteria in the current status' criteria-specific properties. If a routable object cannot be promoted to the next status in the workflow because there are no criteria that permit the promotion, the defined people are notified by automatic email.

Use the Address Book dialog to add users, user groups, Originators, Approvers (all approvers for statuses the routable object has already gone through), and Observers (all observers for statuses the routable object has already gone through) to be notified.

**Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

## Manual Valid Next Status

**Answers the question**: Besides the next status in the status list, what are other statuses the routable object could go to from the present one?
**General default (in default workflows)**: (no general default)

For each status, use the Choices–Selected dialog to define additional valid "next statuses" that the routable object can move into from the current status. The system understands that the *next* status in the workflow's status list is always valid, so that status is not present in the Choices list. (The current status is, of course, never a valid "next status.") The system does *not* assume validity for Complete (since there is *no* status after a routable object arrives at Complete), Cancel or Hold (since they are never part of a workflow), so these must be specified.

**Caution** In assigning privileges to move a workflow to the Cancel status, the administrator must recognize the potential for user-error. Because a Canceled workflow can **not** be re-activated by any means, the fact that it is relatively easy to change a workflow to Cancel must be recognized and allowed for.

### Save Routing Slip Content when Advancing Change

When the user manually advances a workflow to the next status, the routing slip appears to be filled in. This contains approvers, observers, "notify" users, and comments, and the user may spend

several minutes completing the slip. When user clicks **Route**, Agile performs an automatic workflow status audit and presents the user with a dialog if a problem is found. The user is asked if he wants to continue: previously, if No was selected, the user was returned to the routable object and lost all the work; this has been fixed so the routing slip content is preserved.

### Send Workflow Status Comments and Notifications to all Previous and Current CCB Members

When sending comments from a Review or Released status in a change, the user can specify whether the comments are to be sent to only the *current* review or released status reviewers – that is, for the workflow's current status – or if comments are to be sent to all reviewers in both current and previous Review and Released statuses. A dropdown box (after the "Notify Reviewers for" checkbox) contains Current **Status** and **Current and Previous Statuses** for selection.

## AutoPromote

**Answers the question**: Will the system automatically promote the routable object to the next status?
**General default (in default workflows)**: No

Use the AutoPromote property to permit a routable object to automatically advance to its next status. For each Review and Released status type, you can determine if the workflow can be autopromoted to the next status.

When a routable object enters a status where AutoPromote is enabled, the system immediately attempts to autopromote it; if there are no approvers and all the required fields are complete, the routable object advances to the next status. Status advance is usually more involved than that scenario: for example, Review statuses will often have one or more approvers (and all approvers at a current status must approve the routable object for it to advance), and required fields can easily block autopromotion to the next status.

Also, the Change Status Approver Rejected Change SmartRule must be set to Allow for an approver's rejection not to hold up autopromotion of the workflow.

**Note** Autopromote checks required fields only once when the workflow *enters* a specific status. However, when the last required field is populated, the system registers this fact and the routable object does autopromote to the next status.

**Note** However, Autopromote does check the workflow every time an approver either approves or rejects. If there are more approvers who have not yet approved or rejected, the status is not advanced; when the final approver approves or rejects, the Autopromote function evaluates according to the above-named SmartRule and promotes the status or not accordingly.

If not all approvers have approved the routable object, anyone who has the Override privilege can still manually promote it. If the routable object is rejected by an approver or other participant with appropriate permissions, the routable object goes to the status defined in If Rejected, Set Status To (see If Rejected, Set Status To), which is done automatically, whether the AutoPromote feature is on or off.

The AutoPromote feature needs to be set in relation to the Override privilege and the following two SmartRules:

- Change Status Approver Rejected Change

- Change Status Observer Rejected Change

| | |
|---|---|
| **Caution** | When these two SmartRules are set to Warning, and an approver or observer rejects to the routable object, the system still autopromotes the routable object and records the warning on the **History** tab of the routable object. |

You can define required fields that, when filled in on a routable object, determine if it can be promoted to the next status (see Entry Required Fields and Exit Required Fields). All required fields for a status must be filled in before it can be autopromoted to the next status. For example, when the last user approves the routable object and all required fields are present, the routable object immediately advances to the next status.

| | |
|---|---|
| **Caution** | Do not alter any property before the Released status in any workflows for CTOs (content transfer orders). For instance, if you set AutoPromote to Yes, the CTO output files will not be generated. |

# If AutoPromote Fails, Notify

**Answers the question**: Who should be notified if AutoPromote fails?
**General default (in default workflows)**: $CHANGEANALYST (see Routing Manager Variables on page 17)

For each Review and Released status type, the Agile PLM administrator defines who is notified by automatic email if AutoPromote fails.

When the routable object is approved by all approvers, the system attempts to autopromote the routable object. AutoPromote fails if:

- Not all the required fields are filled in
- SmartRules are not adhered to

Autopromote checks required fields only once when the workflow *enters* a specific status. However, when the last required field is populated, the system registers this fact and the routable object does autopromote to the next status.

If the autopromotion fails, this fact is captured and automatically emailed to users and groups identified in the current status's setting for If AutoPromote Fails, Notify.

## Ability to Fail Autopromote More than Once within same Change Status

This feature notifies whomever is specified in the status's If Autopromote Fails, Notify setting each time a change order fails to autopromote if new approvers were added since the last autopromote failure. Additionally, History is updated. Autopromote fails when a new approver is added but doesn't complete all workflow-required fields.

Use the Address Book dialog to add users, user groups, Originators, Approvers (all approvers for statuses the routable object has already gone through), and Observers (all observers for statuses the routable object has already gone through) to be notified.

The email regarding failure of autopromotion is sent only when there is a failure after the last approver (designated for that status) has approved the routable object. The system sends *one*

email notification of failure to autopromote.

**Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

# Ad Hoc Approvers/Observers

**Answers the question**: Can “ad hoc” approvers and observers be added once the routable object has entered the status?
**General default (in default workflows)**: Yes

For each Review and Released status type, the Agile PLM administrator defines whether approvers and observers can be added as a routable object actively progresses. When this property is set to Yes, users with Add Approver and Remove Approver privileges can add or remove approvers and observers on the **Workflow** tab of a routable object in a Review or Released status type. They are also prompted with an Ad Hoc Approvers/Observers window during the release cycle.

**Note** A user with the Remove Approver/Observer privilege may successfully remove an observer from one status and then not be allowed to remove approvers and observers on the next status; this indicates the Ad Hoc Approvers/Observers property for the latter status is set to No.

# If Rejected, Set Status To

**Answers the question**: Should the status be automatically changed if a user rejects the routable object? (If so, enter the appropriate status.)
**General default (in default workflows)**: (none)

For each Review and Released status type, the Agile PLM administrator defines If Rejected, Set Status To. If the routable object is rejected by an approver or other participant with appropriate permissions (this status property does not apply to observers), the routable object goes to the status defined in this setting. This is done automatically, whether the AutoPromote feature is on or off.

**Note** Although it is possible to set this property so the routable object moves forward despite a rejection, Agile recommends that it be used to prevent the routable object from advancing until the reason it was rejected is examined. Note that if this status property is set to null (no setting), then when a user rejects the change, the workflow will advance to the default status, that is, the next status in the workflow.

# If Rejected, Notify

**Answers the question**: Who should be notified if the routable object is rejected?
**General default (in default workflows)**: $CHANGEANALYST (see Routing Manager Variables on page 17)

For each Review and Released status type, the Agile PLM administrator defines If Rejected, Notify. This property indicates who is notified by automatic email if the routable object is rejected by an approver or observer.

Use the Address Book dialog to add users, user groups, Originators, Approvers (all approvers for

statuses the routable object has already gone through), and Observers (all observers for statuses the routable object has already gone through) to be notified.

**Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

## Reminder Period

**Answers the question**: How long after a routable object has been routed for approval should approvers be sent email reminders to approve or reject it (at that status)?
**General default (in default workflows)**: 48 hours

The reminder period is a period of time defined for each Review and Released status type in all workflows. When the reminder period elapses, if an approver has not approved or rejected a routable object, a reminder email notice is sent to the approver. The routing manager receives a similar email notice, which also includes a list of the approvers who received reminder email.

If you delete an existing value or attempt to leave the field empty (null), clicking **Save** results in a 0 being placed by the system. A value of 0 is the same as disabling the property, and no reminder notifications will be sent.

**Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

Each Review and Released status type has its own reminder period. The reminder period is set in hours and begins when the routable object enters each Review or Released status type. If a workflow has multiple Review or Released status types, each of these may have reminder periods of different lengths. The reminder periods for the default workflows are all 48 hours.

## Review Escalation Period

**Answers the question**: How long should the system wait before escalating a routable object?
**General default (in default workflows)**: 96 hours

The review escalation period is a period of time defined for each Review and Released status type in all workflows. When the escalation period elapses, if an approver has not approved or rejected a routable object, email is sent to the approver's appropriate designated escalation person. The routing manager receives similar email, notifying her that the routable object has been escalated to the specified designated escalation person. The designated escalation person may approve or reject the routable object in the place of the original approver.

If you delete an existing value or attempt to leave the field empty (null), clicking **Save** results in a 0 being placed by the system. A value of 0 is the same as disabling the property, and no reminder notifications will be sent.

**Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

Each Review and Released status type has its own review escalation period. The review escalation period is set in hours and begins when the routable object enters a Review or Released status type. If a workflow has multiple Review or Released status types, each of these may have review escalation periods of a different length. The review escalation periods for the default workflows are

all 96 hours.

Escalation occurs only once per status for any sequence of user (or user group) to the designated escalation person.

### Designated Escalation Person

You can define a designated escalation person for each user from the **Users** base node (see Users and User Groups). This is useful for expediting the routable object when a particular user has not reviewed and signed off on the routable object. You can also set a designated escalation person for each global user group from the **User Groups** node.

All escalation assignments are done from the Administrator nodes. Each user or user group may have multiple designated escalation persons. (Personal groups do not have designated escalation persons.)

If the review escalation period elapses, the approver's designated escalation persons are notified by email that the approver has not responded to the routable object. Now the designated escalation persons may approve or reject the routable object in the place of the original approver. The designated escalation person's name is recorded in Agile Java Client on the **Workflow** tab and the **History** tab as the Signoff user.

> **Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

You determine whether a user's designated escalation persons can sign off a routable object at any time (= Always), or only after the escalation period has elapsed (= After Escalation). (See User Properties Defined on page 170.) The escalation person may or may not have appropriate permission to approve or reject the routable object, but the notification still goes through. If, for example, a user's manager is his designated escalation person but is not involved with the routable object approval process, the manager would at least know the routable object had been held up, and could take appropriate action.

> **Note** If the designated escalation person is a group, only one member of the group needs to sign off the routable object. When the routable object has been signed off, the routable object no longer appears in the Inbox of the other members of the group.

## Comments for Approval

**Answers the question**: Should comments be required in the Review status of this workflow when it is rejected?
**General default (in default workflows)**: Allowed.

For each Review status, the Agile PLM administrator can define whether approval comments are allowed, required, or prohibited. The default "Allowed" means that comments are allowed, but not required for the workflow in that status. "Required" means that when an Approver approves the workflow at that status, he must enter a comment for the workflow to proceed. "Prohibited" means that when an Approver approves the workflow at that status, he cannot enter a comment for the workflow to proceed.

## Comments for Rejection

**Answers the question**: Should comments be required in the Review status of this workflow when it is rejected?
**General default (in default workflows)**: Allowed.

For each Review status, the Agile PLM administrator can define whether rejection comments are allowed, required, or prohibited. The default “Allowed” means that comments are allowed for rejection, but not required for the workflow. “Required” means that when an Approver rejects the workflow at that status, he must enter a comment for the workflow. “Prohibited” means that when an Approver rejects the workflow at that status, he cannot enter a comment for the workflow.

# Criteria-Specific Properties Defined

This section describes all of the Criteria-specific Properties found in Agile PLM workflows. When you double-click a default workflow in the Workflows window, then click the **Status** tab, the criteria-specific properties are found in the Criteria table. First select a status (row) in the Status table, that status appears in the Criteria table; double-click the row in the Criteria table, the Workflow Criteria: [Criteria Name] dialog appears.

## Approvers and Observers

**Answers the question**: Who should approve or reject, and who should simply observe, the routable object in this status?
**General default (in default workflows)**: (none)

The Agile PLM administrator defines default approvers and observers, who are selected automatically—and notified by email—when the routable object enters the Review or Released status type. The default approvers and observers are assigned to a routable object based on the attribute values and the matching criteria of the workflow when it enters the Review or Released status.

> **Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

## Criteria Usage

**Answers the question**: What are the criteria-specific properties for this status?
**General default (in default workflows)**: All Change Orders for Default Change Orders workflow

Criteria are the conditions that specify how the reusable criteria work in the specific status. Each status must have at least one reusable criteria in its criteria-specific properties.

Criteria are checked when the routable object is about to exit the particular status. Criteria also define which criteria-specific properties apply. Several criteria-specific properties could be applied to a single routable object.

Because this is a comprehensive topic, for more information about criteria, see:

- Criteria on page 117

- Privilege Mask Component 3: Criteria on page 226
- In *Getting Started with Agile PLM*, please see the chapter “Finding Agile Data with Searches” for information about how to use the Advanced Search process, which is similar to the Create Criteria process.

# Default Change Analyst / Component Engineer

**Answers the question**: Who should be the default routing manager on this routable object? (See Routing Manager Variables on page 17.)
**General default (in default workflows)**: Default routing managers can be chosen; for Pending status only.

You can define different default routing managers for each workflow (component engineers for MCOs, price administrator for PCOs, change analysts for the other changes, and so forth). Under the criteria-specific properties in the Pending status, you can select a different default change analyst for each criteria. If there is no default change analyst, the user can enter one. If the **Change Analyst** field is left blank in Agile Java Client, all change analysts are notified if $CHANGEANALYST (or $COMPONENTENGINEER) is set by the administrator.

Users can edit the **Change Analyst** field, assuming they have the proper privilege, but the system does not assign the new change analyst until the routable object moves from Pending to the next status. If the change analyst has been selected by the user, but the workflow has been set up with a default change analyst who does not match, the user is warned.

If you define more than one change analyst (that is, if the routable object matches multiple criteria for different change analysts), the user is warned when the routable object moves from Pending to the next status. He sees a dialog box with a list of change analysts and can select one for that workflow.

For information about adding approvers and observers, see More about Affected Items: Adding Approvers and Observers on page 123.

# Notify Upon Entry

**Answers the question**: Who should know that the routable object entered a new status?
**General default (in default workflows)**: (none)

The Notify Upon Entry property indicates which users are notified by email when a routable object enters each status.

Use the Address Book dialog to add users, user groups, Change Analysts, Originators, Approvers (all approvers for statuses the routable object has already gone through), and Observers (all observers for statuses the routable object has already gone through) to be notified.

> **Note** If the email notification is disabled, email will not be sent. See Notifications on page 275.

You can set whether the change analyst receives email notification that the routable object has been released. The Notify Upon Entry property is also used to set the Notify defaults for the status transition dialog boxes for manual status transitions. The user can determine who is notified (for example, originator, approvers and observers, change analyst).

**Note** If a user does not have Field-level Read privilege for the **Change Analyst** field on a routable object, and the default change analyst is specified on the Notify Upon Entry list, the label No Privilege appears in the **Notify** field of the Change Status dialog box. No Privilege is, of course, not recognized as a valid user, and will generate an error message.

# Pass Release Audit

**Answers the question**: Should all the release audits (for example, SmartRules, required fields) be addressed before moving to the next status?
**General default (in default workflows)**: No
**User-available audits affected by Pass Release Audit**: **Status Audit and Release Audit**
**Status Audit answers:** Can the routable object advance to the next status?
**Release Audit answers:** Do conditions (in this status) support releasing the routable object?

Agile PLM performs a complete release audi*t aut*omatically when a routable object is promoted to its first Released status type.

At any time, an Agile PLM client user can use the **Audit Status** button or **Audit Release** button (or menu command) to initiate a status audit or release audit, respectively. These are discussed in Two Levels of Audit.

For each Pending, Submit, or Review status, the administrator specifies whether a routable object can advance from its present status with or without an audit, as follows:

- **Yes** — A release audit is performed, and the routable object does not advance unless the required fields are completed and the release checks and SmartRule conditions are met (see Release Audit, or Audit Current Status for Conditions to Release).
- **No** — The routable object can advance without being audited.
- **Warning** — During a status audit, you are prompted to choose whether to follow it with a release audit ("OK to perform Release Audit?") or not ("Cancel Release Audit").

  This prompt may result in either an Error or Warning. There is an anomaly that causes a Warning to appear when you would expect an Error: when the user does not have Override privilege, and Pass Release Audit = Warning or No, the Status Audit should give an Error (which requires user to fix conditions) instead of a Warning.

  **Note** Setting Pass Release Audit to Yes does, in effect, examine if the conditions in the present status would permit the routable object to be released, but this is not a cumulative process; that is, the audit does *not* consider the required fields of any *previous* or *future* statuses.

Even with the automatic release audit at the Released status, an advantage to setting up earlier audits (by setting Pass Release Audit to Yes for specific statuses) is to prevent impeding the routable object if it fails the audit at a Released type status. This can happen if the routable object has advanced past a status where a person who has pertinent information might enter it, for instance, the originator of the routable object who reviewed it at the Pending status, or a routing manager who reviewed it at the Submit status. It is better that a release audit reveal that something is missing at the status where the person who has the missing information can provide it.

For example, for ECOs, the release audit checks whether the rev number already exists. If you set

Pass Release Audit to Yes at the Pending status of a routable object, this forces the originator to supply the correct rev number.

Please see the Caution about this property in Override on page 219.

## Two Levels of Audit

There are actually two levels of audit available to the user, a status audit and a release audit, which are defined below. Every release audit includes a status audit.

The ability to audit a routable object does not require a privilege. The results of any audit can be copied to the Clipboard, pasted to a new document, and printed.

## Status Audit, or Audit Current Status for Conditions to Advance

The status audit can be thought of as “auditing the current status for proper conditions to advance.”

The **status audit** is performed automatically when you use the Change Status function or the workflow attempts to autopromote to the next status. You can perform a status audit at any time by using the **Audit Status** button or menu selections. It performs the following checks:

- Makes sure the Exit Required Fields for the current status are filled in.
- If the current status is a Review status type or a Released status type, makes sure that all approvers have approved.
- Checks the workflow settings of the Pass Release Audit property for the current status. If Pass Release Audit is set to either Yes or Warning, Agile PLM also performs a release audit.

## Release Audit, or Audit Current Status for Conditions to Release

The release audit can be thought of as “auditing the current status for proper conditions to release the routable object.”

Agile PLM performs a release audit automatically when the routable object is promoted to the first Released type status in the workflow. You can perform a release audit at any time by using the **Audit Release** button or menu selections.

The release audit performs the status audit for both the current status and the release status.

It performs the following additional checks:

- For ECOs, checks if the proposed new revision number on the **Affected Items** tab already exists.
- For ECOs, SCOs, PCOs and MCOs, checks if there are manufacturer redlines that will be lost after the routable object is released.
- For ECOs and SCOs, checks if there are BOM redlines that will be lost after the routable object is released.
- For ECOs, SCOs, PCOs and MCOs, checks if there are conflicts due to another ECO, SCO, or MCO being released first.
- For ECOs, SCOs, and MCOs, if the BOM Multi-level Recursive SmartRule is set to Disallow, checks for recursive BOM structures, that is, when a subassembly includes one of its parent items from an upper level of the BOM tree structure on its own BOM. (A recursive BOM

structure, if carried to its logical conclusion, would repeat indefinitely).

The BOM tree is evaluated from top to bottom. The Audit Release error message indicates, for each affected item with a recursive BOM, the first item found in the BOM tree whose first-level BOM table includes an item that appears at a higher level in the BOM tree, thus causing a recursive BOM structure. Although there may be multiple instances of recursive BOM structure in an affected item BOM tree, only the first found instance is displayed in the audit results.

- For ECOs, SCOs, and PCOs, depending on SmartRule settings, checks for duplicate items, duplicate find numbers and duplicate reference designators caused by a change rebase. (When there are multiple pending changes for the same item, when one change is released, the remaining pending changes are rebased on the latest released change.)
- For ECOs, SCOs, and MCOs, checks for duplication of the same manufacturer part on an item. (MultipleItemsPerManufPart is set to Warning or Disallow.)

  (Duplication of the same manufacturer part on an item occurs when an item has multiple pending changes which associate the same manufacturer part to that item and one change is released. The remaining pending changes are rebased on the latest released change. The remaining pending changes now include manufacturer part redline modifications to add a manufacturer part that already exists on the released item's Manufacturers table.)
- For ECOs, SCOs, and MCOs, checks for the use of the same manufacturer part on multiple items. (MultipleItemsPerManufPart is set to Warning or Disallow.)

  (Duplication of the same manufacturer part on multiple items occurs when multiple changes add the same manufacturer part to two or more item's Manufacturers tables, either by redline addition or redline replacement. When one of the changes is released, the manufacturer part's Where Used table is rebased to refer to the newly released item. The remaining pending changes now include manufacturer part redline modifications that would allow a single manufacturer part to be used by multiple items.)
- For ECOs, SCOs, and PCOs, if the Item Release First SmartRule is set to Disallow or Warning, then the routable object is checked to see if the child components have been released.
- Checks if any of the attachments of the routable object remain checked out.
- For every affected item, checks if any of the attachments remain checked out.
- Checks the Effectivity Date Order and Effectivity Gap SmartRule settings.
- Checks if items on the **Affected Items** tab have any other pending routable objects.
- For ECOs, validates the affected item's **Sites** tab **Make/Buy** setting with the site's **Site AML Allowed** and **AML Required on Buy** settings. For more information, see *Product Collaboration User Guide*.

# Exit Required Fields

**Answers the question**: What are the required fields on the routable object and its affected items?
**General default (in default workflows)**: (none)

Exit required fields are attributes or fields that must be completed for each Pending, Submit, Review, Released, and Complete status type before a routable object can exit that status.

Some required fields are uneditable, or defaults.

| | |
|---|---|
| **Caution** | Required fields are primary targets for status and release audits (see Pass Release Audit). In the default workflows, required fields are concentrated in the Review status type. In your custom workflows, however, you may wish to introduce required fields for other statuses. This will help users verify that specific required fields are passing the audits, which will prevent trying to pass a single audit containing many required fields at the first Released status. |

Please see the Caution about this property in Override on page 219.

When exit required fields are not filled in by the time the routable object moves from one status to another, the user sees a dialog box listing the fields that are required.

If a user does not have the privilege to modify a required field, the routable object cannot be advanced to the next status until a user with sufficient privileges modifies it.

For Criteria written against Affected Items fields, the required fields are calculated independently for each affected item, based on which Criteria were matched.

# Workflow Functionality Testing

It is strongly recommended that you thoroughly test the workflow before releasing it for general use. Depending on the size of your system, this may or may not be safely done within your users' production environment.

## Areas of Workflow Validation

This section outlines some aspects of workflows that you will want to validate, within or outside the production environment. You may find reasons to modify a workflow as you test its functionality and understand how it will support your change processes.

When testing workflow functionality, you should validate the following areas:

- Reusable criteria applied to the workflow and to each status
- Manual Valid Next Status settings for each status
- Status properties for each status
- Criteria-specific properties for each status
- Criteria Matching Type, for Matching Criteria of the workflow and for each status' criteria-specific properties
- All entry and exit required fields
- Autopromotions
- All notifications and approvals
- Reminder and escalation periods
- SmartRules
- Privileges for each role

## Quick Tips on Testing

- You must enable the workflow before it will be available in the Agile PLM clients (see Status Properties Defined on page 145).
- If you are testing a new workflow (not modifications to one of the default workflows), you will also need to create some basic Change Status privileges so users can move routable objects through the workflow (see Change Status on page 217). To move a routable object through the new workflow, users must have Change Status privileges for that specific workflow.
- Once you enable a workflow, you can make only limited changes to its configuration (see Modifying Properties in an Enabled Workflow).
- To make more extensive modifications, you must disable the workflow (see Disabling, Changing, and Reactivating Workflows).
- Additionally, even if you disable the workflow, you may not make certain modifications if a routable object is in the workflow, including deleting or adding statuses from the status list (see What You Cannot Modify If a Workflow Has Any Routable Objects Applied). You can use Save As on a test workflow to create a new disabled workflow that will allow more extensive modifications.

In addition to functional testing, which determines if you have configured Agile PLM workflows correctly, it is recommended that you validate your business process and data migration systems. Consulting services are available from Oracle Consulting – Agile Practice consultants, who can provide you with guidance and assistance in creating business process and data migration validation plans. During the business process validation, you should use your proposed workflows to accomplish real scenarios and simulate typical usage. These activities will help you refine and confirm your workflow configuration and allow you to document your change processes in the context of the Agile PLM solution for future reference and training purposes.

# Migrating Workflows to Production

If you are in the first phase of an Agile PLM implementation, this section may not apply to you, because you do not yet have a separate production environment, and migration of initial Agile PLM configuration is a normal part of your production cutover process.

## Quick Tip on Migrating Workflows

Quick Tip on Migrating Workflows

- In the Workflows window, use **Export** and **Export All** to export workflows, and use **Import** to migrate workflows from a test to a production environment, greatly reducing the number of manual steps required (see Administrator Import and Export on page 27.

## Workflow Migration Process

Once you have validated your workflow configuration, you need to implement the workflows in your

production Agile PLM system. The following steps outline the general process of migration.

1. Complete validation of workflow configuration in a test environment.
2. Run the following Administrator reports and save them to a local drive. Run additional reports as needed.
   - Agile Classes Report
   - Workflows Configuration Report
   - Criteria Library Configuration Report
   - Users Configuration Report
   - User Groups Configuration Report
   - SmartRules Configuration Report
3. Ensure that you have your workflow configuration clearly documented.

# Changing Active Workflows

You may have to change a workflow that has been enabled. Some possible situations and your options are detailed in this section.

**Note** It is not possible to delete a workflow that has been assigned to a routable object, even if the assignment has been withdrawn and the workflow is not assigned to any other routable object. We recommend either disabling an unwanted workflow, or modifying it so that it is once again usable.

## Modifying Properties in an Enabled Workflow

You can modify a workflow that is enabled and visible to your Agile PLM users, either with or without a routable object moving through it. If the workflow is already in use (a routable object has been assigned the workflow), you cannot add statuses, although you can make other modifications, as described below.

**Caution** If you add a status to an enabled workflow, you *must* modify or create appropriate permissions and other Change Status privileges for each role in your Agile PLM system.

**Note** When you attempt to add a status to an enabled workflow, the workflow's Enabled property automatically changes to No.

### When You Can Disable, Modify, and Re-enable a Workflow

If no routable object has been applied to a particular workflow, there is no problem with disabling it, modifying it, and re-enabling it. (To save duplicating efforts, you might email people who could be developing or close to originating a new routable object.)

### What You Can Modify If a Workflow Has Any Routable Objects Applied

You *can* alter many properties after a workflow is enabled and a routable object is actively

progressing. These properties are listed in the following table. You must disable the workflow to alter settings for properties. Also, be aware that a change analyst or an approver may want to react to a progressing routable object while it is disabled.

When you disable a workflow, these two best-practice actions are recommended:

- Notify appropriate people that the workflow is going to be disabled, and
- Alter the workflow promptly, enable it, and re-notify the affected users.

| Properties of workflow | Status properties of each status | Criteria-specific properties of each status |
|---|---|---|
| Workflow name | Ad Hoc Approvers/Observers<br>AutoPromote<br>If AutoPromote Fails, Notify<br>If No Criteria Apply At Exit, Notify<br>If Rejected, Notify<br>If Rejected, Set Status To<br>Manual Valid Next Status<br>Reminder Period<br>Review Escalation Period | Approvers<br>Default Change Analyst/<br>Default Component Engineer<br>Notify Upon Entry<br>Observers<br>Pass Release Audit<br>Required Fields |
| Properties of statuses | | |
| Status name | | |
| Criteria-specific properties | You can add new criteria-specific properties to any status. | |
| **Note**: Depending on the setting being altered, routable objects already in process may or may not use the new setting if they are already in the status. For example, changing the Notify Upon Entry setting does not resend email about a routable object already in the status that is changed. | | |

## What You Cannot Modify If a Workflow Has Any Routable Objects Applied

Once a routable object has been originated in a default or new workflow, and that object has gone through even a single status transition, you cannot modify certain things:

- You cannot add or delete a status to that workflow.
- You cannot alter the order of the statuses in that workflow.

If these are the elements of an active workflow that you need to alter, see Disabling, Changing, and Reactivating Workflows.

# Disabling, Changing, and Reactivating Workflows

Despite best efforts at fully defining your workflow processes and the many elements of a customized workflow, naturally the time will come when an active workflow—with multiple routable objects already applied and moving through it—simply needs to be fixed.

**Caution** All modifications to enabled workflows have the potential to cause “ripple effects” that have to be detected, evaluated, and corrected across multiple nodes in Administrator. In particular, alterations to workflow statuses have implications for transfer orders (see *Agile Content Service User Guide*) and process extensions (see Process Extensions on page 103).To fix an active workflow:

1. Move each active routable object using that workflow to the Pending status type.

   **Note** This step is no longer absolutely required, and may not be necessary if you are doing a small and quick modification (see Modifying Properties in an Enabled Workflow). This precaution is helpful when the change may take some time and many objects and people may be affected.

2. Disable the workflow.
3. Rename the workflow, using the Save As procedure (described in Creating New Workflows with Save As on page 135).
4. You may now wish to rename the old workflow, especially if you decide it can still be useful but has a different purpose.
5. Modify or fix the new workflow.
6. When you are confident that the existing problem has been remedied, enable the workflow.
7. Apply the routable objects from the old to the new workflow.

# Chapter 12

# Account Policy

**This chapter includes the following:**


- Before You Begin ..... 161
- Configuring the Account Policy ..... 161


To provide an enhanced level of access security in Agile PLM, you can set a systemwide user account policy. The features of this policy include password aging, length, uniqueness, and lockout.

## Before You Begin

Before configuring systemwide user account policy for Agile PLM, make sure you answer the following questions:

- Will you be using an LDAP system to create Agile PLM users? You may decide to set up **Account Policy** functionality on your LDAP server. See LDAP as a Node in Administrator on page 168.
- Do you want user passwords to expire? If so, how often (number of days)?
- Do you want users to be able to change their passwords at any time? If not, how often (number of days) should they be changed?
- What is the minimum number of characters permitted?
- Do you want to prevent users from using the same password over again? If so, how many previous passwords do you want the system to remember?
- Do you want a systemwide lockout policy to prevent unauthorized attempts to log in to the system?
- If a user is locked out, do you want yourself or others to be notified?

## Configuring the Account Policy

**To configure your systemwide account policy:**

1. Under **User Settings,** double-click **Account Policy**. The Account Policy window appears.
2. Configure your systemwide account policy by setting the properties described in the table below.

| Property | Description |
|---|---|
| **Account Lockout** | Controls whether the system remembers failed logons. When this property is set to Disabled, the system does not remember failed logons, and these properties are not in use: Logon Attempts, Reset Count Time, Logout Duration, and Account Lockout Notify User. When Account Lockout is set to Enabled, these properties are enabled. |
| **Logon Attempts** | Enabled when Account Lockout is set to Enabled. Enter a value between **1** and **999,999,999**. The default is 3 failed attempts, in which case the lockout of that user account occurs upon the third failure. |
| **Reset Count Time (in minutes)** | Enabled when Account Lockout is set to Enabled. Enter a value between 1 and **999,999,999**. The default is 30 minutes, after which that user account reverts to allowing the number of attempts set in Logon Attempts.<br>**Note** A user who is locked out, and whose Lockout Duration value is set higher than Reset Count Time, will still be locked out when Reset Count Time expires, while another user can log on once Reset Count Time has expired. |
| **Lockout Duration (in minutes)** | Enabled when Account Lockout is set to Enabled. Enter a value between **1** and **999,999,999**. The default is 0, which means the user is locked out until you reset that user account. Lockout Duration and Reset Count Time do not "trump" each other, meaning a user, upon being locked out, will have to wait for the higher value of these two properties to expire. |
| **Maximum Password Age (in days)** | Enter a number of days you want as a limit of how long passwords may remain in effect. Default is 0. |
| **Minimum Password Length** | This value sets the minimum number of *characters* that any user's password can be. Default is 1. |
| **Password Uniqueness** | The value 0 (default) permits users to re-use passwords they have used before when it comes time to change passwords. To prevent users from reusing passwords, enter a number, which is the number of passwords that the system remembers for each user, and does not allow to be re-used.<br>For example, with a value of 3, users will not be able to re-use their first password when it is time to select their fourth password. When it is time to select their fifth password, however, each user could re-use their first password, because the system remembers their second, third, and fourth passwords, but not their first. |
| **Account Lockout Notify User** | Enabled when Account Lockout is set to Enabled. To assign users to receive email notification if a user is locked out, click the button at the right for the address book to appear, and select any number of users to be notified.<br>On the **User Groups** tab of the address book, when you select a user group, only the button that adds all the members of the group is enabled; you can, however, add all the members of a group and then remove individuals from the Recipients list. |

**Note** The account lockout rules apply to all Agile PLM clients, including Java Client, Web Client, ChangeCAST, AIS, ACP, Agile Drive, and SDK.

## Administrator Reset of Locked Out Account

If a user is locked out of their account because of entering the wrong password more times than **Logon Attempts** allows, and the **Reset Count Time** is a high value, you should reset the user's login password. This will reset the user's locked status and allow them to access the system.

**Note** An additional setting concerning passwords (for Web Client users) is under **Preferences** node. See Allow Password Reset on page 377.

# Chapter 13

# Users

**This chapter includes the following:**


- Adding New Users ..... 165
- About Usernames ..... 167
- About Passwords ..... 167
- LDAP as a Node in Administrator ..... 168
- Modifying User Properties ..... 169
- User Properties Defined ..... 170


From the **Users** node, you can give new users access to Agile PLM or modify the properties of existing users, including changing user group, role, and escalation approval settings.

The **Users** node lets you view and configure the following aspects of Agile PLM:

- Add new users
- Modify any user properties such as licenses, roles, sites, and passwords
- Disable or delete users

The process of adding users involves establishing the properties that govern their access to Agile PLM applications. For example, their passwords allow them to log in, and their role assignments establish their access to Agile PLM objects from point of discovery forward. It is important to consider precisely what access and permissions your users require.

**Important** Also available is the possibility of integrating aspects of your PLM system with Single Sign-On (SSO) capability. With SSO configured and deployed for your PLM system, a user that has signed in to the system once (for instance, through the corporate portal) is not prompted again by a "login" dialog (see Appendix A, Configuring Single Sign-On on page 413).

**Note** A “supplier user” is a particular kind of user who is associated with a Supplier; supplier users, or "contact users" may not be created by the administrator but by certain Manager roles. You may assign the Organization Manager or Compliance Manager role to an Agile PLM user, and this person creates Suppliers and populates them with contact users who can respond to RFQs or RFIs via their (Restricted) role. For more information, see *Product Cost Management User Guide* and *Product Governance & Compliance User Guide*.

# Adding New Users

Once you have assembled information about new users and assessed their access requirements, you can add them to the Agile PLM system.

Users created in an LDAP system must be imported to Agile PLM and synchronized in the Agile PLM database using the **Refresh** button. See LDAP as a Node in Administrator on page 168.

# Before You Begin

Before creating a new Agile PLM user, make sure you answer the following questions:

- What does this user need to be able to do in Agile PLM? What default roles are required for this user?
- What should this user be prevented from doing in Agile PLM?
- Will this user need to have separate Login and Approval passwords?
- On which Agile PLM lists will the user's name appear?
- Which Agile PLM searches should the user be able to use?
- Is the user a Power User? A Power User can log in at any time and is not counted as a member of the concurrent user pool.

**To create a new user in Java Client:**

1. Under **User Settings,** double-click **Users**. The Users window appears.
2. Click the **New** button. The Create User dialog box opens.
3. Enter values for **User ID, First Name**, **Last Name**, and **Email**.
4. Type the **Username** and **Password**. Repeat the password in the **Retype Login Password** field.

   **Note** Passwords are case-sensitive. For example, if you enter the password here in all capital letters, then the user must always enter the password in all capital letters when logging in to Agile PLM.

5. If the user requires a separate approval password, uncheck **Use Login Password for Approval Password**, then enter an approval password (the new user's approval password) in the **Approval Password** and **Confirm Approval Password** fields.
6. Click **OK**. The setup window for the new user appears.
7. Under **Profile**, click the **Role(s)** list and assign roles to the user. Remember, the user cannot do anything in Agile PLM until a role is assigned.
8. In the **Profile** section, enter other values as needed. These fields are not required and may be filled in later.
9. On the **Preferences** tab, enter values for the properties. You can accept the default values.
10. Click **Save**.

**To create a new user in Web Client:**

1. From the left navigation pane choose **Tools and Settings > Address Book > Users**.
2. Click **Add**. The Create New User window opens.

3. Type the **Username** and **Login Password**. Repeat the password in the **Confirm Login Password** field.

   **Note** Passwords are case-sensitive. For example, if you enter the password here in all capital letters, then the user must always enter the password in all capital letters when logging in to Agile PLM.

4. If the user requires a separate approval password, uncheck **Use Login Password for Approval Password**, then enter an Approval Password (the new user's approval password) and Confirm Approval Password.
5. Enter values for **First Name**, **Last Name**, and **email**.
6. Click **Save**. The General Information page of the newly created user appears ready to edit.
7. To open the **Role(s)** list, click **Launch the Palette** to the right of the field. Assign roles to the user. Remember, the user cannot do anything in Agile PLM until a role is assigned.
8. In the **Profile** section, enter other values as needed. These fields are not required and may be filled in later. Click **Save**.
9. To confirm the addition and review the new user's properties, click **Tools and Settings > Address Book > Users**. Double-click the new user ID in the table of users. The new user's page appears.

# About Usernames

For security reasons, Agile PLM usernames must be unique. You can create multiple users with the same name (first and last), as long as the username is unique.

If there is an active or inactive user with the same username as a deleted user and the deleted user is undeleted, you must change the username of the deleted user to make it unique. All history references from other objects to the undeleted user remain intact.

**Note** The "Username" property is now called "User ID." Both of these terms are seen in the Agile PLM clients and documentation, and they can be thought of as equivalent.

**Caution** To prevent frequent user inconvenience, it is important to periodically evaluate your license requirements with your Agile Account Executive, and upgrade as needed.

# About Passwords

When you add new users, you assign passwords so that they can log in to Agile PLM for the first time. Users can then change their passwords themselves based on the settings in their Password property and the systemwide account policy. The Agile PLM administrator can also change a user password later if necessary.

Although the account policy established from the **Account Policy** node governs password expiration, length, and uniqueness, you can specify settings from LDAP.

For additional security, users can have different passwords for login and approval. For more information, see Changing a User's Password on page 169.

For more information, see Account Policy on page 161.

## Change Internal User Password

There are four “special users” in Agile PLM, which are used for internal purposes.

| Special User | Purpose |
|---|---|
| **superadmin** (internal user) | used internally to gain access to all data |
| **agileuser** (internal user) | used internally to gain access to all data |
| **ifsuser** (external user) | needed for access by File Manager |
| **etluser** (external user) | needed for access by ETL tool |

The difficulty with these users is that these users are not exposed through the UI, that is, they exist in the PLM database table but are not visible in Java Client. However, anyone who knows the password can access agile data through SDK.

You can change the default passwords for the “IFS” user (file server), the “ETL” user (Datamart), and the “Agile” user.

**Important** The “Super Admin” user password cannot be changed.

**To reset, for example, the “ifsuser” password:**

1. Under **User Settings,** double-click **Users**. The Users window appears.
2. In the menu bar, click the **Change Internal Password User** dropdown list.
3. Click **Change Password of “ifsuser”**. The Change Password of “ifsuser” dialog opens.
4. Fill with appropriate values for the Old Password, New Password, and Retype New Password fields. When you are finished, Click **OK**.
5. When the password for ifsuser is reset, the following steps must be performed:
   - Run **<AgileHome>\agileDomain\bin\encryptpwd.cmd** to encrypt the new password.
   - Paste the new password to **agile.properties** under **<AgileHome>\agileDomain\config** and **server.conf** under **<AgileHome>\apache-tomcat\webapps\Filemgr\WEB-INF\classes\com\agile\webfs\configuration** and **<AgileHome>\apache-tomcat-6.0.18\webapps\webdav\WEB-INF\classes**.

     If the **ifsuser** password is changed during installation, only this second step needs to be performed.

   The File Manager needs to be restarted for the configuration change to take effect.

# LDAP as a Node in Administrator

Administrator has an **LDAP** node under the **Server Settings** node folder in which to configure a Lightweight Directory Access Protocol. Please see LDAP.

# Modifying User Properties

You can modify any of the fields for existing users from the **Users** node. The following is a general process for changing property settings.

**Note** For information about the filter bar at the top of the Users window, see Filtering Data on page 9.

**To modify a user's fields in Java Client:**

1. Under **User Settings,** double-click **Users**. The Users window appears.
2. Double-click the user's name in the list. That user's window appears.
3. Modify fields directly, make a selection from a list, or make changes in a dialog box.
4. Click the **Save** button.

**To modify a user's fields in Web Client:**

1. Open the User object, either from a search result or from the **Users** tab of **Address Book**.
2. Click **Edit**. The fields become editable. You can directly edit a field, make a selection from a list, or make changes in a dialog box. Modify the fields you want.
3. Click the **Save** button.

The next time the user logs in, he will see the results of changes you have made to his user properties. However, for some properties (for example, **Time Zone** and **Receive Email Notification**), you must log out and log back in to Web Client to see your changes to these properties.

# Changing a User's Password

When you are adding new users, you assign passwords so that users can log in to Agile PLM for the first time. Users can then change their passwords within Agile PLM if necessary. The Agile PLM administrator can also change a user's password.

Each user can have both a login password and an approval password, although a single password can be used for both purposes.

**Important** If the Signoff User Dual Identification Type preference is set to Login Password then the Login and Approval password may not be identical.

You can specify whether to establish separate login and approval passwords, or to use a single password for both. The login password is required to establish access to Agile PLM. In addition, the system prompts users who approve objects, such as changes, for an approval password.

Using separate approval passwords provides additional security. If your company requires this type of security, or requires the assurance that only the correct person can approve an object (perhaps for audit purposes), set **Use Login Password for** Approval to **No**.

**To change a user's login or approval password in Java Client:**

1. Under **User Settings,** double-click **Users**. The Users window appears.

2. Double-click the user's name in the list. That user's window appears.
3. Click the **Reset Password** button to change the login password.
4. The Reset Password (or Reset Approval Password) dialog box appears. Select the Reset Login Password checkbox. Enter the new login or approval password, remembering that passwords are case-sensitive. If you enter the password here in all capital letters, then the user must always enter the password in all capital letters when logging in to Agile PLM.
5. Confirm the password by retyping it in the Retype New Password field. Click **OK**.
6. Click the **OK** button in the confirmation message that appears.

**To change a user's login or approval password in Web Client:**

1. Open the User object, either from a search result or from the **Users** tab of **Address Book**.
2. Click the **Reset Passwords** button to change the login password.
3. The Reset Passwords window appears. Enter the new login or approval password, remembering that passwords are case-sensitive. If you enter the password here in all capital letters, then the user must always enter the password in all capital letters when logging in to Agile PLM.
4. Confirm the password by retyping it. Click **Save**.

## Changing a User's Role Assignments

You can modify a user's role assignments from the **Users** node, the **Roles** node, or the user object.

- From the **Users** node, you can change an individual user's role assignments by selecting a user and clicking the **Assign Roles** button.
- From the **Roles** node, you can see and change the users assigned to a particular role from the **Users** tab of a particular role.
- From the user object, you can see or change an individual user's role assignments by editing the roles under the **Profile** heading of the **General Information** page.

# User Properties Defined

The table below lists and describes the properties common to all users in the Agile PLM system. The **General Info**, **Preferences**, **Escalations**, and **User Groups** tabs are documented. **Share**, **Subscription**, **Attachments**, and **History** tabs are all automatically populated, and do not have fields or properties.

**Note** Preferences for users are found on the **Preferences** tab of any user object. There are three sub-categories of user preferences: system-related, format-related, and display-related. These "user preferences" should not be confused with **Administrator > Server Settings > Preferences** node of "systemwide preferences."

| Property | Description (default is underlined) |
|---|---|
| | |

| Property | Description (default is underlined) |
|---|---|
| **General Info tab, required fields** | |
| **User ID (username)** | The user's Agile PLM login user identification, or username. It must be unique in the Agile PLM address book. The maximum is 128 characters. See About Usernames on page 167. |
| **First Name** and **Last Name** | The user's public name. Neither of the user's Names fields has to be unique in the system, only the user's **User ID** must be unique (see About Usernames on page 167). |
| **Email** | The user's valid email address for change notification/routing. *Example*: danny.design@agile.com. **Note**: If there is not a valid email address in the **Email** field, the user will not receive any email notifications. |
| **General Info tab, optional user information** | |
| **Status** | Active or Inactive. (This is the same thing as Enabled or Disabled, respectively.) |
| **Title** | The user's title, for example, Senior Engineer. |
| **Address** fields | There are four address fields that are used for informational purposes only: **Address** (street), **Geography**, **City**, and **Postal/Zip Code** . The Geography field is a series of dropdown lists. |
| **Phone** number fields | The user's **Business Phone**, **Home Phone**, and **Mobile Phone** numbers (for informational purposes only). |
| **Fax** | The user's fax number (for informational purposes only). |
| **Pager** | The user's pager number (for informational purposes only). |
| **Secondary Email** | The **Secondary Email** is optional, but when the field is filled in, the user receives notifications at both email addresses. |
| **General Info tab**, **Profile fields** | |
| **Role(s)** | The user's role assignments. This property determines a user's access to the objects in Agile PLM from the point of discovery forward. The default roles (automatically assigned when you create a user) are My User Profile and Read And Discover By Creator. For more information, see Roles on page 187, and also Read and Discover Objects I Created Role. |

| Property | Description (default is underlined) |
|---|---|
| **Lists** | **Note**: these lists are particular to routable objects and their workflows, as specified below. They are not editable at the Lists node; names are added or removed from these lists based on this setting in a user's profile.<br>▫ **Change Analyst List** controls whether the user's name appears in the **Change Analyst** list on the **Cover Page** of Engineering Changes.<br>▫ **Component Engineer List** controls whether the user's name appears in the **Component Engineer** list on the **Cover Page** of Manufacturing Changes.<br>▫ **Compliance Manager List** controls whether the user's name appears in the **Compliance Manager** list on the **Cover Page** of Declarations.<br>▫ **Price Administrator List** controls whether the user's name appears in the **Price Administrator** list on the **Cover Page** of Price objects (PCOs, Quote Histories, and Published Prices).<br>▫ **Quality Administrator List** controls whether the user's name appears in the **Quality Administrator** list on the **Cover Page** of Quality Change Request objects (CAPAs and Audits).<br>▫ **Quality Analyst List** controls whether the user's name appears in the **Quality Analyst** list on the **Cover Page** of Product Service Request objects (Problem Reports and NCRs). |
| **Searches** | **Change Analyst Searches** controls whether a user has the following in Java or Web Client:<br>▫ A Change Analyst Searches folder<br>▫ Change Analyst queries in My Inbox<br>**Component Engineer Searches** controls whether a user has a Component Engineer Searches folder in Java or Web Client.<br>**Quality Searches** controls whether a user has the following in Java or Web Client:<br>▫ A Quality Search folder<br>▫ Quality queries in My Inbox<br>**Content Manager**, **Price**, **Project**, **Supplier RFQ**, and **Sourcing Searches** |
| **User Category** | Power, Concurrent, or Restricted. See User Licenses. |
| **Use Login Password for Approval** | Each user has a login password and approval password (used to approve changes). Set **Use Login Password for Approval** to Yes to use one password for both. Set it to No if you require an approval password. Use the **Change Password** button at the top of the setup window to define new (case-sensitive) passwords for the user. |

| Property | Description (default is underlined) |
|---|---|
| **Allow Escalation Designation Approval** | If the user has identified one or more designated escalation persons (DEPs), this setting determines when the DEPs can approve or reject a routable object.<br>*Settings*:<br>After Escalation – The user's DEP can approve a routable object only after it has been escalated.<br>Always – The user's DEP can always approve or reject a routable object.<br>(For more information about designated escalation person status, see Assigning Escalation Persons to User Groups on page 180.) |
| **Home Organization** | The name of the company profile (organization name) for all users created from the **Users** node and the supplier name and supplier number for all users created from the **Supplier Groups** node. |
| **Default Site** and **Site(s)** | **Sites** are used for distributed manufacturing, and indicate all of the company's locations where the user is involved; selecting from the dropdown list for this property populates the dropdown list for Default Site. **Default Site** is the user's main base of work. |
| **Ship-To** address fields | **Home Ship-To** is the primary location where the user is responsible for sourcing activities.<br>**Authorized Ship-To** (called "Site" in earlier versions of PCM), used only by PCM, indicates all company locations where the user can initiate sourcing activity; selecting from the dropdown list for this property populates the dropdown list for **Home Ship-To**. |
| **Program Manager for Partners** | The partners for which this user is program manager (which is the change analyst for Package objects). See Setting Up User Groups as Partners on page 183. |
| **Rate** fields | There are three **Rates** fields: **Bill Rate**, **Labor Rate**, and **Overhead Rate**. The dropdown lists in these fields show all currencies entered in the **Currency Exchange Rates** node. You can select a currency different from the user's Preferred Currency (see **Preferences** tab). Used by PPM. |
| **Comment** | The Agile PLM administrator may enter a comment regarding the user. This comment does appear to the user when he or she clicks **Settings > User Profile**. |
| **Project Type** | Types of projects the user can participate in. Available values depend the Project Type List. |
| **Region** | Regions the user has belongs to. Available values depend the Region List. |
| **Division** | Divisions the user belongs to. Available values depend on the Division List. |
| **Product Line** | Product lines the user has access to. Available values depend on the Product Line List. |
| **Customer** | Customers the user is associated with. Available values depend on the Customer List. |
| **Launch Year** | Product launch years the user is associated with. Available values depend on the Launch Year List. |
| **Category 7, Category 8, Category 9, Category 10** | Optional Product Portfolio Management Dashboard fields. |

| Property | Description (default is underlined) |
|---|---|
| **Preferences tab: System Preferences** | |
| **Preferred Client** | This field controls the type of link provided in system notifications to the user. Select Web Client or Java Client to determine which client is automatically opened to the object to which the notification's link points.<br>**Important:** Agile's PCM, PG&C, and PPM solutions operate only in Web Client. Users who work primarily in those areas should have their Preferred Client set to Web Client. |
| **Preferred File Manager** | Select from the list of file servers. For best performance, select a file server that is a local server for the user. For example, for a user in the United States, select a server located in the United States. (For more information about file management servers, see File Management in Agile PLM on page 348.) |
| **Receive Email Notification** | Controls whether the user can receive automatically generated email notifications from the system.<br>*Settings* = Yes or No |
| **File Productivity Preference** | Controls whether the user has access to advanced productivity components, or standard components, or wants to be prompted in each situation. If set to Standard mode, non-supported files will not be opened automatically. Instead, the user is prompted to choose Save or Open.<br>*Settings* = Prompt, Advanced, or Standard<br>**Note**: this user preference will not appear in any user's Profile if the systemwide preference Allow Download of Productivity Components is set to No. (See Preferences on page 371.) |
| **Preferences tab: Format Preferences** | |
| **Language** | English is available with all Agile PLM licenses. Agile PLM includes Japanese, Traditional Chinese, Simplified Chinese, French, or German, per your license. |
| **Workweek Settings** | This setting is for future use only, and is not referenced in any Agile PLM module.<br>**Note**: this property is not applicable to any Agile PLM module including Product Portfolio Management and Microsoft Project schedules. |
| **Preferred Date Format** | The format in which dates are displayed to the user. Default is MM/dd/yyyy. |
| **Time Zone** | The time zone where the user is located. |
| **Preferred Time Format** | The format in which times are displayed to the user. The default is hh:mm:ss aaa (aaa = am or pm). |
| **Preferred Currency** | The currency in which prices/costs are displayed to the user. The default is U.S. Dollar. The dropdown list in this property shows all currencies entered from the **Currency Exchange Rates** node. |
| **Number Format** | The format in which numbers are displayed to the user. The administrator or user selects a type of format (for example, use commas for thousand separators and point for decimal separator; use nothing for thousand and comma for decimal). An example of each format is displayed in the dropdown list. |

| Property | Description (default is underlined) |
|---|---|
| **Preferences tab: Display Preferences** | |
| **Preferred Start Page** | Select between Home, Dashboard, Dashboard Financial, Dashboard My Activities, Dashboard Projects, and Dashboard Resources. |
| **Response Edit Mode** | Choose between Basic, Advanced Table Edit, and Advanced Wizard Edit. The Basic setting brings the user – generally a "supplier" who does not work at the "buyer" company – to a simplified Web Client user interface. The two Advanced settings bring the user to the familiar Web Client UI. For more information about these settings, see *PCM Supplier Handbook* and *PG&C Supplier Guide*. |
| **Static Table Headers** | Set to Yes to carry table headers to additional pages. |
| **Encode Type** | Select from: **Western European (ISO)**, **Japanese (Shift JIS)**, **Traditional Chinese (Big 5)**, **Japanese (EUC)**, **Simplified Chinese (GB2312)**, and **Unicode (UTF-8)**. |
| **Thumbnails** | Off or On Off is default |
| **Table Display Mode** | Standard or With Thumbnails |
| **Search Display Mode** | Only Thumbnails, Standard, or Standard with Thumbnails |
| **Show Timesheet** | Set to Yes to display the **Timesheet** tab. Default setting is No. |
| **Escalations** tab | |
| **Criteria**<br>and<br>**Notify Users** | Set **Criteria** and **Notify Users**, which is an individual user or user group (personal or global) to be notified when a workflow status exceeds its escalation time period without a signoff by the user. A user can have multiple designated escalation persons assigned to a variety of workflow-related criteria. For more information, see Assigning Escalation Persons to User Groups on page 180. |
| **User Group** tab | |
| **Group Name** and **Status** | This will list the user groups that the selected user is a member of. From here the administrator can add a user group membership for the user or remove the user from membership of a user group. A user can belong to multiple groups. These fields can be set in this tab or on the **Users** tab of the user group setup window. For more information, see User Groups on page 177.<br>Note that this does not display the Personal User Groups created by the user. |

# Chapter 14

# User Groups

**This chapter includes the following:**


- Creating a New User Group ..... 177
- User Group Properties ..... 179
- Modifying a User Group ..... 180
- Assigning Escalation Persons to User Groups ..... 180
- Adding User Groups as Approvers of a Change ..... 181
- Setting Up User Groups as Partners ..... 183


From the **User Groups** node, you can assign users and designated escalation persons to user groups, change a user group's name, or create new user groups.

You can also define project teams, partners, departments, business units, divisions, site-related groups and global groups and their assigned users. This configuration is reflected in the **User Groups** node of the Agile PLM address book for use in addressing email notifications.

Agile PLM administrators set up the user group configuration for their Agile PLM system from the following two nodes:

- From the **User Groups** node, you set up user groups by creating, naming, and assigning users to them. User groups are displayed in the Agile PLM address book. When a user clicks the **To** button in the Send Object dialog box in Agile PLM clients, the address book opens. You can select a user group in the address book. Any global user group assignments you make in the **User Groups** node are also reflected on the user's **User Groups** tab of the user's window or page.
- From the **Users** node, you configure user properties, including user group assignments. Any user group assignments you make from the **Users** node are also reflected in the Users property of that user group in the **User Groups** node. When you add new users in the **Users** node, you can assign them to one or more user groups. You can change these assignments later as necessary, from either the **Users** node or the **User Groups** node. For information about how to change a user's properties, see Modifying User Properties on page 169.

**Note** If you create a new user group and want to assign existing users, it is faster to configure the new user group's User property than each user's User Group property.

If you have upgraded to Agile PLM from a previous version of Agile Product Collaboration, and want to keep your existing organizational structures in the system, you can create subclasses of user groups called, for instance, Departments, Global Groups, and Partners.

# Creating a New User Group

Before creating a new user group, you should review the user groups you currently have. Run the User Group Configuration report (under the **Analytics and Reports** tab **> Standard Reports > Administrator Reports** node): it lists current user groups and the users assigned to them. (See Administrator Reports on page 20 for more information about reports.)

**To create a new user group in Java Client:**

1. Under **User Settings,** double-click **User Groups**. The User Groups window appears.
2. Click the **New** button . The Create User Group dialog opens.
3. From the **Subclass** dropdown list, select the type of user group.

   User Groups is the only subclass available if you have not created another one in that class. If you want to designate a new group to be of a type that has not been created, you must first create it from the **Classes** node for it to appear in the dropdown list.
4. Fill in a value in the **Name** field.
5. Select if the user group will be Global or Personal.
6. Click next to the **Users** field. The address book appears. Move users (from the **Names** list) or other user groups (from the **Groups** list) to the **Recipients** list.
7. Click **OK**. The new user group object appears.

   The new user group object is available to users from this point on. Its name appears under the **User Groups** node, and user group assignments are updated in the **Users** node. The new user group name also appears in the Agile PLM address book.

**To create a new user group in Web Client:**

1. From the left navigation pane, click **Tools and Settings > Address Book > User Groups**. The User Groups page appears.
2. Click **Add**. The Create New User Group window opens.
3. From the **Type** dropdown list, select the subclass type of user group.

   If you want to designate the new user group to be of a type that has not been created, you will have to create that from the **Classes** node in Java Client before it will appear in the dropdown list.
4. Fill in a value in the **Name** field.
5. Select if the user group will be Global or Personal.
6. Click **Save**. The User Group General Information page appears and is ready to edit.
7. Enter user group properties as needed. For more information, see User Group Properties on page 179.
8. When you are finished, click **Save**. Then click the Users tab.

   To create the new user group without assigning users, click **Finish**. The new user group name appears on the User Groups page on the **Users** tab, and in the Agile PLM address book.
9. To assign a user to the user group, click **Add**. You may type a name to find the user, or search a list of existing users.
10. Select one or more names and drag and drop them onto the table of users for the user group or double-click.

    The new user group name appears on the User Groups page, and user group assignments are updated on the Users page. The new user group name also appears in the Agile PLM address book.

# User Group Properties

The following table lists user group properties.

| Property | Description |
|---|---|
| **Name** | Allows you to enter a user group name of up to 29 characters. |
| **Type** | Select a subclass. The out-of-box subclass is User Group. The administrator can create other subclasses in the User Groups class; these will appear in this dropdown list. |
| **Description** | Description of the user group. |
| **Status** | Whether the user group is active or inactive. An active group appears in the address book, an inactive group does not. |
| **Global/Personal** | Global user groups can be used for all users. Personal groups are seen and used only by their creators. |
| **Roles** | Roles applied to groups apply to all the users in the group. |
| **Resource Pool** | Allows this user group to be used as a resource pool for assignment in Product Portfolio Management objects |
| **Define as Partner Group** | If the user group is to be a partner (members can create and submit package objects), select Yes. The user group will then appear in the list of valid partners when you set any user's Program Manager for Partners property. See Setting Up User Groups as Partners on page 183. |
| **Allow Escalation Designation Approval** | If the user group has identified one or more designated escalation persons (DEPs), this setting determines when the DEPs can approve or reject a routable object.<br>*Settings*:<br>After Escalation—the user group's DEP can approve a routable object only after it has been escalated.<br>Always—the user group's DEP can always approve or reject a routable object.<br>(For more information about designated escalation person status, see Assigning Escalation Persons to User Groups on page 180.) |
| **Owner** | Names the creator of the user group; for information purposes. The owner is the creator by default, but can be modified by a user. |
| **Max Number of Named Users** | Maximum number of users the user group can have.<br>*Settings:*<br>Blank (default) — The number of users is unlimited.<br>Any positive integer — Defines the maximum number of users contained in the user group. The number can exceed the number of user licenses, but it cannot be less than the number of users already assigned to the user group. |

| Property | Description |
|---|---|
| **Lists, Project Type, Region, Division, Product Line, Customer, Launch Year, Labor Rate, Overhead Rate**, and **Bill Rate** | See User Properties Defined. |

# Modifying a User Group

The following sequence applies to modifying any property of a user group.

**To modify the** User Groups **settings in Java Client:**

1. Under **User Settings,** double-click **User Groups**. The User Groups window appears.
2. Double-click a specific user group, say, Customer Support. The Customer Support window appears. The user group tabs are **General Info**, **Users**, **Escalations**, **Attachments**, **History**, and **Share**.
3. In any editable field, enter a new value or select from the dropdown list.
4. When you have finished, click **Save**.
5. If you wish to revert to the original settings or start over, click **Close**. You will be prompted whether you want to save new settings on the window; click **Yes** if you do or **No** if you do not.

> **Note** If your company uses an LDAP system with PLM, if the User Group "synch" function enabled, you cannot remove or add users on a user group's **Users** tab that have been synchronized (that is, where users have been added to a user group via LDAP). See Synchronizing Users and User Groups on page 370.

**To modify the** User Groups **settings in Web Client:**

1. Click **Tools and Settings > Address Book > User Groups**. The User Groups page appears.
2. Click a specific user group, say, Customer Support. The User Group: Customer Support page appears. The user group tabs are **General Info**, **Users**, **Escalations**, **Attachments**, **Share**, and **History**.
3. On the **General Info** tab, click **Edit**.
4. In any editable field, enter a new value or select from the dropdown list.
5. When you have finished, click **Save**.
6. If you wish to revert to the original settings or start over, click **Cancel**. You will be prompted whether you want to cancel new settings on the page; Click **OK** if you do or **Cancel** if you do not.

# Assigning Escalation Persons to User Groups

When a workflow status exceeds its escalation time period without a signoff, user groups or individual users can be notified. These entities are called designated escalation persons. The Allow Escalation Designation Approval property determines when the designated escalation person can approve or reject a routable object.

**To assign designated escalation persons to a user group in Java Client:**

1. Under **User Settings,** double-click **User Groups**. The User Groups window appears.
2. Double-click the name of the user group you want. The user group's tabbed window appears.
3. Click the **Escalations** tab to bring it forward.
4. Click the **Add** button. Use the dropdown list to select the reusable criteria for this escalation, such as All Change Orders.
5. Click next to the **Notify Users** field. In the Select Users window, select and move appropriate users (from the **Users** list) or other user groups (from the **User Groups** list) to the **Recipients** list, and Click **OK**.

   Click **Save**. The escalations are added to the **Escalations** tab.

   If you want to cancel your changes, click **Close**.

   Now, if the escalation period for the specified criteria passes without a signoff, those users who were selected receive an escalation notification email.

**To assign designated escalation persons to a user group in Web Client:**

1. Click **Tools and Settings > Address Book > User Groups**. The User Groups page appears.
2. Click the name of the user group you want. The user group's tabbed object page appears.
3. Click the **Escalations** tab to bring it forward.
4. Click **Add**. In the Add Escalations page select the Criteria from the dropdown list to select the reusable criteria for this escalation, such as All Change Orders.
5. Click the Address Book button next to the **Notify Users** field. Use the selection window to search for users or groups and drag and drop them onto the table of users for the escalation or double-click.
6. Click **Add More** or **Add** (There is also a **Cancel** button.) The escalations are listed on the **Escalations** tab.

   Now, if the escalation period for the specified criteria passes without a signoff, those users who were selected receive an escalation notification email.

# Adding User Groups as Approvers of a Change

In an Agile PLM workflow, two properties—Approvers and Observers—must be filled out for criteria-specific properties of each Review and Released status.

**To add an approver or observer in Java Client:**

1. Select and open the routable object.
2. On the **Workflow** tab, select the status you want the additional people to approve.
3. Click the **Add Approver/Observer** button .
4. The Add Approvers and Observers dialog box appears.
5. Complete the **Approvers** and **Observers** fields. Type the names of the approvers or observers to add to the list (with semicolons separating the names).

Or click the **Approvers** or **Observers** button to open the address book for each field.

From the **Users** or **Groups** tab, you may select individual usernames or user groups.

6. If you want to add an entire user group as an approver, you can either add the group itself or the members of the group. This determines whether only one member or all members of the group must sign off.
   a. Click the **Group** tab in the address book.
   b. Select one or more group names.

      (To select user names, expand a group by clicking the plus sign **+**.)

   c. To add the group name to the recipient list, click the button. The group name is added to the recipients list. Only one user in the group needs to sign off.
   d. To add the members of the group to the recipient list, click the button. The usernames of the members of the group are added to the recipient list. Each user in the group must sign off.
7. Add any comments.
8. Click **OK**.

**To remove an approver or observer in Java Client:**

1. Select and open the routable object.
2. On the **Workflow** tab, under the appropriate status, select the approvers or observers you want to remove from the list.
3. Click the **Remove Approver/Observer** button.
4. When prompted to confirm the removal action, click **Yes**.

You can remove an approver only if that approver has not yet approved or rejected the routable object.

**To add an approver or observer in Web Client:**

1. Select and open the routable object. Click the **Workflow** tab.
2. Click the **Add Reviewers** button.

   The Add Approvers or Observers dialog appears.
3. Select the status you want the additional people to approve.
4. Complete the **Approvers** and **Observers** fields. Type the names of the approvers or observers to add to the list (with semicolons separating the names).

   Or click to open the address book palette.

   You may select individual usernames or user groups. For more information about using the address book palette, see Selecting Users or User Groups to Modify a Field in Web Client.

   To add an entire user group as an approver:
   a. In the address book palette, select User Groups in the dropdown list and execute a quick search to find the user groups you want.
   b. Select a group by double-clicking it. The Group Sign-off Criteria dialog appears.

c. Select the appropriate checkbox and click **Apply**. The checkbox choices are:

Only one group member must approve

All group members must approve.

5. Add any comments.
6. When you have completed the Add Approvers and Observers dialog, click **Add**.

**Note** When you add a user group as an approver such that only one member of the group needs to signoff (*UserGroupName* - Any), escalation behavior is determined by the user group's profile settings for **Allow Escalation Designation Approval** (After Escalation or Always) and the user group's assigned escalation persons. However, when you add a user group as an approver such that all members of the group must signoff (*UserGroupName* - All), Agile PLM automatically adds each member of the group as an individual approver and, therefore, escalation behavior depends on the user profile settings and assigned escalation persons of each individual user.

**To remove an approver or observer in Web Client:**

1. Select and open the routable object.
2. Click the **Remove Reviewers** button.

   The Remove Approvers and Observers dialog appears.
3. Select the status for which you want to remove approvers or observers.
4. In the **Remaining Approvers** and **Remaining Observers** fields, click the small **x** button on each approver or observer name you want to remove.
5. Enter any comments.
6. If desired, select the **Send notification as urgent** checkbox.
7. Click **Save**.

You can remove an approver only if that approver has not yet approved or rejected the routable object.

# Setting Up User Groups as Partners

This section collects the various elements that must be set up to enable Agile partners to send packages and to have program managers manage their package submissions. A “partner” may be defined as a company that works with your company and that has Agile users who need to send packages; or, a partner could be any Agile user that represents a company to your company.

**Note** The term “program manager” refers to the change analyst who manages package objects (instances of the Packages class); here it does *not* refer to the Agile role “Program Manager.”
Also, this use of “program” is *not* involved with the Programs base class.

**To create partners and program managers to function together in Agile:**

1. Define Agile partners, which are Agile users – usually external – who belong to a user group that has been defined as a partner. Each user group object has a **Define as Partner Group**

property. When set to Yes, the user group will appear on the list of partners available under any user's **Program Manager for Partners** property.

2. Partners can be assigned the Partner role, which gives them the privileges to create and submit packages.
3. Define one or more Agile users – usually internal – as a program manager of each partner. The **Program Manager for Partners** user property lists all "Partner" user groups, which you set up in Step 1. You pick the partner groups for each program manager, that is, you can have one user be program manager for one or more of the partner groups, and another user can be program manager for the same or different partner groups.
4. Program managers for partners need to be assigned the Content Manager role, which enables them to work with Agile package objects. Again, as stated in the Note above, do not be misled by the Agile role called Program Manager – that role deals with functionality in the PE/PPM solution.

# Chapter 15

# Supplier Groups

**This chapter includes the following:**


- Overview ..... 185
- Properties ..... 185


## Overview

The **Supplier Groups** node provides a way to "bundle" approved suppliers so that users who are involved in the PCM solution can easily select multiple suppliers during the RFQ process.

Using Java Client or Web Client, the administrator can set up global supplier groups that can be leveraged by users with appropriate roles. An Agile PLM user can also define (create, edit, delete) a unique supplier group that meets his particular needs as a personal supplier group. Administrators (users who have the ability to create and modify user profiles) can also create, edit, and delete user-level supplier groups for each user.

## Properties

In Java Client, under **User Settings,** double-click **Supplier Groups** to display the main Supplier Groups window. Supplier groups are created with the following properties.

| Property | Description | Editable | Default |
|---|---|---|---|
| **Name** | Name of supplier group | Yes | Empty |
| **Description** | Description of supplier group | Yes | Empty |
| **Suppliers** | List of suppliers in the supplier group | Yes | Empty |
| **Type** | Type of supplier group: Global or Personal group.<br>**Note** If you create a supplier group in Java Client, the default is Global with no alternative; if you create a supplier group in Web Client, Personal is available. | Not after creating the supplier group | Global |
| **Enabled** | Whether the supplier group is active or inactive | Yes | Yes |

### Available actions

- **Create** – A user administrator can create a new supplier group.
- **Edit** – A user administrator can edit a supplier group. To add or remove suppliers (users who have been assigned Supplier Administrator, Supplier Manager, or Supplier Project User roles)

from the group, the user administrator can add or remove suppliers from the group.

- **Enable** and **Disable** – Supplier groups can be enabled or disabled in Edit mode (Web Client) or using the buttons at the top of the Supplier Group window (Java Client).

**To create a supplier group in Java Client:**

1. Under **User Settings,** double-click **Supplier Groups**. The Supplier Groups window appears.
2. Click the **New** button. The Create New Supplier Group dialog box opens.
3. Fill in the **Name** and **Description** fields, and select Yes or No in the **Enabled** field.
4. The **Type** field is grayed, with the value of Global entered. (As noted in the table, “Personal” is an available choice when creating a supplier group in Web Client.)
5. Click next to the **Suppliers** field, and move suppliers from the **Choices** list to the **Selected** list.
6. Click **OK**, and Click **OK** again. The new user group object is listed in the Supplier Groups window. It is available to users from this point on.

**To create a supplier group in Web Client:**

1. Click **Tools and Settings > Address Book > Supplier Groups**. The Supplier Groups page appears.
2. Click **Add**. The Add Supplier Group window appears.
3. Fill in the **Supplier Group Name** and **Supplier Group Description** fields.
4. Select the **Supplier Group Type**, either Global or Personal.
5. Click next to the **Suppliers** field. Select suppliers from the list.

   **Note** The supplier group must have at least one supplier selected.

6. Click **Add.**

**To edit a supplier group in Java Client:**

1. Under **User Settings,** double-click **Supplier Groups**. The Supplier Groups window appears.
2. Double-click the group you want to modify. The Supplier Group setup window appears.
3. On the **General Info** tab, select or accept values from the dropdown lists. Add or remove supplier users by moving them into or out of the **Selected** list.
4. When you are finished, click **Save**.

# Chapter 16

# Roles

**This chapter includes the following:**


- Overview of Roles and Privileges in Agile PLM ..... 187
- Getting Familiar with Roles ..... 188
- Default Agile PLM Roles ..... 190
- Securing and Maintaining Roles and Privilege Masks ..... 197
- Working with Roles ..... 200


This chapter explains how to use Agile PLM's preconfigured roles and how to create new ones.

# Overview of Roles and Privileges in Agile PLM

Roles and privileges govern a user's access to Agile PLM business objects and functionality.

For a PLM user to be able to perform tasks, users must be assigned to at least one role. Each user (or user group) can be assigned to multiple roles, depending on which actions they need to perform.

Agile PLM includes a wide-ranging set of preconfigured roles and base privileges. While you can always modify roles and tailor privileges as needed, you are encouraged to become familiar with the default roles in Agile PLM before constructing your own; the default roles are listed in a few pages.

## What is a Privilege Mask?

This chapter examines PLM roles – both default and customized – and the next chapter takes up privileges and privilege masks. However, it is important to understand from the start what these elements are. (The next page describes the process to view the default roles in the **Roles** node; the **Privileges** node can be expanded in the "node tree" to see a list of the default privileges.)

The following table defines *privilege*, *privilege mask*, and *role*.

| Term | Definition | Examples |
|---|---|---|
| **Privilege** | The action users are allowed to take.<br>A privilege is only a "building block": to be assigned to users, it must be combined with other components, like Criteria and a target object. | Administrator (privilege)<br>Create<br>Read<br>Modify |
| **Privilege mask** | A set of *criteria* statements that define under what specific conditions an action can be taken on a named business object.<br>Each role requires at least one privilege mask. | Administrator (privilege mask)<br>Create Designs; Create Items<br>Read Designs; Read Items<br>Modify Designs; Modify Items |

| Term | Definition | Examples |
|---|---|---|
| **Role** | A common set of privileges that can be assigned to one or more users who perform the same functions in the change control process.<br>A role is a tailored assembly of privilege masks.<br>A user can be assigned to more than one role.<br>The default roles are set up for immediate use. | Administrator (role)<br>Change Analyst<br>Compliance Manager<br>Component Engineer<br>> Each role comprises several-to-many privilege masks. |

From this table, we can make these summary statements:

- The actions that users can perform in Agile PLM – such as creating, sending, or canceling – are based on *privileges*. But privileges do not function "by themselves".
- *Privilege masks* are combinations of: privileges, reusable criteria (introduced in earlier chapter), and target object (which can be as granular as required). A privilege mask is a functional unit that acts as a filter to manage user actions. Privilege masks can be added to roles (default or custom) or can be assigned to users without being part of a role.
- *Roles* are groupings or assemblies of several-to-many privilege masks. (A role like "Enforce Field-Level Read" has but 2 or 3 privilege masks, while "Change Analysts" has dozens.)

  Roles provide a way to allocate a common set of privileges to a group of users who have common functions in the change control process.

  When you assign a role or a privilege mask to a user, it takes effect only after the user has logged out and logged back into an Agile PLM client.

This introduction to "roles *and* privileges" in Agile PLM now turns to Roles; the following chapter goes into detail about privileges.

# Getting Familiar with Roles

You can modify the default PLM Roles in the following ways:

- Change the name of an existing role.

  Exceptions are the **(Restricted)** roles, and the Administrator, My User Profile, and View Historical Report roles, whose names cannot be changed.
- Enable or disable a role on its **General Information** tab.

  Exceptions are the **(Restricted)** roles, and the Administrator, My User Profile, and View Historical Report roles, which cannot be disabled.
- Assign a user to a role or remove a user assignment from a role on the **Users** tab.

  Exceptions are the **(Restricted)** roles, and the My User Profile and View Historical Report roles, which don’t allow you to modify user assignments.
- Add or remove privilege masks for the role by using the **Privileges** tab.

  Exceptions are the **(Restricted)** roles, and the My User Profile and View Historical Report roles,

which don't allow you to add or remove privilege masks.

- On the **Privileges** tab, double-click the name of a privilege mask to display the privilege mask's tabbed object window. When a privilege mask's tabbed window is displayed, you can modify that privilege mask.

You *cannot* do the following from the Roles window:

- Change the object type of a privilege mask.
- Modify any of the Example Roles (or its privilege masks) or the Example Privileges. These are found in **Examples** > **Example Roles** and **Example Privileges**.

# Viewing the Roles Window

**To view the Roles window:**

1. Under **User Settings,** double-click **Roles**. The Roles window appears
2. You can filter roles records to narrow your search.

   For example, set **Filter By** to Description, **Match If** to Contains, and **Value** to Change, then click **Apply** to find all the roles pertaining to "change" objects. (See Filtering Data on page 9.)

   To see all roles currently in PLM, set **Match If** to Show All and click **Apply**.

   The list of roles is displayed in the table. The Roles table shows the name, description and enabled status for each role.

The buttons on the Roles window allow you to perform various role management tasks.

| Button | Action |
|---|---|
| **New** | Create a new role. See Creating a New Role on page 202. |
| **Delete** | Deletes the selected roles. A role cannot be deleted if it is already in use. |
| **Enable** | Enables the selected roles. |
| **Disable** | Disables the selected roles. |
| **Import** | Import a text file to create a new role. See Object History on page 24. |
| **Export** | Exports role data for the selected role. See Object History on page 24. |
| **Refresh** | Refreshes the table with the latest information about the list of roles. |

# Viewing a Role

**To view a specific role:**

1. In the Roles window, double-click the role you want.

   The tabbed window for that role appears. The buttons that appear at the top of the window of each role are **Save As**, **Delete**, and **Export**.

2. The role's basic properties are displayed on the **General Information** tab. These include Name, API Name, Description, and Enabled.

3. The *privilege masks* that are assembled in the role are displayed on the **Privileges** tab.

   You can double-click any row to open that privilege-mask object; on its **General Information** tab, the field Privilege (which is grayed-out because it cannot be changed) shows the basic privilege that drives that privilege mask.

4. The **Users** tab lists users who have been assigned to this role. The **User Groups** tab lists user groups that have been assigned to this role.

   On both these tabs, you can add or remove users or user groups, respectively. Note that when you "delete" a user or user group from one of these tabs, you are only removing the role assignment (the role you are "in") from that "user object."

5. The **History** tab displays the history of actions taken on this role object.

## Guidelines for Working with Roles

Follow these steps when working with roles:

1. Print a Roles and Privileges Summary report and a Privilege Mask Detail report to see the definitions currently active in your Agile PLM system. See Administrator Reports on page 20.

| | |
|---|---|
| **Caution** | The Privilege Mask Detail report can be extremely long. You may wish to generate and view the report before printing it. |

2. Read and follow the security recommendations in Securing and Maintaining Roles and Privilege Masks on page 197.
3. If necessary, modify and create roles, as described later in this chapter.
4. Assign users to appropriate roles. See Modifying a Role on page 200.

# Default Agile PLM Roles

When you double-click the **Roles** node, you see a list of your currently configured roles. The table below lists the out-of-box roles and briefly describes what each role allows the user to do.

You can assign a user as many or as few roles as he needs to perform his duties. You can use the roles provided as they are, or you can copy a provided role (using Save As) and modify the copy to create a new role, or you can create a new role from scratch.

| | |
|---|---|
| **Caution** | As much as possible, copy (using Save As) and adapt the roles and privilege masks provided with your Agile PLM installation rather than create new ones. This is especially true for roles, since the nature of their privilege mask combinations is not immediately obvious. |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **Administrator roles**<br><br>These roles apply only to users who work in the Administrator modules in Java Client (**Admin** tab) and in Web Client (**Tools and Settings > Administration**).<br><br>An important part of tailoring the Administrator roles is the AppliedTo property in the Administrator privilege;see Administrator Privilege and the AppliedTo Capability on page 229. | | |
| **Administrator** | All Classes | Perform all administrative tasks. The other, specialized Administrator roles (including Discussion, Folder, Price, Program, Quality, Resource Pool, Sourcing, and User administrators) all have less capability than this role. |
| **User Administrator** | Users, User Groups | Perform administrative tasks (create, modify, delete) as defined by the Agile administrator |
| **General-function Agile PLM roles**<br><br>These roles apply to users who work across all Agile PLM solutions. | | |
| **Approve/Reject** | Changes, Packages, QCRs, PSRs, ATOs, and Projects | Approve or reject routable objects via workflows, and read items and changes |
| **Creator can read and discover object he or she created** | All objects | The creator of a business object can always read and discover that object. This role is enabled and part of every new user's profile, by default. There is no change in behavior with existing users (that is, "legacy" users for upgrading customers).<br><br>Removing this role from a user *prevents* Read or Discovery of objects they create. |
| **Discussion Administrator** | Discussions | Create and manage discussion objects. Discussions are used primarily in Agile PPM and PCM solutions, but it is possible to use that class in other solutions. |
| **Discussion Participant**<br><br>**(Restricted) Discussion Participant** | Discussions | Manage portions of discussions; the (Restricted) Discussion Partner would generally be someone who works outside the enterprise. |
| **Enforce Field-Level Read** | (Works in conjunction with other roles) | This role (contains only Enforce Field-level Read privilege mask) is used in conjunction with other roles: it enforces the AppliedTo fields under Read privileges in all roles assigned to the user, which reduces the user's capabilities. |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **Export** | (Allows access to Export utility.) | Allows user to export objects for which they have Discover and Read privileges. (Contains only Export privilege mask.) |
| **Folder Administrator** | File Folders | Create and manage file folders. For additional information, see Attachment Privileges on page 392. |
| **Folder Manager** | File Folders | Create and manage file folders. For additional information, see Attachment Privileges on page 392. |
| **Markup for All** | File Folders<br>(Affected Item Attachments) | Modify the redline Markups or other users for File Folders or for the Affected Item attachments of change orders. User must have Read privileges for the file folder objects or change order objects. For additional information, see Redline Markup Default Roles and Privileges on page 396. |
| **Markup for Self** | File Folders<br>(Affected Item Attachments) | Create and modify the user's own redline Markups for File Folders or for the Affected Item attachments of change orders. User must have Read privileges for the file folder objects or change order objects. For additional information, see Redline Markup Default Roles and Privileges on page 396. |
| **My File Folder** | File Folders | Permits reading file folders created by the assignee. |
| **My User Profile**<br>**(Restricted) My User Profile** | Users | View and modify his own user profile properties under My Settings, and create and modify personal user groups. Assigned to every user, and is required to use the Agile PLM system.<br>**Note** The Discover Users privilege is not part of the My User Profile role by default. |
| **(Restricted) Grant** | All objects | Grant roles to users in a controlled, finite way, using Access Control List (ACL) capability. |
| **Engineering Collaboration role**<br>This role applies to users who work in the Agile EC solution. | | |
| **Design Engineer** | File Folders (Designs class objects) | Create and manage Designs class objects and Design Model structures. |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **Product Collaboration roles**<br><br>These roles apply to users who work in the Agile PC solution. | | |
| **Change Analyst** | Changes, Items, Manufacturers, Mfr. Parts, File Folders | Create items and changes, and manage the routing and release process of changes through workflows. Also create BOMs, MPNs, AMLs, and to manage file folder references |
| **Component Engineer** | Items, Manufacturers, Mfr. Parts, Mfr. Orders (MCOs) | Create manufacturer change objects and manage the routing and release process of MCOs |
| **Incorporator** | Changes, Items, Mfr Parts, Manufacturers, Prices, | Incorporate items and numerous functions with the listed base classes (this role is similar to Change Analyst, but with fewer capabilities). |
| **Item Content Manager** | Changes, Items | Create items, and create and submit changes |
| **Manufacturer Content Manager** | Items, Manufacturers, Mfr. Parts, Mfr. Orders (MCOs) | Create items, manufacturers, and manufacturer parts, and create and submit MCOs |
| **Modify Item Released** | Items | Change the Description of an item that has been released (contains only one privilege mask for this). |
| **Partner** | Packages | Create and submit package objects |
| **Product Content Read Only** | Changes, Items, Manufacturers, Manufacturer Parts | Discover, read, comment, get, print, send, and view items, changes, manufacturers, and manufacturer parts |
| **Product Quality Management roles**<br><br>These roles apply to users who work in the Agile PQM solution. | | |
| **Organization Manager** | Suppliers, Customers | Manage suppliers and customers (also applies to PCM solution) |
| **Quality Administrator** | Quality Change Requests | Manage the corrective and preventive action and audit processes |
| **Quality Analyst** | Product Service Requests | Submit quality incidents and manage their resolution |
| **Quality Analytics User** | (Quality Analytics application) | Permits PQM user to access (Run, Discover, Read) the Quality Analytics application. |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **Product Cost Management roles**<br><br>These roles apply to users who work in the Agile PCM solution. | | |
| **Organization Manager** | Suppliers, Customers | Create and manage suppliers and customers (also applies to PQM solution) |
| **Price Administrator** | Prices, Price Change Orders | Control price management activities, including PCOs |
| **Price Manager** | Prices, Price Change Orders | Manage pricing information through creation of price objects and PCOs |
| **(Restricted) Price Collaborator** | Prices, Price Change Orders | Manage pricing information through creation of price objects and PCOs, but more limited than price manager role; typically provided to supplier users |
| **RFQ Manager** | RFQs, RFQ Responses | Create RFQs and manage the RFQ process |
| **(Restricted) RFQ Responder** | RFQ Responses | Respond to RFQs; this role is generally provided to supplier users |
| **Sourcing Administrator** | Sourcing Projects, RFQs | Control sourcing activities, including the ability to view and modify all sourcing projects and RFQs |
| **Sourcing Project Manager** | Sourcing Projects | Create and manage sourcing projects |
| **(Restricted) Supplier Manager** | Suppliers | Manage supplier information, limited to suppliers' own organizations, including the ability to create users in the supplier organization |
| **Product Portfolio Management roles**<br><br>These roles apply to users who work in the Agile PPM solution. (Note: The base class Programs is changed to Projects as of Rel. 9.3.0; the PPM roles and privilege masks may still use the term "Program".) | | |
| **Executive** | Projects | Global read access to all projects that have matching categories; for example, if a project is associated with the North American region, and a user has an Executive privilege for all North American, then he can read them |
| **Portfolio Analytics User** | (Portfolio Analytics applications) | Permits PPM user to access (Run, Discover, Read) the Portfolio Analytics application. |
| **Program Administrator** | Projects | Create and manage complete projects |
| **Program Manager** | Projects | Create and manage routing and release process for projects |
| **Program Team Member** | Projects | Manage portions of projects |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **Resource Pool Administrator** | Projects | Create and manage resource pools |
| **Resource Pool Owner** | Projects | Manage resource pools |
| **Timesheet Administrator** | Projects | Perform administrative tasks on all timesheets that are created in the system |
| **Reports roles**<br><br>These roles apply to users who work with Reports. | | |
| **Report Manager** | Reports | Access and manage all reports, even those they did not create |
| **Report User** | Reports | Create new custom reports and manage reports they created |
| **View Historical Report** | Reports, Historical Report File Folders | View specific instances of previously executed reports; this is a non-editable role and is automatically assigned whenever a report instance is shared with a user |
| **Agile Content Services role**<br><br>This role applies to users who work in the ACS solution. | | |
| **Content Manager** | Transfer Orders, Packages | Create, modify, route, release and implement CTO and ATO objects, as well as package objects |
| **Product Governance & Compliance roles**<br><br>These roles apply to users who work in the Agile PG&C solution. | | |
| **Compliance Manager** | Declarations, Commodities, Specifications, Substances | Create and manage PG&C objects, run PG&C reports, and route material declarations to suppliers |
| **(Restricted) Material Provider** | Declarations | Create, modify, and complete material declarations; this role is generally provided to supplier users |
| **Agile Configuration Propagation roles**<br><br>These roles apply to users who work in the ACP solution. For more information see the current *ACP User Guide*. | | |
| **(Propagation) Administrator** | All Classes | Perform propagation tasks. |

| Role | Applies to these base classes (or other PLM utility) | Description |
|---|---|---|
| **(Propagation) User Administrator** | Users, User Groups | Perform propagation tasks that are limited to users and user groups. |

# Example Roles

Your Agile PLM installation includes both a read–write version and a read-only example or reference copy of the roles. double-click the **Example Roles** node under the **Example** node folder to see the reference copy, which is provided so that you always have default copies of the roles as they were installed. To restore the default version of any role, you can just make a copy (using Save As) of its Example role.

For example, the following are the two versions of the Change Analyst role provided at installation:

- **Change Analyst** — This is a modifiable (Read–Write) role that you can assign to users.
- **Example–Change Analyst** — This is an uneditable (Read-Only) reference copy of the Change Analyst role.

**Note** If Agile PLM users are unable to move a routable object from one status to another, for example from Hold to CCB, often they don't have appropriate privileges to make these status changes. In such cases, check the roles they are assigned and the privilege masks included in those roles.

More information about example roles is in Using the Example Roles on page 205.

# Restricted Roles

Some roles are restricted in the extent to which they can operate on their specified objects; these role names are preceded by the word Restricted in parentheses. Their purpose is the same as the like-named role that is not preceded by "(Restricted)". They are generally assigned to members of supplier organizations, that is, Agile PLM users who work outside the enterprise.

# Role-based Enhancements

## User Administrator Enhancement

The PLM role User Administrator has a privilege mask called "Admin Access for User Admin"; its AppliedTo property dictates which User-related Administrator *nodes* can be accessed (Users, User Groups, and so forth). The User Administrator can also access the User-specific functions in Web Client (accessed via **Tools and Settings > Address Book**).

The User Administrator role also has a privilege (and privilege mask) called "User Administrator". The AppliedTo property of the User Administrator privilege mask is populated with PLM *Roles*. This means that a user administrator can be configured so that he can only assign specified roles to PLM users.

This is more fully described in Options for Building Administrator Assistants on page 231, which is part of the important topic AppliedTo Capability on page 228.

### Advanced Search Security

Two restrictions have been added to Advanced Search utility.

The first restriction applies to the "Search for" field when a user creates any type of search (Quick, Advanced, Parametric search). Now he will see (in the drop-down list of classes) only those classes for which he has been assigned the Discovery privilege.

Similarly, "Field" criteria condition (drop-down list) will not display fields if user does not have a Read privilege (or Enforce Field-level Read privilege) to them. In order to ensure restricted field-level read, the user's Read privilege mask must include in its AppliedTo property only the fields you wish the user to read. In addition, the user must also be assigned an Enforce Field Level Read privilege mask, for example, by assigning the Enforce Field Level Read role.

For a user to be able to view **Workflow Routings** folder under **Search** tab (on the Navigation pane), in the user's profile > Searches field, "Workflow Routings" must be moved from Choices to Selected table.

# Securing and Maintaining Roles and Privilege Masks

| | |
|---|---|
| **Important** | This section is intended to both warn you of potential risks if you do not have a "security and maintenance plan" and provide some steps for simple troubleshooting if you do encounter a problem in PLM. Please discuss this matter with your Oracle Consulting – Agile Practice representative, or call Agile Support. |

By following the recommended plan described in this section, you will help prevent security violations, interruptions in your change control process, and confusion if you need to restore the system to a previous security configuration.

| | |
|---|---|
| **Caution** | Failure to follow the recommendations in this section could have serious consequences: without your knowledge, users may unexpectedly be able to perform actions they had been prohibited from performing; also, users could suddenly lose their ability to carry out required actions. |

When corruption or unplanned alteration to your Agile PLM system occurs, you will have to interrupt regular system management just to restore essential permissions, which itself could result in additional violations.

## Using Agile Configuration Propagation to Restore Admin Settings

You have access to Agile Configuration Propagation (ACP), which lets you propagate the configuration of one Agile instance to another instance of the same version. ("The configuration" is a general term that refers to "all settings content of all Java Client Administrator nodes in one Agile

instance".) Any single propagation may consist of the complete Administrator data for an instance, or it may consist of a selected subset of Administration data for an instance.

The subsections that follow, regarding Admin Export and running Administrator Reports, offer some steps to troubleshoot or to diagnose a problem you may suspect exists. By these means, if you detect a problem with a role or some roles or privileges in your production instance of PLM, you may be able to resolve it from within the Administrator module itself.

However, if you determine that your production instance has a serious and widespread problem in Roles and Privileges (or, indeed, with other parts of the PLM application), you may need to do a propagation using ACP to restore correct (and validated) settings. In this case, see *Agile Configuration Propagation Guide* for this release.

# Recommended Security and Maintenance Plan

You should take great care when making changes to the powerful Agile PLM roles. Please follow these configuration and maintenance recommendations:

- It is strongly recommended that one person be assigned responsibility for configuring roles and privileges. If a change in ownership is to occur, the current administrator should explain to the new owner:
  - The roles and privileges configuration
  - Any changes that have been made
  - The system of tracking changes to roles and privileges
- Perform a roles export (in the Roles window, select all roles, and click **Export**) and run the Roles and Privileges Summary report and the Privilege Mask Detail report before you change or create roles or privilege masks. Save the results for later comparison in case you need to determine what changes caused particular effects. Keep a log of changes to the privileges configuration. For more information about exporting, see Object History on page 24. For more information about reports see Administrator Reports on page 20.
- Use the supplied roles and privileges and make only required alterations.
- Follow the Action–Criteria–Object Type privilege mask naming convention (described in About Privileges and Privilege Masks on page 207) to avoid confusion and potential security breaches. Include at least these three recommended basic identifying parts, and maintain consistent word order. Use the names of the privilege masks provided at installation as your guide.
- Avoid changing role and privilege assignments after their initial assignment. For example, if a change originator or CCB member has the Discovery privilege removed after an ECO has been routed, the originator or CCB member still receives notifications but cannot view the change.
- Avoid changing role and privilege assignments while users are working on the system. If such changes are made, users must log out and restart Agile PLM clients again for some changes to take full effect.

# Database Backup Procedures

Before you make any changes, it is a good idea to back up the entire Agile PLM database, and run Administrator reports. You can use the database backup to rebuild the previous database. You can compare “before-and-after” reports and use them to diagnose the cause of changes to Agile PLM

security.

## Exporting Agile PLM Administrative Settings

Follow the instructions in Object History on page 24. When you have finished backing up current role and privilege settings, run your reports.

| | |
|---|---|
| **Note** | If you are importing these settings, be certain you are importing to the same database or the Agile PLM administrative data will not match. |

## Running Reports

When you run an Administrator report, include the report date and time in the filename, such as Roles_072601_3pm.csv. If you keep these report files, you can compare them later to track changes.

| | |
|---|---|
| **Note** | Be aware that the Privilege Masks Details report can be quite long (hundreds of pages), depending on the number of users on the system. |

**To run the Roles and Privileges Summary report:**

1. Select the **Analytics and Reports** tab in the navigation pane. The **Analytics and Reports** folders appear.
2. Expand the **Standard Reports > Administrator Reports** folder.
3. Double-click (Java Client) or click (Web Client) the Roles and Privileges Summary report. The Roles and Privileges Summary Report page appears.
4. Click **Execute**. The Get Attachment window appears.
5. Click **Continue**. This accepts the displayed default encoding type (Western European (ISO)). If you need to use a different encoding type, select it in the dropdown list and click **Continue**.
6. Follow the directions in the File Download and Save As dialog boxes to save the file to disk and specify a location to save it.
7. Modify the filename to include the report date and time. For example, **Roles072602_3pm.csv**
8. When the download is complete, Click **OK** in the Get Attachment dialog box.

**To run the Privilege Mask Detail report:**

1. Select the **Analytics and Reports** tab in the navigation pane. The **Analytics and Reports** folders appear.
2. Expand the **Standard Reports > Administrator Reports** folder.
3. Double-click (Java Client) or click (Web Client) the Privilege Mask Detail report. The Privilege Mask Detail Report page appears.
4. Click **Execute**. The Get Attachment window appears.
5. Click **Continue**. This accepts the displayed default encoding type (Western European (ISO)). If you need to use a different encoding type, select it in the dropdown list and click **Continue**.
6. Follow the directions in the File Download and Save As dialog boxes to save the file to disk and specify a location to save it.

7. Modify the filename to include the report date and time. For example, **Privmask072602_3pm.csv**
8. When the download is complete, Click **OK** in the Get Attachment dialog box.

> **Note** Be aware that the Privilege Masks Details report can be quite long (hundreds of pages), depending on the number of users on the system.

By running the Roles and Privileges Summary report and the Privilege Mask Detail report, you create a record of the current roles and privileges configuration. These reports are ASCII text files in .CSV format (comma separated values) that can be opened with an analysis application, such as Microsoft Excel.

You can also use a word processing program to compare two versions of the same report. If the changes you make to the security configuration produce unexpected results, you can run these reports again, and compare them in the word processing program to see the changes that you made.

# Working with Roles

You can modify existing roles and create new ones. Once a role exists, you can assign it to users. This section describes the following role-management tasks:

- Modifying a Role
- Creating a New Role
- Deleting a Role

## Modifying a Role

When you open a role, you can change its name, description, whether it's enabled or disabled, its list of privilege masks, and the users assigned to it. You cannot modify the properties for the Example–ReadOnly roles.

**To change the name or description of a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click a role to open it.
4. In the **Name** field, type a unique name (up to 255 characters).

> **Note** You cannot rename the Administrator, My User Profile, or View Historical Report roles, or any of the Restricted roles.

5. In the **Description** field, type a short description (up to 510 characters).
6. Click **Save**.

**To disable a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.

2. Use the filter bar to display the role you want to modify.
3. Click the **Disable** button.

> **Note** You can also disable a role by opening it and changing its **Enabled** property. You cannot disable the Administrator, My User Profile, or View Historical Report roles, or any of the Restricted roles.

**To remove privileges from a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click a role to open it.
4. Click the **Privileges** tab.
5. Select the privilege you want to remove.
6. Click **Remove** to delete the selected privilege from the role.
7. Click **Yes** on the Delete dialog.

**To add privileges to a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click a role to open it.
4. Click the **Privileges** tab.
5. Click **Add Privileges** to open the Select Privileges dialog box.
6. Select privilege masks in the **Choices** list and use the right arrow to move privilege masks to the **Selected** list.
7. When you are finished, Click **OK**.

**To remove a user from a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click a role to open it.
4. Click the **Users** tab.
5. Select the user you want to remove.
6. Click **Remove** to delete the selected user from the list of assigned users.
7. Click Yes on the Delete dialog.

**To add users to a role:**

1. Under **User Settings**, double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click a role to open it.
4. Click the **Users** tab.
5. Click **Add Users** to open the Select Users dialog box.
6. Select and use the arrows to move users from the **Names** list to the **Recipients** list.
7. When you are finished, Click **OK**.

# Building and Testing Attributes, Read/Modify Privileges, and Roles

Agile's privilege model is powerful and flexible, but it is also complex. The following task is a simple procedure for you to check that attributes (both pre-defined and user-defined attributes), privilege masks, and roles all work together as you customize them for your users' work.

**To test that an attribute works with Read or Modify privilege masks and roles:**

1. In a class, for example, Parts class, create a new, simple **Page Two** attribute. In this example, you are creating **Parts.PageTwo.Test01**. (See About Attributes, Flex Fields, and Read-Through Fields on page 70.)
2. Create test privilege masks called Read Parts and Modify Parts. Apply them to the Test01 attribute. (See AppliedTo Capability on page 228. You may want to examine Discovery and Read Privileges on page 236.)
3. Assign the Read Parts and Modify Parts privilege masks to the Content Manager role. (See Adding a Privilege Mask to a Role on page 211.)
4. Assign the Content Manager role to a user, perhaps a test user you create for the purpose. (That task precedes this section.)
5. Log out of your client, log in as that user, and check to see that you can both read and modify the Test01 field.

| | |
|---|---|
| **Note** | This procedure is an overview of building and testing attributes, privileges, and roles. The task of setting up Agile PLM for your company is quite large, and you might consider enlisting the services of an Oracle Consulting – Agile Practice consultant for this important task. |

# Creating a New Role

Before creating a new role, you should review the roles you currently have. Run the Roles and Privileges Summary report to see a listing of current roles and their privilege masks (see Running Reports on page 199).

The recommended method for creating a new role is to copy an existing role, and then make any

necessary changes to the copy (see Creating a New Role Using Save As on page 203). You can also create a new role from scratch (see Creating a New Role from Scratch on page 204).

**Note** If you delete a role, you can reuse the name of the deleted role when you create a new role or rename an existing role.

## Before You Begin

Before creating a new role, answer the following questions:

- What will you name the new role?
- Which users will be assigned to the new role?
- What do you want the users assigned to this role to be able to do in Agile PLM?
- What do you want users assigned to this role to be prevented from doing in Agile PLM?
- Can you modify one or more existing roles to achieve the results you want?
- Is there an existing role that you can copy and modify to avoid having to create the role from scratch?

## Creating a New Role Using Save As

You might find it easier to duplicate an existing role under a new name.

**Note** Creating a new role by duplicating one of the Example roles is a different process and produces different results. Review both processes to determine which one meets your needs. See Using the Example Roles on page 205.

**To create a new role from an existing role:**

1. Under **User Settings,** double-click **Roles**. The Roles window appears.
2. Use the filter bar to display the role you want to modify.
3. Double-click the role you want. The tabbed window for that role appears.
4. Click the **Save As** button. The Save As dialog box appears.
5. Enter a new name and API name for the role.
6. Click **OK**.

   The new role is created with the new name, and its tabbed window appears. The new role has the same description as the one you copied. You can modify the description now, if you want. See Modifying a Role on page 200.

   The list of privilege masks assigned to the original role is also assigned to the new role. However, no users have been assigned to the new role.

   For example, if the original role included the Modify Item privilege mask, the new role also includes the Modify Item privilege mask.

**Note** A read-only example role has read-only example privileges. When you use Save As to create a copy of an example role, the way the privilege masks are copied from the example role is different from what is described above. See Using the Example Roles on page 205 for details.

7. Do one or more of the following:
   - On the **Privileges** tab, remove the privilege masks you do not want in the role.
   - On the **Privileges** tab, add the privilege masks you want to include in the role.

   For instructions on how to add or remove privilege masks, see Modifying a Role on page 200.
8. To assign the new role to specific users, see Modifying a Role on page 200.

## Creating a New Role from Scratch

You may decide to create a role or roles from scratch.

**To create a new role from scratch:**

1. Under **User Settings,** double-click **Roles**. The Roles window appears.
2. Click the **New** button. The Create Role dialog box appears.
3. Type the name, API name and description of the new role.
4. To enable or disable the role, select Yes or No in the **Enabled** dropdown list. It is suggested that you disable the role while you develop it; select No.
5. Do one of the following:
   - To finish creating the role without assigning privilege masks or users (you can assign privilege masks and users later), click **Finish**.
   - To assign privilege masks to the role, click next to the **Privilege** field. The Select Privileges dialog box opens. Continue with step 6.
6. Select privilege masks in the **Choices** list and use the right arrow to move privilege masks to the **Selected** list.
7. When you are finished, Click **OK**.

**Note** You can also click **New** in the dialog box to create a new privilege mask. The Create Privilege dialog box opens. See Creating a New Privilege Mask from Scratch on page 234. (Start with step 3.) The new privilege mask will be added to the role, and the privilege mask will appear in the list of available privilege masks in the **Privileges** node.

8. Do one of the following:
   a. To finish creating the role without assigning users, click **Finish**. You might want to do this while developing the role and its privilege masks.
   b. To assign specific users to the new role, click next to the **Users** field. The address book opens. Continue with step 9.
9. Select users from the **Names** tab—and groups from the **Groups** tab—and use the right arrow to move them to the **Recipients** list.
10. When you are finished, Click **OK**.

11. When you are finished defining the new role, click **Finish**.

    The new role name appears in the Roles table. (Click **Refresh** if you do not see the role in the table.) Add privileges and users as needed. If you disabled the role in step 4, the **Enabled** field says No. You must enable the role for its assignment to users to go into effect.

# Using the Example Roles

Agile PLM includes a number of example roles that you can use as references or as a starting point for your own roles. They are stored under the **Example** node folder in the **Example Roles** node. Example roles are read-only and cannot be modified. However, you can use Save As to make a copy of an example role and modify the copy.

Each example role corresponds to one of the preconfigured roles provided when Agile PLM is first installed. If you have modified the preconfigured roles and privileges, you can view the example roles to compare your modifications to the original construction of the preconfigured roles and privileges.

## Example Privilege Masks in Example Roles

Example roles include example privilege masks. Example privilege masks are also read-only and cannot be modified. When you use Save As to create a copy of an example role, the system populates the new role with copies of the read-only example privileges.

For example, the Example - Creator role includes an example privilege mask named Example - CS - Submit ChgOrder. If you use Save As to make a copy of the Example - Creator role, the new role will include a privilege named Copy of Example - CS - Submit ChgOrder. This privilege mask copy is a new privilege mask that did not exist in the database before you copied the example role. All the privilege masks in the new role are copied in the same manner.

## Creating a New Role from an Example Role Using Save As

**To create a new role from an example role:**

1. Under **Examples,** double-click **Example Roles**. The Example Roles page appears.
2. You can filter roles records to narrow your search. (See Filtering Data on page 9.)
3. In the Example Roles window, double-click the example role you want. The tabbed window for that example role appears.
4. Click the **Save As** button. The Save As dialog box appears.
5. Enter a new name and API name for the role.
6. Click **OK**.

   The new role is created with the new name, and its tabbed window appears. The new role has the same description as the example role you copied. You can modify the description now, if you want. See Modifying a Role on page 200.

   The privilege masks in the new role are copies of the example privilege masks in the example role. For more information, see “Example Privilege Masks in Example Roles” above.

7. Do one or more of the following:
   - On the **Privileges** tab, remove the privilege masks you do not want in the role.

- On the **Privileges** tab, add the privilege masks you want to include in the role.

For instructions on how to add or remove privilege masks, see Modifying a Role on page 200.

You can also change the names of the privilege masks or change the criteria or applied to properties of the privilege masks.

On the **Privileges** tab, double-click a privilege mask to display its tabbed object page. Once you have displayed the privilege mask window, you can:

- Change the name or description.
- Modify the privilege mask criteria (see Modifying Privilege Mask Criteria).
- Modify the privilege mask applied to property (see Viewing and Modifying the AppliedTo Property on page 229).

8. For information about how to assign the new role to specific users, see Modifying a Role on page 200.

# Deleting a Role

If a role is no longer needed and no users have been assigned it, you can delete it.

**To delete a role:**

1. Under **User Settings,** double-click **Roles**. The Roles window appears.
2. Click the role you want to delete. The tabbed window for that role appears.
3. Click the **Users** tab to bring it forward.

> **Note** You cannot delete a role if it has been assigned to any users or user groups. Before you can delete the role, you must remove all of its users and user groups. Remember that it is always possible to disable a role if you are not ready to delete it but want to block its effect on assigned users and user groups.

4. If there are any users listed on the **Users** tab, select all the users on the table.
5. Click **Remove** to clear the **Users** tab.
6. Repeat steps 3, 4, and 5 for the **User Groups** tab.
7. Click the **Delete** button.

# Chapter 17

# Privileges and Privilege Masks

**This chapter includes the following:**


- About Privileges and Privilege Masks ..... 207
- Getting Familiar with Privileges and Privilege Masks ..... 207
- Default Agile PLM Privileges ..... 211
- Privilege Mask Components ..... 221
- Enabling a Privilege Mask ..... 227
- Modifying Privilege Masks ..... 228
- Creating New Privilege Masks ..... 233
- Combining Options in Criteria and AppliedTo Attributes ..... 235
- Discovery and Read Privileges ..... 236
- Modify Privilege ..... 242
- Applying Create User Criteria ..... 246
- Using Privilege Mask Criteria Variables ..... 247


This chapter explains how to use Agile PLM's preconfigured privileges and privilege masks, and how to create new ones.

## About Privileges and Privilege Masks

This chapter focuses on privileges and the construction of privilege masks. The actions that users can perform in Agile PLM—such as creating, sending, or canceling—are based on *privileges*. Privileges are combined with reusable criteria to create *privilege masks*, which act as filters to manage user actions. Privilege masks are then grouped into *roles* that provide a way of allocating a common set of privileges to a group of users who have common functions in the change control process.

Agile PLM provides a comprehensive set of preconfigured roles, privilege masks, and reusable criteria that are accessible from the **Roles**, **Privileges**, and **Criteria** nodes, respectively. You can change the preconfigured roles and privilege masks as necessary, or you can create new ones to fit your specific needs.

| | |
|---|---|
| **Note** | When you assign a role or a privilege mask to a user, it takes effect only after the user has logged out and logged back into an Agile PLM client. |

## Getting Familiar with Privileges and Privilege Masks

**To view the Privileges window:**

1. Under **User Settings,** expand the **Privileges** node. The **Privileges** node expands to show all the basic privileges (the action component of privilege masks) as subnodes.

2. Double-click a **Privileges** subnode. The page that opens is called **Privileges for <selected privilege>**, although no privilege masks are listed. For example, when you double-click **Administrator**, the Privileges for Administrator page opens. You must set some filter criteria; often it is useful to set the Match If field to **Show All**, then click the **Apply** button to the right. All the privilege masks that use the Administrator privilege are displayed.

   **Important** The page in this example reads "Privileges for Administrator" but makes more sense as follows: "**Privilege Masks that use Administrator privilege**".

3. You can filter the returned privilege masks to further narrow your search of privilege masks. In this example, setting a new filter with Match If set to **Contains** and Value with **User** typed in, clicking **Apply** reduces the returns to Admin Access for User Admin. (See Filtering Data on page 9.)

   You can click a column header in the privilege mask table to sort the table by that column.

# Viewing Privilege Masks

The privilege masks that make up a role define the actions its users can perform. Agile PLM provides you with a set of preconfigured privilege masks for the installed roles. Every role has at least one privilege mask.

**To view the privilege masks for a particular role:**

1. Under **User Settings,** double-click **Roles**.
2. In the Roles window, double-click the role. The role's tabbed window appears.
3. Click the **Privileges** tab. The privilege masks are displayed in the table.

## Viewing Privilege Mask Properties

Each of the privilege masks that make up a role has its own set of properties that define the privilege being controlled. The available privilege mask properties are listed and described in the following table. Not all of these properties are configurable for all privilege masks.

| Property | Description |
|---|---|
| **Name** | A name, for example, "Modify Resume Date."<br>A privilege mask name of up to 255 characters. |
| **Description** | A maximum 510-character description of the privilege mask, for example Modify Resume Date After Release. |
| **Enabled** | Yes or No. When set to No**,** allows you to work with a privilege without its being in effect. This is useful when setting up privileges. |
| **Privilege (governing action)** | A privilege, for example, Create.<br>Identifies the privilege, or action, such as Create, Delete, Modify, Print, or Read. Only one privilege is defined in a privilege mask.<br>After the privilege mask has been created, this property is not editable. |

| Property | Description |
|---|---|
| **Privilege Criteria** | The reusable criteria selected for this privilege mask. Only one reusable criteria is selected for each privilege mask. The selected reusable criteria defines two important properties of the privilege mask:<br>▫ **Object Type** — Identifies the Agile PLM object type for the privilege mask.<br>▫ **Criteria** — Specifies the exact conditions for applying the privilege, for example, Cover Page.Status Equal to StatusType.Released. The reusable criteria specifies all possible fields and match-if values for the object type. |
| **AppliedTo** | Modify, Read, Administrator, Dashboard Management, and Microsoft Project privileges only.<br>Specifies the attributes that a Modify or Read privilege mask is applied to, such as Cover Page.Date Originated. See Discovery and Read Privileges on page 236 and Modify Privilege on page 242.<br>Specifies which administrator node the Administrator privilege mask is applied to, such as Users. See AppliedTo Capability on page 228.<br>Two other privileges that use AppliedTo are described in Dashboard Management on page 36 and PPM-specific Privileges. |
| **Workflow** | (Change Status and Override privileges only)<br>For Change Status, specifies the workflow in which the user can change statuses or override change status actions.<br>For Override, the From Status and To Status properties define the change status action the user can override. |
| **From Status** | (Change Status and Override privileges only)<br>For Change Status, specifies the statuses from which the user can change to another status.<br>For Override, the From Status and To Status properties define the change status action the user can override. |
| **To Status** | (Change Status and Override privileges only)<br>For Change Status, specifies the statuses the user can change to.<br>For Override, the From Status and To Status properties define the change status action the user can override. |

**To view privilege mask properties:**

1. Under **User Settings,** double-click **Privileges**.
2. In the Privileges window, double-click the privilege mask you want to view. The privilege mask window appears.

   **Note** Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.)

3. To modify a property, enter text in text fields or use the dropdown lists.

**Note** The **Privilege (governing action)** field is not editable.

4. When you are finished, click **Save,** or click **Close** to exit without saving your modifications.

# Tasks Performed from the Privileges Window

You perform various privilege mask management task from the Privileges window by selecting one or more privilege masks, then clicking the appropriate button.

| Page menu button | Action |
|---|---|
| **Enable** | Enables the selected privilege masks. When the privilege mask is enabled, Yes appears in the **Enabled** column. |
| **Disable** | Disables the selected privilege masks. When the privilege mask is disabled, No appears in the **Enabled** column. |
| **Delete** | Deletes the selected privilege masks. |
| **New** | Opens the Create Privilege dialog box. See Creating a New Privilege Mask from Scratch on page 234. |
| **Assign Roles** | Allows you to add the privilege mask to one or more roles. **Note**: When you assign a role or a privilege mask to a user, or change the privilege masks in a role, the change takes effect for a given user only after the user has logged out and logged back in to an Agile PLM client. |
| **Export** | Exports the selected privilege masks to a text file. See Object History on page 24. |
| **Export All** | Exports all privilege masks to a text file. See Object History on page 24. |
| **Import** | Imports a text file to create one or more new privilege masks. See Object History on page 24. |
| **Refresh** | Refresh the displayed table with the latest information. |

When you double-click a privilege mask, the privilege mask's tabbed window appears.

## Removing a Privilege Mask from a Role

**To remove a privilege mask from a role on the privilege mask Where Used tab:**

1. Under **User Settings,** expand the **Privileges** node in the navigation pane. The list of basic privileges appears.
2. Double-click the basic privilege for the privilege mask you're interested in. The form for that privilege appears.
3. Click the **Refresh** button at the bottom of the form. All the privilege masks for that privilege appear.
4. Double-click the specific privilege mask you want. Its tabbed window appears.
5. Click the **Where Used** tab to bring it forward. The roles that include the privilege mask are listed in the table.

6. Select the role from which you want to delete the privilege mask, and click the **Remove** button.
7. When the confirmation dialog appears, Click **OK**. You are not deleting the role itself, you are removing the privilege mask (which is the actual object you have opened) from that role.

### Adding a Privilege Mask to a Role

**To add a privilege mask to a role on the privilege mask Where Used tab:**

1. Under **User Settings,** expand the **Privileges** node in the navigation pane. The list of basic privileges appears.
2. Double-click the basic privilege for the privilege mask you're interested in. The form for that privilege appears.
3. Click the **Refresh** button at the bottom of the form. All the privilege masks for that privilege appears.
4. Double-click the specific privilege mask you want. Its tabbed window appears.
5. Click the **Where Used** tab to bring it forward. The roles that include the privilege mask are listed in the table.
6. Click the **Add Roles** button. The Select Roles dialog box appears.
7. Move roles from the **Choices** list to the **Selected** list.
8. Click **OK** when you are finished. You are not adding a role in any way, you are adding the privilege mask (which is the actual object you have opened) to that role.

**Note** You may want to review the simple tutorial Building and Testing Attributes, Read/Modify Privileges, and Roles on page 202.

# Default Agile PLM Privileges

The following table lists and defines the privileges supplied with Agile PLM. Some privileges depend on the basic Read privilege for their effectiveness, since users must first be able to open and read objects before they can perform most other functions. This table indicates which privileges work only if the user also has a Read privilege.

For more information about the Read privilege and privilege masks, see Relationships among Discovery and Read Privileges on page 238.

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| **Add Approver/ Observer** | Add approvers/observers to the **Workflow** tab of a routable object.<br>For more information, see Add Approver/Observer and Remove Approver/Observer on page 217. | Yes |

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| **Administrator** | Log in to Java Client with access to PLM Administrator functions.<br>The Administrator role includes the Administrator privilege mask, which can be made specific to nodes based on values selected in the AppliedTo property.<br>For more information, see Administrator Privilege and the AppliedTo Capability on page 229. | No |
| **Approve/Reject** | Approve or reject a routable object that the user has been named as an Approver. | Yes |
| **Attachment Redlines for Self** | This privilege is disabled out-of-the-box. They should be enabled if the administrator wants to create Criteria specific to Workflow Statuses that, for example, prevents users from redlining once the workflow has reached Review or Released statuses.<br>**Note:** The general ability to enable and control access to attachment redline markups is provided by the Markups subclass (of File Folders class) and its associated Create, Read, Modify, Checkin, and Checkout privilege masks.<br>For more information about relining in PLM, see Redline Markup Default Roles and Privileges on page 396. | - |
| **Attachment Redlines for Others** | This privilege is disabled out-of-the-box. They should be enabled if the administrator wants to create Criteria specific to Workflow Statuses that, for example, prevents users from redlining once the workflow has reached Review or Released statuses.<br>**Note:** The general ability to enable and control access to attachment redline markups is provided by the Markups subclass (of File Folders class) and its associated Read, Modify, Checkin, and Checkout privilege masks.<br>For more information about redlining in PLM, see Redline Markup Default Roles and Privileges on page 396. | - |
| **Cancel Checkout** | Cancel a checkout of an attachment (file or URL).<br>For more information, see Attachment Privileges on page 392. | Yes |
| **Change Status** | Move a routable object from any status to any other status. The class of the criteria that is selected determines the list of available workflows to which to apply the Change Status privilege.<br>For more information, see Change Status on page 217. | Yes |
| **Checkin** | Check in an attachment (file or URL). (For more information, see Attachment Privileges on page 392.) | Yes |
| **Checkout** | Check out an attachment file (file or URL).<br>For more information, see Attachment Privileges on page 392. | Yes |

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| **Comment** | Comment on a routable object. The comment is emailed to specified users, and is recorded in History. (Not the same as comments submitted during approval or rejection of a routable object.) | Yes |
| **Create** | Create a new object in the business class specified in the privilege mask.<br><br>**Note 1:** The Create privilege is required for users of the Agile PLM Import utility.<br><br>**Note 2**: If you have activated the CreateUser attribute for an Agile PLM class by making it visible on the **Page Two** tab, the user can open and create objects that have a Read privilege. See Applying Create User Criteria on page 246.<br><br>**Note 3**: The Create privilege allows the user to fill in required fields at the time of creation even if the user does not have Modify privilege for those fields. Required fields are fields with their attribute property **Required** set to Yes; see Defining Attribute Properties on page 77. | Yes |
| **Create From Template** | Used in Create Program From Template privilege mask, which is enabled out-of-the-box for Program Manager and Program Administrator roles. | Yes |
| **Dashboard Tab View** | Used in Read Dashboard Tabs privilege mask, which is enabled out-of-the-box for Executive, Project Manager, and Project Administrator roles.<br><br>All newly created Dashboard tabs under **Systems Settings > Dashboard Management** require this privilege to permit viewing the tab in Web Client.<br><br>For more information, see Dashboard Management on page 36. | Yes |
| **Delete** | Delete an object.<br><br>**Note**: A routable object must be Pending or Unassigned. | Yes |
| **Discovery** | Learn that an object exists.<br><br>**Note 1**: You must restart the WebLogic server when you change the Enabled property of Discovery Privilege in the **Database** node.<br><br>**Note 2**: If Discovery Privilege is disabled in the **Database** node, users with a basic Read privilege can discover all objects.<br><br>For more information, see Discovery Privilege on page 237. | No |
| **Enforce Field Level Read** | Control certain performance consequences when a user does a Field-level Read query.<br><br>For more information, see Enforce Field-Level Read Privilege on page 240, including **Important** note about a workaround due to | Yes |

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| | removal of the Specify Output Columns privilege. | |
| **Export** | Extract data from selected objects and export it to either a comma-delimited text file or a PDX package. For more information about this privilege, see Export on page 218. | Yes |
| **FileLoad** | Run the FileLoad utility. See the Import and Export Guide for more information about the FileLoad utility. | *Yes* |
| **FullSearchDisplay** | See a full return of reports queries on objects. For more information, see Full Search Display on page 218. | Yes |
| **GetFile** | Get or open a file from an object's **Attachments** tab. Note that GetFile privilege works in tandem with Checkout privilege to actually deliver the attachment file to the user's machine. The GetFile privilege (without Checkout) allows the administrator to permit a user to get a file without being able to change it in the product record.<br>For more information, see Attachment Privileges on page 392. | Yes |
| **GlobalSearches** | Create, modify, or delete a search that shows up in everyone's search list. This privilege also allows user to order searches and search folders: in Java Client, there is a button with a down arrow when you click on any search folder or search; in Web Client, in the Organize Search popup, there is an extra Order button. | No |
| **Grant** | Grant roles, and therefore privileges, to users in a controlled, finite way, using Access Control List (ACL) capability. | Yes |
| **Import** | Use the Import process. See the *Import and Export Guide* for more information about the Import tool. | Yes |
| **Incorporate** | Toggle the incorporation status on the **Attachments** tab.<br>**Note:** An item can be Incorporated or Unincorporated under an Introductory rev. | Yes |
| **Manage Report** | Modify and delete report schedules and layouts, and create and delete global report folders. | Yes |
| **Microsoft Project** | Access, use, or modify Microsoft Project, and part of Project Manager and Project Administrator roles. See Microsoft Project Privilege. | Yes |
| **Modify** | Modify fields on the tab of an object. Modify privileges are assigned using the AppliedTo attribute settings.<br>For more information about various aspects of the Modify privilege, see Modify Privilege on page 242, AppliedTo Capability on page 228, and Modify Privilege and Attachments on page 393. | Yes |

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| **Override** | Override incomplete required fields and required approvers who have not signed off, and move the routable object to the next status.<br>For more information, see Override on page 219. | Yes |
| **PrintFile** | Print from the AutoVue for Agile window. The user must also have ViewFile privilege.<br>For more information, see Attachment Privileges on page 392. | Yes |
| **PrintTab** | Print from the tabs of an object. | Yes |
| **Purge Folder Version** | Purge unused file folder version from the Agile File Management vault. See Purge Folder Version Privilege on page 220. | |
| **Read** | Open an object to read all tabs.<br>For more information, see Discovery and Read Privileges on page 236.<br>The Read privilege is based on the AppliedTo property for specific attributes. See Field-Level Read Privilege and AppliedTo Capability on page 228.<br>The Read privilege is further applied on a field-by-field basis when Enforce Field-level Read is assigned to the same user.<br>For more information, see Enforce Field-Level Read Privilege on page 240.<br>**Note**: If Discovery Privilege is enabled in the **Database** node, a corresponding Discovery privilege mask is required. | Yes |
| **Remove Approver/ Observer** | Remove approvers/observers from a routable object.<br>For more information, see Add Approver/Observer and Remove Approver/Observer on page 217. | Yes |
| **Reset** | Reset the checksum in the Agile PLM database to match a referenced file. Reset privilege is found in Reset File Checksum privilege mask, which is not included in any out-of-box Agile role. See Handle File Checksum on page 350. | Yes |
| **Run Report** | Run reports, create schedules for reports, and create report layouts. Modify and delete a user's own schedules and layouts. | Yes |
| **SaveAs** | Copy an object by saving to another name. | Yes |
| **Send** | Carry out a **File \| Send** on an open object. A corresponding Create privilege mask is required. | Yes |
| **Subscribe** | Subscribe to notification emails when selected attributes of an object are modified. Note that the Database-node property Notification Enabled must also be set to Yes. | Yes |

| Privilege | Allows the user to... | Requires basic Read privilege |
|---|---|---|
| **Transfer Authority for Self** | Designates users to approve changes for a specified period of time when the designator—oneself—is an approver (but not observer) on a routable object.<br>For more information, see Transfer Authority for Self on page 220. | No |
| **Transfer Authority for Others** | Designates users to approve changes for a specified period of time when the original user—an "other"—is an approver (but not observer) on a routable object; the original approver is copied on all notifying email.<br>For more information, see Transfer Authority for Others on page 221. | No |
| **Undelete** | Undelete an object. | Yes |
| **Unincorporate** | Unincorporate an item. | Yes |
| **Update All Timesheets** | View all tasks for a selected user and change timesheet details entered by that user for each task. Overrides system privileges for viewing tasks – the user accessing the timesheet need not be a team member on the task. This privilege also allows the user to search timesheet entries by all users in the system. | Yes |
| **User Administrator** | Uses its AppliedTo property to tailor which **roles** that user can assign to other users. | Yes |
| **ViewFile** | View files in the AutoVue for Agile window.<br>For more information, see Attachment Privileges on page 392. | Yes |

# Further Discussion of Specific Privileges

The following section gives more information about selected privileges:

- More Information about Selected Privileges on page 217

Later this chapter explores how privilege masks are constructed, beginning with Privilege Mask Components on page 221. Subsequent sections provide detailed information about the Discover, Read, and Modify privileges, as well as the important AppliedTo property (which is found in Administrator and just a few other PLM Privileges).

- Discovery and Read Privileges on page 236
- Modify Privilege on page 242
- AppliedTo Capability on page 228

# More Information about Selected Privileges

This section provides detailed information about some important Agile PLM privileges. It is not a comprehensive detailing of all the privileges in the Agile PLM system. Keep in mind, the privileges do not work alone; they must be cited by a privilege mask—for example, Create Item—and then activated by inclusion in a role that includes that privilege mask.

**Note** Information about privileges that are involved in working with attachments is provided in Attachment Privileges on page 392.

## Add Approver/Observer and Remove Approver/Observer

The Add Approver/Observer privilege allows the user to designate a user, global user group, or partner to approve or reject a specific routable object. This privilege also allows the designation of any of these categories to be an observer that can see the routable object, is not required to approve or reject the routable object, but still can approve or reject if necessary.

The Remove Approver/Observer privilege allows the user to remove any of these from the list of approvers or observers on a specific routable object. For more information about adding an approver or observer, see More about Affected Items: Adding Approvers and Observers on page 123.

**Note** To successfully add or remove approvers or observers from a Review or Released status of a workflow, the Ad Hoc Approvers/Observers property for that status must be set to Yes.

## Change Status

Change Status is crucial in the approval process that advances changes through workflows. The Change Status privileges use privilege criteria to control the statuses a user can manipulate—for specified changes, move forward or backward in the workflow. The function of the privilege is to designate which statuses can be affected by a user in a given role.

**Note** Each Change Status privilege mask applies to one named status of one specified workflow. You must create Change Status privilege masks for each status of every new workflow you create and enable them, including the statuses of workflows created by SaveAs (that is, a workflow that is given another name via SaveAs, whether it is modified or not, needs its own Change Status privileges).

The status types that are available to the privilege are determined by the workflow selection for the privilege when it is created. This is also affected by the reusable criteria applied to the privilege.

There can be multiple “From” or “To” settings defined in Change Status. The administrator can designate a role to move a change from multiple statuses to a single status, for example, Pending, Submit, or Review to Hold status. Or the administrator can designate a role to move a change from a single to multiple statuses, for example, Hold to Pending, Submit, or Review status.

**To define a Change Status privilege mask:**

1. Under **User Settings,** double-click **Privileges**.

2. In the Privileges window, click the **New** button. The Create Privilege dialog box appears.
3. Fill in the **Name** and **Description** fields for the Change Status privilege mask.
4. By default, the privilege mask is enabled. If you want, select No in the **Enable** field to disable the privilege mask.
5. In the **Privilege** field, select Change Status from the dropdown list.
6. In the **Criteria** field, select a criteria from the list, and click the **OK** button.

   Or click the **New Criteria** button to define a new reusable criteria. (See Creating a New Criteria on page 119.)
7. In the **Workflow** field, select a workflow from the dropdown list.

   Additional fields, **Status–From** and **Status–To**, are displayed in the dialog box.

   **Note** If you select All in the **Workflow** field, the **Status–From** and **Status–To** fields are not available. When you select All for the workflow name, you create a Change Status privilege mask that allows a user to change from any status to any status in any workflow. This is useful when you want to test workflows.
8. In the **Status–From** dialog box, move statuses from the **Choices** list to the **Selected** list. Click **OK** when you are finished.
9. In the **Status–To** dialog box, move statuses from the **Choices** list to the **Selected** list. Click **OK** when you are finished. When you have complete all the fields in the dialog box, Click **OK**.

## Export

This privilege allows Agile users to extract data from selected objects and export it to such formats as: a comma-delimited text file, a Microsoft Excel file, a PDX package, or an "Agile XML" (aXML) file.

PDX is a standard for electronic exchange of engineering and manufacturing information across the supply chain. A PDX package is an XML file that can be viewed with Agile eXpress, a free client.

Please contact Oracle Consulting–Agile Practice for information about the eXpress viewer.

## Full Search Display

The Full Search Display *privilege* does not apply to searches. It applies to results of report queries. Users with the Full Report Display *privilege mask* will see all results of reports; users without the privilege mask will see up to the maximum number of results specified by the Maximum Query Results Displayed preference. Additionally, all privilege checking is bypassed on users with this privilege when they view report results.

## Grant

A user’s ability to grant roles to another user for a specific object is handled by the Grant privilege.

The *sharing* feature of Agile PLM lets a user “grant” one or more of his roles to another Agile PLM user or user group for specific objects. The capability to share a given role includes:

- The user’s assigned, “permanent” roles, although he shares the role with another user regarding only a specific object

- Roles that have been shared with the user (by another user) for a particular object
- Roles that have been shared with the user by virtue of belonging to a user group to which the role has been shared.

The named user (or members of the named user group) can then perform actions permitted by the roles for that object only; the user does not acquire the role in any permanent or far-reaching way.

A user's or user group's **Share** tab lists those objects for which the user has been granted "shared roles" by a different user. A user can click **User Profile > Share** to see objects being shared with them and what role(s) inform their interaction with each. A user can click **User Profile > User Groups > (specific group) > Share** to see objects being shared with you via user group.

## Override

The Override privilege governs who can override the required fields and required approvers for a status and move the change to the next status without those fields being filled in or approvers signed off. With Override privilege, a user can release a change even when all required fields are not filled in.

The Override privilege is specific to each status. A change can be moved to the next status without all approvals as long as the user has the privilege to move a change to the next status. For example, a user might be able to override the promotion of a change from the Manager Review status to the CCB Review status, but not the promotion of a change from Final Review to Released.

A warning is issued to the user if there are missing fields, approvers, or other requirements for the usual move to the next status level.

| **Caution** | A user with the Override privilege can promote a routable object to the next status in all of the following situations: |
|---|---|

- Regardless of the presence of any required fields.
- When Pass Release Audit is set to Yes.
- When the Change Status Approver Rejected Change SmartRule is set to Warning or Disallow.
- When the Change Status Observer Rejected Change SmartRule is set to Warning or Disallow.
- When the Change Status No Response Change SmartRule is set to Warning or Disallow.

For the Change Status Approver Rejected Change and Change Status Observer Rejected Change SmartRules, a Warning setting does not prevent the routable object from being autopromoted; that is, the SmartRules *must* be set to Disallow to prevent autopromotion.

To allow the change analyst to decide whether or not to promote a routable object that has been rejected, the following conditions need to be true:

- AutoPromotion is set to Yes for that status.
- These two SmartRules are set to Disallow.
- The change analyst has the Override privilege for this status.

The Override privilege calls for specific "From" and "To" settings, such as Change Status. The Override privilege lets a user change the status of a routable object, even if the Pass Release Audit

is set to Yes. The only exception is during the initial release of a routable object. If the Pass Release Audit is set to Warning, then the warning appears even during an override.

Note that when the user does not have Override privilege, and Pass Release Audit = Warning or No, the Status Audit should give an Error (which requires user to fix conditions) instead of a Warning.

## Purge Folder Version Privilege

The Purge Folder Version (PFV) privilege allows a user to purge unused file folder versions from the Agile File Management fault. The file folder version number is removed from the file folder **Version** dropdown list. You can create PFV privilege masks for specific subclasses of folders by selecting the appropriate folder object criteria when you create and define the PFV privilege mask. The user must have appropriate privilege masks to discover and read the file folder object in order to select the file folder version to purge. If a file folder version is checked out, has redlines, or is in use on a business object **Attachments** tab or **Content** tab, it cannot be purged.

The **PFV** command appears on the file folder object **More...** menu (Java Client) and **Actions** menu (Web Client).

## Transfer Authority for Self

The Transfer Authority for Self privilege allows the user to designate a target user, who is allowed to approve or reject changes for her for a specified period of time. The authority transfer applies when the user is an approver on a routable object, but not when the user is an observer. Transfer Authority for Self is very useful for situations where the original user is going on a vacation or extended leave.

The target user can vary depending on criteria within the privilege. For example, all changes for Project A can be directed to Mary and all changes for Project B can go to Joe. If the criteria for the different transfers overlap, then both users who have been defined as the target of the transfer receive the notifications and are able to sign off for the original user.

The transfer of authority is in effect for the duration specified by the start and end dates. Once this time has elapsed, signoff authority is automatically transferred back to the original user.

If a user's authority needs to be transferred to more than one person, such as multiple targets during a longer vacation, the target users should be specified sequentially, as shown in the following table.

| From User | To User | Start Date | End Date |
|---|---|---|---|
| Brian Henson | David Goelz | 14-July-2007 | 30-July-2007 |
| Brian Henson | Kevin Clash | 31-July-2001 | 12-Aug-2007 |

**Caution** If you need to remove a user from the system who is named as the "To" user in signoff authority transfers, see Deleting and Undeleting Users on page 250 for more information.

### Transfer Authority for Others

The Transfer Authority for Others privilege allows an authorized user to transfer an original approver's right to approve or reject changes to another user for a specified period of time. The authority transfer applies when the original user is an approver on a routable object, but not an observer.

This privilege overrides transfers designated by users who have the Transfer Authority for Self privilege. The original approver is copied on all notifying email.

**Note** When you create a transfer criteria, you can include criteria conditions with **Affected Items** tab fields. To select a specific routable object, an **Affected Items** field condition must be true for all the objects on the **Affected Items** tab of that routable object.

For example, if you specify Affected Items.Old Lifecycle Phase Equals Preliminary, then all the objects on the **Affected Items** tab must have the **Old Lifecycle** field equal to Preliminary. If you select Contains as the search operator, then every object on the **Affected Items** tab must contain the specified value in the specified field.

### Site Change Order Save As Limitations

Site Change Orders (SCOs) affect only site-specific information, and other types of changes are not limited to site-specific information (for example, ECOs and MCOs), SCOs can be Save-As-created only from another SCO. You cannot initiate Save As from a non-Site Change Order change object to create an SCO object, nor can you initiate Save As from an SCO object to create a non-Site Change Order change object. The Site Change Order's site-specific only usage makes it incompatible with other change types when using the Save As feature.

### Subscribe

**The Subscribe privilege allows a user to** *subscribe* to an object to receive notification of events that happen to that object. Users can even be notified when specified fields are modified, and can select the events about which to be notified.

The user needs the Subscribe privilege for a specific object class to be able to subscribe to that class's objects.

On the **Subscriptions** tab of a user, you can view the objects to which this user has subscribed.

Subscription events trigger two types of notifications: Email and Inbox. Email notifications are sent only if the user's Receive Email Notification preference is set to Yes; this property is on the **Preferences** tab of the user object.

# Privilege Mask Components

Privilege masks are a powerful and complex feature. Each privilege mask is composed of several components. Together, the privilege mask components define the following:

- What action the user can perform;
- The object type on which the action can be performed;

- Filtering conditions that define a subset of objects on which the action can be performed.
- AppliedTo attributes – found only in privilege masks based on the Administrator, Modify, Read, Microsoft Project, and Dashboard Management privileges – which displays specific object attributes;
- For the Change Status privilege, the workflow, the status the user can change from, and the status the user can change to (see the second diagram in Combining Options in Criteria and AppliedTo Attributes on page 235).

# All Privilege Masks have a Privilege, an Object Type, and a Criteria

Privilege masks act as filters to either enable or prevent user actions. By combining the types of action with the precise conditions under which it should be enabled, privilege masks control user activity in Agile PLM. The following components are used to construct privilege masks and are among the privilege mask properties:

1. The **privilege** or action (see Privilege Mask Component 1: Privileges on page 224)
2. The **object type** to be acted upon (see Privilege Mask Component 2: Object Type on page 225)
3. The **criteria** or conditions under which to apply the action (see Privilege Mask Component 3: Criteria on page 226)

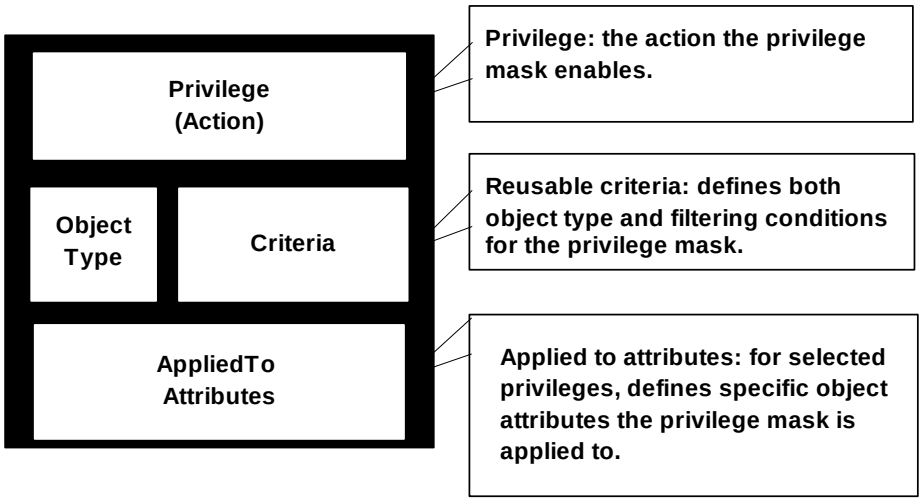
4. The **attributes** to which the action will be applied to (see Privilege Mask Component 4: Attributes on page 226)

## A Few Privileges are the Basis to Privilege Masks with AppliedTo

The final item in the above list refers to attributes in privilege masks; but this element is not found in all privilege masks. A small subset of the PLM privileges – Administrator, User Administrator, Read, Modify, Microsoft Project, and Dashboard Tab View – are the basis of privilege masks that require a fourth component, that is, named attributes found in the AppliedTo property of the privilege. This important property is detailed in AppliedTo Property in Select Privileges on page 228.

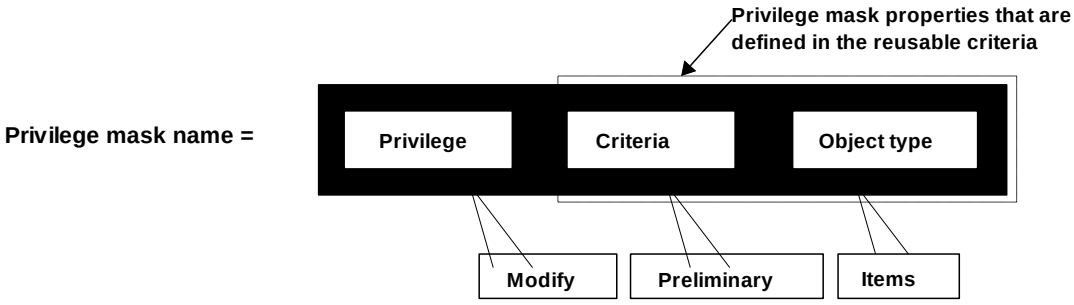
## Diagramming Privilege Masks

Privilege masks employ a reusable criteria to define the object type and filtering conditions. When you create a privilege mask, you select a reusable criteria from the list of criteria objects, as seen in the schematic diagram.




## Naming Privilege Masks

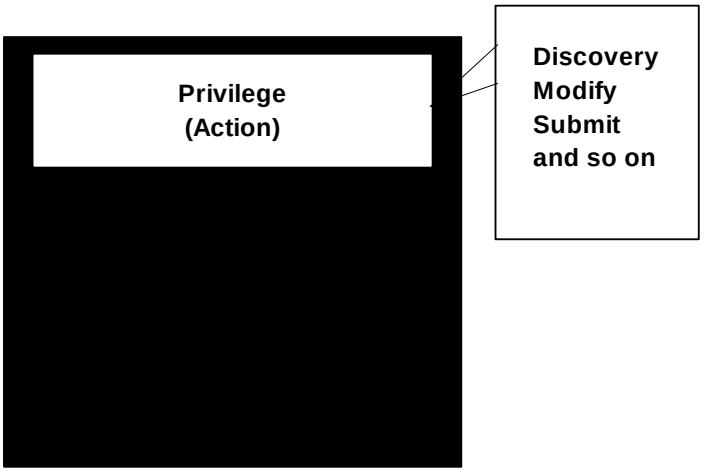
The following convention for naming privilege masks has been used so their purpose and function is as clear as possible.




Some privilege masks consist of only the privilege and the object type, for example, Send Changes.

# Privilege Mask Component 1: Privileges

The fundamental building block of a privilege mask is the *privilege*, or the action it enables, such as Discovery, Modify, or Submit. We have already looked at the basic privileges in the Agile PLM system in Agile PLM Privileges.




This basic privilege is defined when the privilege mask is created, and can be viewed on the **General Information** tab of the privilege mask. For example:

**To view the properties of the Modify Eng Changes privilege mask:**

1. Under **User Settings,** double-click **Privileges**.
2. In the Privileges window (which actually lists Privilege Masks based on the named Privilege), double-click the privilege mask **Modify Eng Changes**. The privilege mask object appears.

   **Note** Filter privilege records by Privilege Contains Modify to find the all the Modify privilege masks on the list. (See Filtering Data on page 9.)

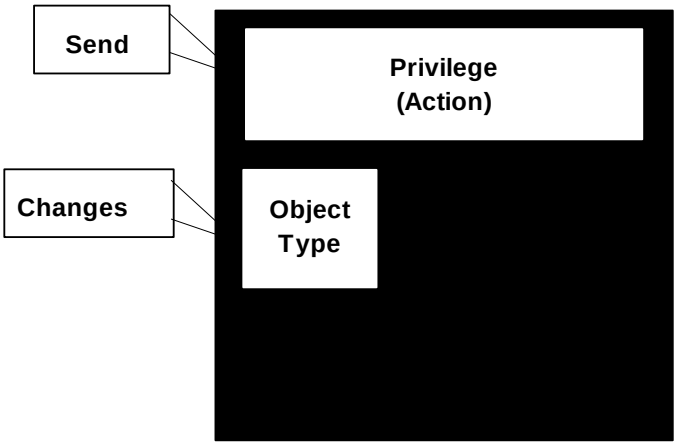
3. You can now modify the editable fields of the **General Information** tab.

   Notice that you cannot edit the **Privilege** field, indicating that you cannot change this value after the privilege mask is created. Notice, also, that you can edit the **Privilege Criteria** field.

4. Click **Cancel** when you have finished viewing the editable properties of the privilege mask.

# Privilege Mask Component 2: Object Type

Most privilege masks consist of the *object type* to be acted upon, for example, Changes base class, Change Orders class, or ECO subclass. The object type is also a privilege mask property that is defined when the privilege mask is created, when you select the reusable criteria. The object types are defined in the **Classes** node. For more information about object types, see Installed Object Types on page 52.




Some privilege masks, such as Send Changes (depicted above), consist of only the privilege and the object type. For this type of privilege mask, select a reusable criteria that specifies the object type only with no filtering criteria, such as the For Changes Only reusable criteria.

You can use the object type to broaden or narrow the privilege. The following example illustrates how a Modify Changes privilege mask could be broadened to include all changes, or narrowed to include just ECOs. See Installed Agile PLM Classes, Base Classes, and Subclasses for a complete list of Agile PLM object types.

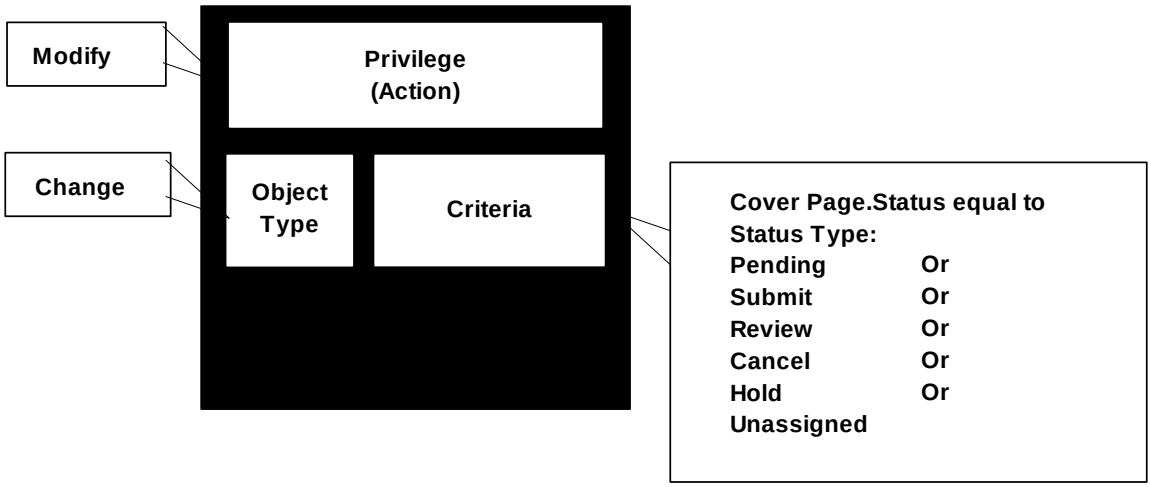
| **Select this object type** | **To allow user to modify...** | |
|---|---|---|
| | *These classes:* | *These subclasses:* |
| **Changes** | Change Orders class | ECO |
| | Change Requests class | ECR |
| | Deviations class | Deviation |
| | Manufacturer Orders class | MCO |
| | Price Change Orders class | PCO |
| | Site Change Orders class | SCO |
| | Stop Ships class | Stop Ship |
| **Change Orders class** | Change Orders class | ECO<br>(also includes any other Change Order subclasses that you have defined) |

| Select this object type | To allow user to modify... | |
|---|---|---|
| | *These classes:* | *These subclasses:* |
| **ECO** | N/A | ECO<br>(includes this subclass only) |

**Note** If you remember to use the Action–Criteria–Object Type naming convention, the object type is part of the privilege mask name. (See Naming Privilege Masks on page 223.)

# Privilege Mask Component 3: Criteria

Another property of a privilege mask is the set of *criteria* you want to apply, or the conditions under which the privilege mask will work. Criteria are defined in the reusable criteria when it is created. The criteria for the privilege mask is defined when the privilege mask is created and the reusable criteria is selected from the list. You can select a different reusable criteria later.



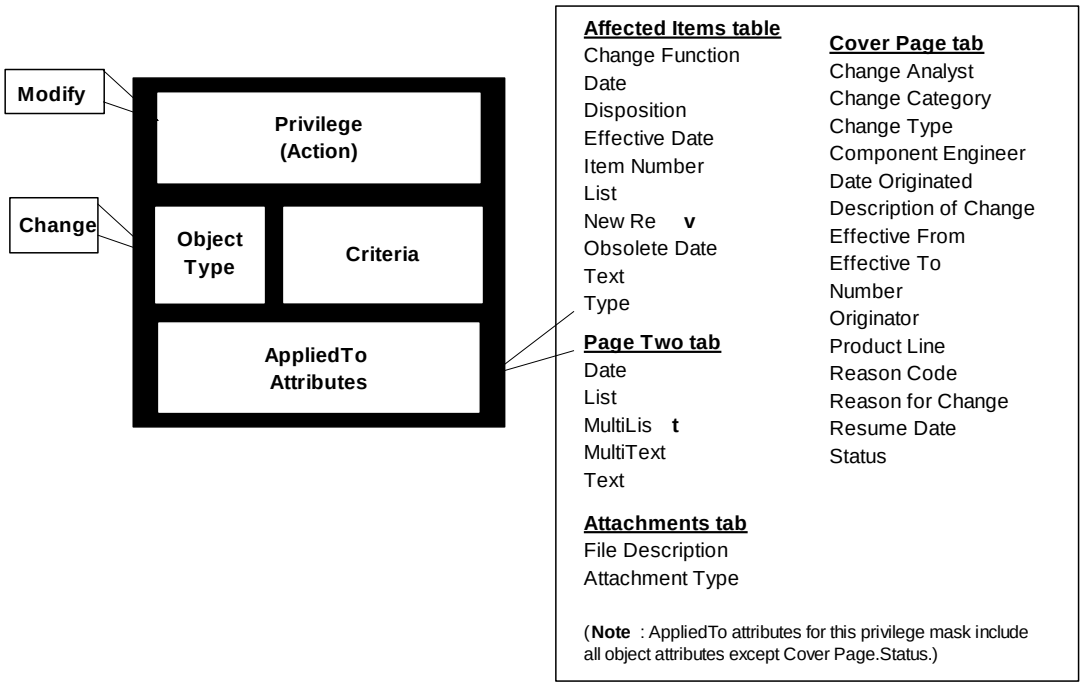

Using this particular reusable criteria allows users to modify only engineering changes in one of the status types defined in the reusable criteria (Pending, Submit, Review, Cancel, Hold, or Unassigned).

# Privilege Mask Component 4: Attributes

Another building block of a privilege mask is the *attribute* (or attributes) you want the privilege mask applied to. These are called AppliedTo attributes, and are used only in Modify and Read privilege masks. AppliedTo attributes are defined when the privilege mask is created and can also be changed later.

Modify privilege masks are designed to let users modify the attributes on a particular tab, such as the **Cover Page**, **Title Block**, or **General Information** tab.

For example, the Modify Eng Changes privilege mask provided with your Agile PLM installation is designed to apply to, or allow users to modify, the attributes shown in the figure below:




**To view these AppliedTo attributes:**

1. Under **User Settings,** double-click **Privileges**.
2. In the Privileges window, double-click the privilege mask Modify Eng Changes. The privilege mask appears; see the schematic above.

   **Note** Filter privilege records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.)

   The AppliedTo attributes are listed on the **General Information** tab.
3. To view the lists of available attribute values and selected attribute values, click the down-arrow next to the **AppliedTo** field. The **Choices** and **Selected** lists appear in the selection dialog box.
4. Click **Cancel** when you have finished viewing the lists. Click **Close** when you have finished viewing the **General Information** tab fields.

# Enabling a Privilege Mask

**To enable a privilege mask that is currently disabled:**

1. Under **User Settings,** double-click **Privileges**. The Privileges window appears.
2. Click the privilege mask you want to enable.

| Note | Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.) Or you can click the Enabled column header to sort by Yes (enabled) or No (disabled). |
|---|---|

3. Click the **Enable** button on the window toolbar.

   The disabled privilege mask is now enabled.

| Note | You can also enable and disable a privilege mask when you edit the **General Information** tab of the privilege mask. |
|---|---|

# Modifying Privilege Masks

You can modify privilege masks in many ways. This section describes these modifications.

## Modifying Privilege Mask Criteria

**To modify existing Privilege Mask Criteria:**

1. Under **User Settings,** double-click **Privileges**.
2. In the Privileges window, click the privilege mask you want to modify. The privilege mask window appears.

   | Note | Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.) |
   |---|---|

   The privilege mask properties are displayed on the **General Information** tab.
3. To modify a property, enter text in text fields or use the dropdown lists.
4. In the **Privilege Criteria** field, select a different reusable criteria from the dropdown list.

   Or click **Create** to define a new reusable criteria. (See Creating a New Criteria on page 119.)
5. When you are finished, click **Save**. To cancel the changes, click **Close**.

## AppliedTo Property in Select Privileges

The AppliedTo capability is set in privilege masks based on the following privileges.

Remember, in a privilege object, the rows are its constituent privilege masks: double-click a row to open the privilege mask object (which has AppliedTo property if it is based on these privileges):

- **Administrator** privilege: Administrator and "Admin Access for User Admin"; see Administrator Privilege and the AppliedTo Capability on page 229.
- **User Administrator** privilege: User Administrator privilege mask sets roles that user can assign to other users; Specialized Administrator Privilege Masks on page 230.
- **Read** privilege: see Field-Level Read Privilege on page 239 and Enforce Field-Level Read Privilege on page 240.

- **Modify** privilege: see Modify Privilege on page 242.
- **Microsoft Project** privilege, a PPM-specific privilege: see Project Summary Page Configuration on page 233.
- **Dashboard Tab View** privilege, a PPM-specific privilege: see Dashboard Management on page 36.

Although the following discussion centers on the AppliedTo capability as it works with the Administrator-specific privilege masks, the information in this section is pertinent to the privilege masks based on Read, Modify, Microsoft Project, and Dashboard Tab View privileges.

## Administrator Privilege and the AppliedTo Capability

The Administrator role is built on the "Administrator" privilege mask, which provides access to Administrator nodes in Java Client, and the "Admin Access for User Admin" privilege mask, which provides access to Administrator nodes in Web Client.

The User Administrator role includes the "Admin Access for User Admin" privilege mask and the "User Administrator" privilege mask, which defines roles that can be assigned to other users (more on this in Options for Building Administrator Assistants on page 231).

If an Agile administrator or assisting “user administrator” does not have a particular node properly selected in the AppliedTo property of their “Administrator” privilege mask or "Admin Access for User Admin" privilege mask, that node will *not* be a *live link* in that user’s view of the Administrator nodes, so that user will not be able to access the capability.

### *Show Visible Attributes Only Checkbox*

In the task below, you will view the **General Info** tab of a privilege mask that contains the AppliedTo property. When you click the down-arrow of the AppliedTo property, the **Show visible attributes only** checkbox is displayed. It is checked by default: uncheck it if you want to view attributes whose Visible property is set to No; all non-visible attributes are then displayed in the Choices list.

### *Viewing the AppliedTo Property*

This task can be applied to any of the privilege masks that contain the AppliedTo property.

**To view the AppliedTo property of the Administrator privilege:**

1. Under **User Settings** node folder, expand **Privileges** node.
2. Double-click **Administrator** privilege. The Privileges for Administrator filter dialog appears.
3. In Match If field dropdown list, select **Show All**. Click the **Apply** button.
4. Available Administrator-related privilege masks appear. Double-click **Administrator** privilege mask.
5. On the Administrator privilege mask’s **General Info** tab, click the down-arrow of AppliedTo property.
6. The **Choices** list displays all Administrator nodes or other Admin utilities (for example, Global Replace) that are not currently available (visible) to any user whose roles and privileges include this privilege mask. The **Selected** list displays all nodes that are visible. (The **Show visible attributes only** checkbox will be checked; it does not really affect matters in the Administrator

privileges.)

**Important** The **Choices** list reflects the nature of the privilege mask. "Admin nodes and utilities" are selected in Administrator-, Read-, and Modify-based privilege masks. Microsoft Project and Dashboard Tab View-based privilege masks offer a different array of choices; see Dashboard Management on page 36 and Project Summary Page Configuration on page 233.

### *Modifying the AppliedTo Property*

**To select the AppliedTo attributes for a qualified privilege mask:**

1. Under **User Settings,** double-click **Privileges**. The Privileges window appears.
2. Double-click the qualified privilege mask you want to modify. The privilege mask window appears.

   **Note** Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.)

   The privilege mask properties are displayed on the **General Information** tab.
3. On the **General Information** tab, click the down-arrow next to the **AppliedTo** field. The selection dialog box opens.
4. Use right arrow to move selected values from the **Choices** list to the **Selected** list.
5. When you are finished, Click **OK**.
6. To finish editing the privilege mask, click **Save**.

   **Note** The AppliedTo selection dialog box lets you select from all available fields on a tab. However, users may not be able to modify all the fields you select under all conditions.

For example, when creating the Modify Released Eng. Changes privilege mask, you can select from all fields on the **Affected Items** tab, including **Item Number, New Rev,** and **Type.** However, these fields cannot be modified once the routable object is released. Because *you* can select these attributes from the **Choices** list in the AppliedTo selection dialog box, it does not necessarily follow that users can modify those fields under all conditions.

## Specialized Administrator Privilege Masks

It is possible to create specialized **Administrator** privilege masks. Particularly with this privilege mask, it is safer to alter the individual case, as it is assigned to one or several users, than to change the out-of-box privilege.

**Caution** Be extremely careful about modifying the AppliedTo properties of Administrator privilege masks that are already in use. For example, if you were to modify the AppliedTo property of the Administrator privilege mask so that it no longer included Privileges, you and other administrators would no longer have access to the **Privileges** node, and it would be very difficult to modify the Administrator privilege mask to apply to privileges again. (This kind of scenario is where **Example Roles** could be needed.)

If you need multiple administrator privileges and roles, test the new roles and privileges carefully before you modify or disable any existing administrator roles and privileges.

**Note** Carefully consider the impact on your company of having multiple administrative users, each with specific tasks. This allows you to divide administrative tasks among a larger group of people, restricting each user to a specific type of administrative task. Be sure to read Options for Building Administrator Assistants on page 231.

Here are specific caveats for removing nodes from the AppliedTo property of a user's **Administrator** privilege mask, regarding access to Admin nodes in Java Client.

- If **Criteria** is removed, there will be no Criteria link from the privilege object's **General Information** tab, the **Workflow** node, the workflow object's **General Information** tab, or the subscriber object's **General Information** tab. Also, the **New Criteria** buttons in Create Workflow and Create Privilege dialog boxes will be disabled.
- If **Role** is removed, there will be no Role link from the **Users** or **Deleted Users** nodes.
- If **Workflow** is removed, there will be no Workflow link from the Criteria Where Used table.
- If **Privilege** is removed, there will be no Privileges link from the Criteria Where Used table.

### *Options for Building Administrator Assistants*

The User Administrator role operates with a "User Administrator" privilege and privilege mask. The User Administrator privilege mask performs a different function than "Admin Access for User Admin."

You, the main Agile PLM administrator, have decided that you want the assistance of a few select users in your administrative tasks. (How many and what kind of assisting administrators depends on the size of your company, the number of PLM solutions purchased, and so forth.) Here are your options:

- Assign and refine the Administrator role to give a user access to administrative functions in Java Client. In each user object (to whom you are assigning Administrator privilege mask), use the AppliedTo property to tailor which nodes that user can work in.
- Assign and refine the User Administrator role to give a user access to administrative functions in Web Client. The User Administrator performs administrative tasks via Web Client > **Tools and Settings** > **Address Book** and **Tools and Settings** > **Administration**.
  - The User Administrator role uses the privilege mask called Admin Access for User Admin, with a reduced AppliedTo list of nodes. In each user object (to whom you are assigning Admin Access for User Admin privilege mask), use the AppliedTo property to tailor which Web Client nodes that user can work in.
  - The User Administrator role also uses the privilege mask called User Administrator. Using the same sequence of steps (in "Viewing and Modifying AppliedTo Property" above), you see that the **Choices** list is populated not with names of Admin nodes but with a complete list of PLM roles. In each user object (to whom you are assigning User Administrator privilege mask), use the AppliedTo property to tailor which roles that user can assign to other users.

    A role that is moved over to the **Selected** list permits that user to assign that role to another user.

### *Save Import Preference Setting (AppliedTo Choice)*

"Save Import Preference Setting" is a new Choice in the Administrator-specific AppliedTo property (found in **Administrator** and **Admin Access for User Admin** privilege masks). This setting (when moved to Selected) enables you or an assistant administrator to set default preferences in the PLM Import utility (**Web Client > Tools > Import**) that users of Import will see when they use the utility.

Of course, any user can reset preferences in Import for their own use, but these will not be saved. However, it is possible to assign the **Admin Access for User Admin** privilege mask to any user and select this one Choice in AppliedTo: this enables that user to have no other "User Administrator" capability other than saving preferences in Import.

### *Manage Report Template (AppliedTo Choice)*

"Manage Report Templates" is a choice in the Administrator-specific AppliedTo property (that is, found in **Administrator** and **Admin Access for User Admin** privilege masks).

When it is moved to the Selected table, Manage Report Templates enables you or an assistant administrator to upload templates for reports in the PLM Report Template utility. Choose **Web Client > Tools and Settings > Administration > Report Template** to bring up the Manage Report Template page.

Users see these templates, per assigned privileges, when they use the Reports function in Web Client.

## Dashboard Management

From the **Dashboard Management** node, you can configure the information display and behavior of the Agile PLM Dashboard. The Dashboard is highly configurable and can be used to present key project information to the user, according to preferences.

As an administrator, you can add an unlimited number of system-level tabs to the dashboard. The visibility of each tab can be controlled through roles and privileges.

You can configure a dashboard tab to generate and display content from any Agile object. Custom dashboard extensions (DXs) can be used to retrieve and present data from external systems (for example, ERP inventory numbers) into the dashboard. Tables within a dashboard tab can be configured to display data retrieved through advanced search queries or process extensions in various graphical formats. You can even configure a dashboard table to display an internal or external Web site.

Within Dashboard Management, you can:

- View and edit all dashboard tabs
- Add or delete optional dashboard tabs
- Add tables to optional dashboard tabs
- Configure multiple sources of data for the tables, such as:
  - Advanced search queries
  - Dashboard extensions (chart and table)
  - Custom content through URL process extensions (PXs)
- Configure the display of dashboard tabs

### Project Summary Page Configuration

The Administrator privilege mask with Project Summary Page Configuration specified as the Applied to property is used configure the widgets for the Summary page of a project. Widget and page configuration for the Project Summary page can be done using the **Tools and Settings > Administration** menu in Agile Web Client.

# Creating New Privilege Masks

As with roles, it is easy to create new privilege masks, although you should use extreme caution whenever you change user access and permissions. Before creating a new privilege mask, you should review the privilege masks you currently have. Generate the Privilege Mask Detail report from the **Administrator Reports** node for a listing of current privilege masks and their criteria (see Administrator Reports on page 20).

We recommend that you modify (or copy and modify) an existing privilege mask that is similar to the one you need instead of creating a new one. The existing privilege masks have been tested extensively; their operation and interaction in Agile PLM are documented.

## Before You Begin

Consider the following details before creating a new privilege mask.

- Does the privilege mask you want already exist? If so, you can assign it to the role without creating a new privilege mask. If the appropriate reusable criteria exists, check its **Where Used** tab for the privilege mask you want.
- What action do you want the privilege mask to enable in Agile PLM?
- What action do you want the privilege mask to prevent in Agile PLM?
- Upon what object type will the privilege mask act?
- What will you name the new privilege mask? Remember to use the *Action–Criteria–Object Type* naming convention.
- Which role will require the new privilege mask?
- If you are creating a basic Read or Modify privilege mask, do you also need to create a corresponding Discover privilege mask? Does a corresponding Discover privilege mask already exist? (A Discover privilege mask is **not** automatically created when you create a new Read or Modify privilege mask.)
- If it is not a Read privilege mask, does its effectiveness depend on a Read privilege?
- Under what specific conditions do you want to enable this privilege mask? Is there an existing reusable criteria that you can select when you create the privilege mask? If so, check the reusable criteria's **Where Used** tab to see if the privilege mask you want already exists.
- Will the privilege mask conflict with any existing masks in the role?
- Does the role have an existing privilege mask you can modify to achieve the results you want?
- Is there an existing privilege mask, either in this role or another role, that you can copy (using

Save As) and modify to avoid having to create one from scratch?

You can create a new privilege mask by using the Save As process as in the following procedure. You can also create a new privilege mask from scratch.

# Copying a Privilege Mask Using Save As

Use the **Save As** button in the privilege mask window to make a copy of an existing privilege mask. You can change the copy to suit your needs.

**To copy a privilege mask:**

1. Under **User Settings,** double-click **Privileges**. The Privileges window appears.
2. Double-click the privilege mask you want to copy. The privilege mask window appears.

   **Note** Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Modify to find all the Modify privilege masks on the list. (See Filtering Data on page 9.)

3. Click the **Save As** button.
4. Enter a name and API name for the new privilege mask.
5. Click **Save** when you are finished. The new privilege mask appears.

# Creating a New Privilege Mask from Scratch

**To create a new privilege mask:**

1. Under **User Settings,** double-click **Privileges**. The Privileges window appears.
2. Click **New** on the toolbar. The Create Privilege dialog box appears.
3. Enter a name and an API name. Remember to use the Action–Criteria–Object Type naming convention. (See Naming Privilege Masks on page 223.)
4. Enter a description.
5. By default, the privilege mask is enabled. If you want, select No in the **Enabled** field to disable the privilege mask. You may want to do this while developing the privilege mask.
6. Select a privilege from the **Privilege** dropdown list.

   Depending on which privilege you select, more fields may appear for you to complete.

   **Note** If you selected the Change Status privilege, see Change Status on page 217 for detailed information about completing a Change Status privilege mask.

7. Do one of the following:
   - To finish creating a privilege mask that does not require a privilege criteria (reusable criteria), click **Finish**.
8. For all other privileges, in the **Privilege Criteria** field, select a reusable criteria from the dropdown list. The reusable criteria defines both the object type and the filtering conditions (criteria) for the privilege mask.

   Or click the **New** button in the dialog box to define a new reusable criteria. See Creating a New

Criteria on page 119.

(See Privilege Mask Component 2: Object Type on page 225 for more information about selecting an object type for your privilege mask. See Criteria on page 117 for more information about reusable criteria.)

9. If you are creating a Read or Modify or Administrator privilege mask, you can select the object fields for which the Read or Modify privilege is applied. (See Privilege Mask Component 4: Attributes on page 226.)
   - Click at the **AppliedTo** field. The selection dialog box opens.
   - Use the right and left arrows to move selected values from one list to the other.
   - When you are finished, Click **OK**.
10. If you are creating a Change Status or Override privilege, see Change Status on page 217.
11. When you have completed the fields on the Create Privilege dialog box, Click **OK**.

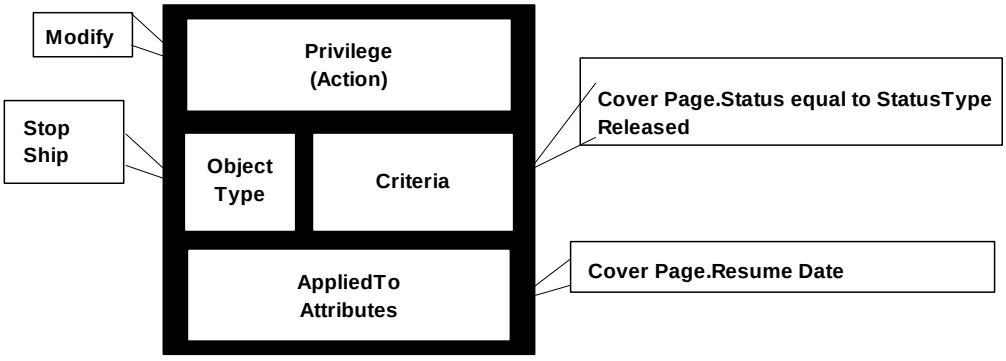
| **Note** | If you disabled the privilege mask while you were creating it, remember to enable it so it can be used. |
|---|---|

# Combining Options in Criteria and AppliedTo Attributes

When you create or modify a privilege mask, the AppliedTo and Criteria dialog boxes let you make selections from all possible combinations of attributes, but not all criteria or attributes make sense for all privilege masks. Evaluate the options you combine to make sure they function as expected.

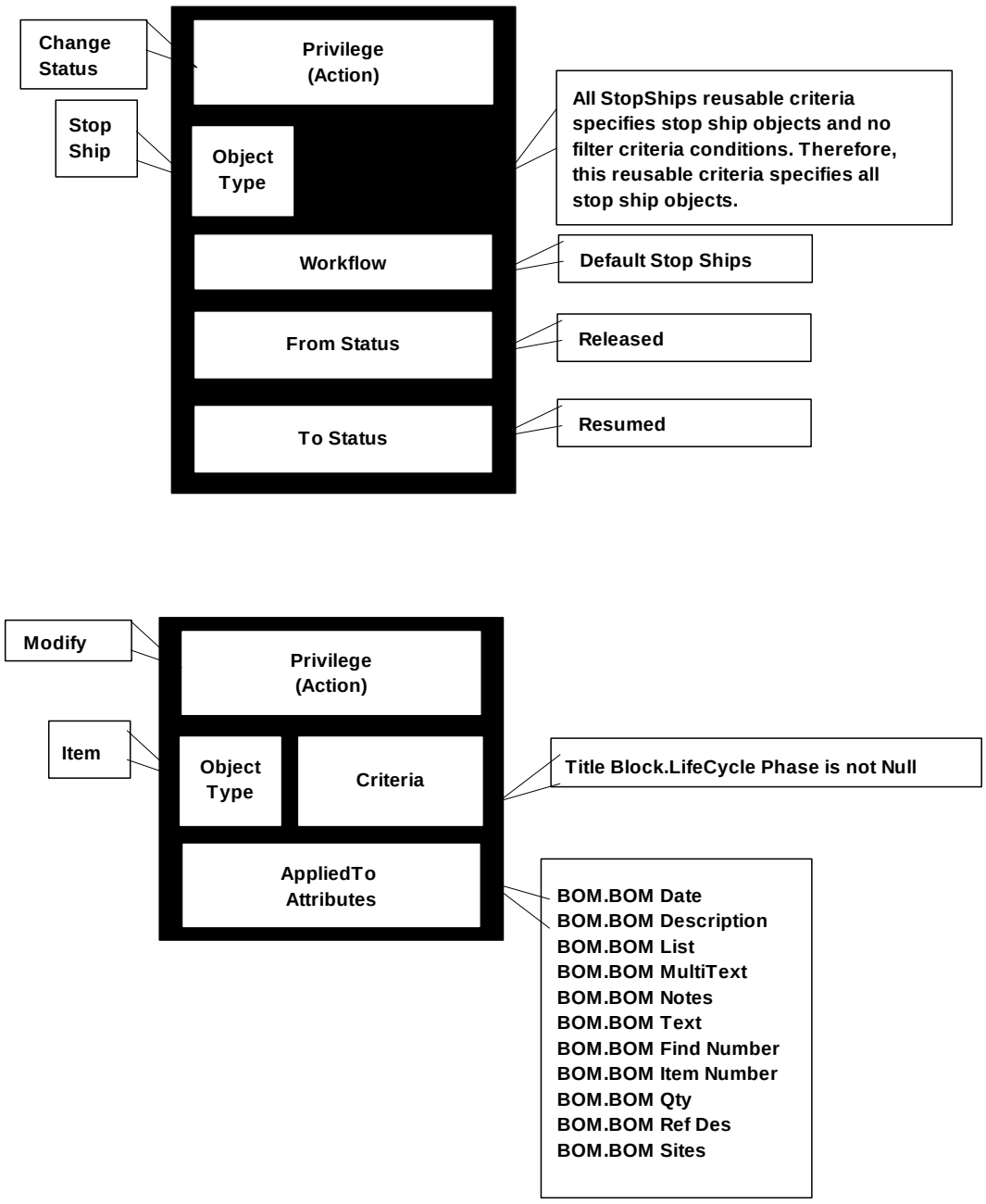
For example, when modifying the Modify Released Eng Changes privilege mask, you can select the Affected Items.Item Number and Affected Items.New Rev fields from the list in the AppliedTo dialog box. However, these two fields cannot be modified once the routable object has been released. Because you *can* specify these attributes in the AppliedTo dialog box, it doesn't necessarily follow that you *can* modify those fields on a released routable object.

The following are examples of privilege masks and their properties that are provided with your Agile PLM installation.




The Change Status privilege does not include AppliedTo properties, like the one in the figure above.

However, the Change Status privilege includes properties for Workflow, From Status, and To Status, as seen in the figure below.




# Discovery and Read Privileges

This section looks closely at Discovery and Read privileges.

The Read and Modify privilege masks are made specific by, among other properties, the AppliedTo

property naming individual attributes that can be accessed. (To contrast, the Administrator privilege mask is made specific by the AppliedTo property naming individual nodes that can be accessed.)

**Note** A section in the previous chapter, Advanced Search Security on page 197, describes enhancements to "search security" that apply to Discovery, Read, and Field-level Read privileges (not repeated in this section).

# Discovery Privilege

As outsourcing and “virtual companies” become more common, Agile PLM customers are allowing employees of other companies in their supply chain to access their Agile PLM database. For security reasons, it is important that these outside users see only information that applies directly to them. For example, you might not want one supplier to know that another supplier is providing the same part, or that you are providing the same part to a company and its close competitor.

The Agile PLM Discovery features—Discovery Privilege database property, Discovery privilege, and Discover privilege mask—are designed to address these security issues. These features control whether users are allowed to learn that certain objects exist in Agile PLM.

You control object discovery in two ways:

- At a global level, by setting the Discovery Privilege database property (in **Server Settings > Database**) to Enabled or Disabled.
- At the level of roles, with Discover privilege masks.

For example, if employees of one supply chain partner have the Discovery privilege only for parts provided by that partner, they can be blocked from seeing parts provided by other partners in search results tables or even on BOMs (see Related SmartRules on page 238).

**Note** If you grant users Read or Create privileges, you must also grant users a corresponding Discovery privilege for objects that they are allowed to read. Users should always be able to discover and read objects that they created.

For example, if you create a role that has a Read privilege mask applying to All Change Orders criteria, you must also include in that role a Discovery privilege mask applying to All Change Orders criteria.

For information about using the Discovery Privilege database property for global discovery control, see Database on page 357.

For more information about the Discover privilege masks assigned to each role supplied with Agile PLM, see Viewing a Role on page 189.

**Note** If a change originator or CCB member has the Discovery privilege removed after an ECO has been routed, the originator or CCB member still receives notifications but cannot view the change. For best results, avoid removing Discovery privileges after their initial assignment.

## Discovery Privilege and Sites

Site objects are explicitly filtered out of Discovery privilege criteria. Therefore, you cannot define a

new Discovery privilege mask to control a user's ability to access Sites.

To control a user's access to Sites, set user properties appropriately. Open a user in Java Client or Web Client, and specify the **Sites** and **Default Site** properties.

### Discovery Privilege and Life Cycle Phases

If you create Discovery privilege criteria based on RFQ Response lifecycle phases, you must restart the Agile Application Server to run a search that uses the criteria. Otherwise, the search won't return any results.

### Related SmartRules

The following SmartRules define how Agile PLM responds when users encounter objects that their privileges don't allow them to discover:

- Display BOM Tables
- Discovery Object Tables
- DiscoveryResultsTable

For more information about these SmartRules and their default settings, see SmartRules Defined on page 254.

### Discovery Privilege and Reports

Agile users do not see and cannot run the Administrator reports. The following information is about the availability of specific objects in the standard reports that users can run.

A user who does not have Discovery privilege for an object cannot include that object in a report. Parts for which a user does not have Discovery privilege are displayed in the same way they are displayed on BOMs.

If you, the administrator, have decided to display a warning message, that warning message will appear in the reports. You also have the option of displaying either the item number only or the item description only, so the user can see all the items in the report, but does not have access to the undiscovered items.

To run a report on a particular type of object, a user must be granted the following privileges:

- Read privilege
- Discovery privilege
- Run Reports privilege

**Note** The Manage Reports privilege alone is not sufficient to execute the report.

## Relationships among Discovery and Read Privileges

Agile PLM administrators can use the Discovery and Read privileges to grant users several levels of access to Agile PLM data, as shown in the following table.

| Access level | Discovery privilege | Read privilege |
|---|---|---|
| Full access | Granted | Granted for all tabs |
| Limited access | Granted | Not granted for **History** or **Workflow** tabs |
| Discovery only | Granted | Not granted |
| No discovery | Not granted | Not granted |

Privilege masks made up of the Discovery and Read privileges build on each other, as shown in the following table.

| Privilege mask type | Effects and comments |
|---|---|
| **Discover** | ▫ Can see that the discoverable object exists |
| **Read** | ▫ Must be able to discover the object<br>▫ Can open the object<br>▫ Can read all tabs as defined in the privilege mask's AppliedTo dialog box (see below) |

**Note** If you remove or modify a Read privilege mask, the Discovery privilege mask is not automatically removed or modified at the same time.

# Field-Level Read Privilege

"Field-level Read" describes an Agile PLM client user's ability or inability to read various fields within tabs through the Read privilege mask. Any field in any tab can be hidden from any user. A good use of Field-level Read is a customer hiding a "cost" field on the **Page Two** tab from certain suppliers.

To see a field of an object:

- A user must have a Read privilege to the object.
- The field must appear in the AppliedTo property for that user's Read privilege mask.

For example, in the **Privileges** window, double-click any Read privilege mask to display it. The **AppliedTo** field on the **General Information** tab lists all the fields to which the Read privilege is applied.

When you create a Read privilege mask, there are no values listed in the **AppliedTo** property. During the creation process, at the **AppliedTo** field, you click to display the selection dialog box. All the fields of the object are in the **Choices** list. You refine the privilege mask by moving fields from the **Choices** list on the left to the **Selected** list on the right.

If you wished to prevent the user from seeing the **Workflow** tab for the object specified by the criteria, all the **Workflow** tab fields would remain in the **Choices** list. Only the fields in the **Selected** list will be visible to the user. This is an example of Field-level Read.

The Field-level Read capability may be affected by whether the user has the Enforce Field-level Read privilege, which is discussed in “Enforce Field-Level Read Privilege” below.

## Modifying the AppliedTo Fields of Read Privilege Masks

This task is specific to Read privilege masks. The AppliedTo capability is used by a few other privileges – for more information, see AppliedTo Capability on page 228.

**To modify the list of fields to which a Read privilege mask is applied:**

1. Under **User Settings,** double-click **Privileges**. The Privileges window appears.

   **Note** Filter privilege mask records to narrow your search. For example, filter records by Privilege Contains Read to find all the Read privilege masks on the list. (See Filtering Data on page 9.)

2. Double-click the privilege mask you want to modify. The privilege mask window appears.

   The privilege mask properties are displayed on the **General Info** tab.

3. On the **General Info** tab, click next to the **AppliedTo** field. The selection dialog box opens.
4. Use the right and left arrows to move selected values from one list to the other list.
5. When you are finished, Click **OK**.
6. To finish editing the privilege mask, click **Save**.

# Enforce Field-Level Read Privilege

The Enforce Field-level Read privilege is a way the Agile PLM administrator can control certain small performance consequences when a user is under the restrictions of a Field-level Read (see Field-Level Read Privilege on page 239). The Enforce Field-level Read role comprises only the Enforce Field-level Read privilege mask.

If a user has a role with Enforce Field-level Read privilege included, the software checks everything at the
“field level,” that is, the fields within tabs in Agile PLM clients. Users with this privilege may notice their computers’ performance on some operations is slightly slower; this is because the software is checking all possible fields. This impact depends on how many items are in the tables being searched (for instance, BOMs, affected items, results).

If the user does not have a role with this privilege included, the system does not check everything at the field level, and so computer performance is not affected. In this case, even if the user has the Read privilege defined at the field level, the system does not check everything at the field level; that is done only when the Enforce Field-level Read privilege is present and enabled.

**Important** Previous versions of Agile PLM used the Specify Output Column privilege to permit users to select output attributes in setting up advanced searches. That privilege is removed; therefore, all users can potentially specify output attributes in advanced searches. However, you may want to assign the Enforce Field Level Read privilege to any user who did *not* have Specify Output Column privilege previously. This ensures that search results will be appropriately filtered—for example, for sensitive fields such as costs.

For a user with two roles, one with Enforce Field-level Read privilege and one without, the default is to "enforce" the Field-level Read. Similarly, for a user with a single role that has two Enforce Field-level Read privileges, one disabled and one enabled, the default is to "enforce" the Field-level Read. In both these cases, users can see only the fields for which they have the Read privilege. For this reason, Agile recommends that only users who are prevented from seeing certain fields have this privilege enabled.

System behavior with Enforce Field-level Read set to Yes interacts with certain privileges and SmartRule as follows:

- **Discovery privilege** — Dictates if the object is listed on Results and other tables.
- **User does not have the Read privilege for the object** — The user sees "no privilege" in the object's fields on the Results and most other tables. The display of these objects on the BOM table is governed by the Display BOM Tables SmartRule. The user cannot open these objects.
- **User has Read privilege for the Object Number and Description fields** — The user sees object number and description on Results and other tables. The user can open these objects.
- **User has Read privilege for all fields** — The user sees all fields on all tables. The user can open these objects.

## Display on Tables Due to Enforce Field-level Read

Up to Agile's Product Collaboration Release 7, if a user had a role that included the Enforce Field-level Read privilege, the display of objects that the user could not read but could discover (in the Results, Where Used, Affected Items, and Manufacturers tables) has "no privilege" in all the corresponding fields, and the user could open the objects to view them. Display of fields on the BOM table was governed by the Display BOM Tables SmartRule.

If you wish a user to "mimic" the 6.x behavior on the tables, you need to add the Read privilege to the user's roles for a subset of fields. The Read privilege could include the object number and description as AppliedTo attributes, and only these fields will be displayed on the returned tables.

For example, for a certain type of object, you might define Attachment Type values such as Internal Only and External, and then give users outside the company privileges only for files marked as External. If an Agile PLM client user adds an attachment to that type of object and defines it as Internal Only, outside suppliers and contractors won't be able to access the file.

To use the Attachment Type attribute in privilege mask criteria, the object type that you specify cannot be a base class (Items, Changes, Reports, and so forth).

**Note** If you decide to enable the Attachment Type attribute to control access to attachments, be sure to tell the users how to use the field. This field can be edited in Agile PLM clients.

# Modify Privilege

The crucial Modify privilege is covered generally in “Some ‘Modify’ Basics and Rules” below and in an important specific context in Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV on page 243. Again, the Read and Modify privilege masks are made specific by, among other properties, the AppliedTo property naming individual attributes that can be accessed.

## Some ‘Modify’ Basics and Rules

The Modify privilege allows a user to modify or edit fields on the tab of an object. Modify privileges are assigned using the AppliedTo attribute settings.

**Note** The Create privilege allows the user to fill in required fields at the time of object creation even if the user does not have Modify privilege for those fields. (Required fields are fields with their attribute property **Required** set to Yes; see Defining Attribute Properties on page 77.)

- The Modify privilege is required for users of the Agile PLM Import utility.
- Modify privilege and Relationships: To create a relationship between two objects without setting up a rule (detailed in *Getting Started with Agile PLM*), you must have the appropriate Modify privilege for both objects tailored to Relationships. To create a rule between two related objects, you must, in addition, have the appropriate Modify privilege for both objects tailored to Rules. For example, for items, a user must have a Modify Items privilege mask that includes, in its AppliedTo property, the **Relationship** tab "Name" attribute and the **Relationship** tab "Rule" attribute.
- Users with the Modify Manufacturer Parts privilege mask can modify Manufacturer Parts attributes that also appear on the **Manufacturers** tab without an MCO or ECO being issued.

**Note** A user must be assigned the Modify privilege for both **Manufacturer.Mfr Name** and **Manufacturer.Mfr Part Number** fields to properly modify Manufacturer (AML) information.

- Item, New Rev, and Type (Lifecycle) fields on the **Affected Items** tab of changes cannot be modified at Released or Implemented statuses.
  - For other changes, the item number fields on the **Affected Items** tab cannot be modified at Released, Closed, Expired, or Resumed statuses.
- To be able to modify a **Page Three** tab, users must have a Modify privilege mask for the specific subclass and attributes.

**Important** Although Agile PLM can be configured to allow modification of the lifecycle phase of previously released revisions by modifying released ECOs, such a practice is highly discouraged because of the following reasons:

**a)** Potential regulatory compliance violations

**b)** Potential data integrity issues in the system

It is strongly recommended that user privileges be configured not to allow modification of

lifecycle of released revisions by directly modifying released ECOs.

Agile PLM does not allow you to modify the item Rev and item Description on the **Affected Items** table of a released ECO, even if you have Modify ECO privileges that would, in theory, allow that.

The Modify privilege is also utilized with regard to attachments; this capability is documented in Modify Privilege and Attachments on page 393.

# Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV

Modify privilege masks for item objects that use $CURRENTREV in their criteria can be used to control a user's ability to modify (or inability to modify) items that are either released, preliminary, or introductory. To further define a user's modify ability, you can use multiple $CURRENTREV conditions to identify an Introductory revision, the latest revision, and the workflow statuses of the change order associated with the selected item revision.

## Applicable Item Attributes

On item objects, the user can use the **Rev** drop-down list to display information for a specific revision. The user's assigned Modify privilege masks determine whether he can modify those attributes.

- **Revision** drop-down list is available for the item **Sites** tab in both Web Client and Java Client.
- **Revision** drop-down list is available for the item **Page Two** and **Page Three** tabs in Java Client
- Because the item **Page Two** and **Page Three** data in Web Client is displayed on the **Title Block** tab, the **Revision** drop-down list on Web Client Title Block tab also controls display of Page Two and Page Three data.

## $CURRENTREV Criteria Logic

Agile has a special variable that can be used in Criteria against items, $CURRENTREV, that allows administrators to build in privilege control based on the displayed revision. With a criteria written using $CURRENTREV, Agile evaluates the currently selected revision in Web Client or Java Client to evaluate the match. That is, when a user has selected a revision in the **Revision** dropdown list, the $CURRENTREV criteria is evaluated against that displayed revision.

This functionality is discussed in Controlling the Ability to Modify Items at Introductory Revision with $CURRENTREV on page 243.

Using $CURRENTREV in an item criteria allows the administrator to build in privilege control based on the displayed revision. Using combinations of multiple $CURRENTREV criteria conditions allows the administrator to narrowly define which items a user is allowed to modify.

A $CURRENTREV criteria condition can be set to be **Equal To** or **Not Equal To** the following values:

- $LATEST (the latest released revision of the item)
- Status type variables, for example, $STATUSTYPE.RELEASED.
- Workflow statuses pertaining to specific workflows, for example, Default Change

Orders.Pending.

- Special values for evaluating Introductory revisions:
  - $INTRODUCTORY_NOCHANGE
  - $INTRODUCTORY_PENDINGCHANGE
  - $INTRODUCTORY_RELEASEDCHANGE

For detailed information about $CURRENTREV criteria values, see Using $CURRENTREV and $LATESTREV on page 123.

### *Definition of a Released Revision*

A Released Revision occurs when the change object (Change Order or Manufacturer Order) that created the Revision is in its workflow at a status type of Released or Implemented. This is indicated in both clients when the Revision is not in parentheses.

### *Definition of a Pending Revision*

A Pending Revision occurs when the change object that created the Revision is in its workflow at a status type of Unassigned, Pending, Submit, Review, or Hold. This is indicated in both clients when the Revision is in parentheses.

**Note** A change object in Canceled status is not applicable because it will not appear in the Revision dropdown listing.

## Building Criteria for Specific Use Cases

Typically, there are three cases where the Agile administrator needs to control Modify privileges. These can be extended with additional criteria as are needed.

1. Allow broad Modify at Introductory only
2. Allow targeted Modify for a Pending revision
3. Allow targeted Modify for a Released revision

### *1. Allow broad Modify at Introductory only*

**Items ( $CURRENTREV Equal To $INTRODUCTORY_NOCHANGE OR**

**$CURRENTREV Equal To $INTRODUCTORY_PENDINGCHANGE OR**

**$CURRENTREV Equal to $UNASSIGNED )**

This combination of statements will allow a user to modify the Introductory revision of an item if the item has never been released. The item may have no changes or it may have a pending change.If the item has a pending change, the user must select the Introductory revision in the Rev dropdown list.

Use of a criteria with this combination of statements in a Modify Item privilege mask with many AppliedTo fields will allow broad Modify privilege at Introductory while there is no Released revision, and then prevent broad Modify privilege after the first Released revision when there is a Pending revision.

### *2. Allow targeted Modify for a Pending revision*

**Items ( $CURRENTREV Equal To $UNASSIGNED OR**

**$CURRENTREV Equal To $STATUSTYPE.PENDING OR**

**$CURRENTREV Equal To $STATUSTYPE.SUBMIT OR**

**$CURRENTREV Equal To $STATUSTYPE.REVIEW OR**

**$CURRENTREV Equal To $STATUSTYPE.HOLD )**

Writing a criteria with these statements will generally satisfy this requirement. Use this criteria in a Modify Item privilege mask with a small number of AppliedTo fields. When the user selects a pending revision from the Rev dropdown list, he will be able to modify the specified fields of the selected pending item revision.

### *3. Allow targeted Modify for a Released revision*

**Items ( $CURRENTREV Equal To $STATUSTYPE.RELEASED OR**

**$CURRENTREV Equal To $STATUSTYPE.IMPLEMENTED )**

This combination of criteria statements evaluates as True when the workflow status of change that released the selected item revision has a status type of either Released or Implemented. Use this criteria in a Modify Item privilege mask with a small number if AppliedTo fields. When the user selects a Released revision in the Rev dropdown list, he will be able to modify the specified fields of the selected released item revision.

### *$CURRENTREV Criteria Examples*

This section shows examples of ten different criteria statements using $CURRENTREV and $LATESTREV and illustrates how they are evaluated for different item revisions.

The following table shows the ten criteria that are used in the examples:

| Criteria # | Description |
|---|---|
| 1 | $CURRENTREV Equal To $INTRODUCTORY_NOCHANGE |
| 2 | $CURRENTREV Equal To $INTRODUCTORY_PENDINGCHANGE |
| 3 | $CURRENTREV Equal To $INTRODUCTORY_RELEASEDCHANGE |
| 4 | $CURRENTREV Equal To $LATEST |
| 5 | $LATESTREV Is Introductory |
| 6 | $LATESTREV Is Released |
| 7 | $CURRENTREV Equal To $UNASSIGNED or<br>$CURRENTREV Equal To $STATUSTYPE.PENDING |
| 8 | $CURRENTREV Equal To ECO.Released or<br>$CURRENTREV Equal To ECO.Implemented |
| 9 | $CURRENTREV Equal To Default Change Orders.Implemented |
| 10 | $CURRENTREV Equal To $INTRODUCTORY_NOCHANGE or<br>$CURRENTREV Equal To $INTRODUCTORY_PENDINGCHANGE or<br>( $LATESTREV Is Introductory and |

| Criteria # | Description |
|---|---|
| | ( $CURRENTREV Equal To $UNASSIGNED or<br>$CURRENTREV Equal To $STATUSTYPE.PENDING ) ) |

The following table shows five items (EX-1 – EX-5) and the revision history for each of these items. This table illustrates how each of the criteria detailed in the above table (Criteria #1 – #10) will evaluate (T = True; Blank table cell = False) when a specific revision is selected from the Rev dropdown list of the example items.

| Item Number | Revision | Change Status | Status Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EX-1 | Introductory | N/A | | T | | | T | T | | | | | T |
| EX-2 | Introductory | N/A | | | T | | T | T | | | | | T |
| | (?) | Unassigned | | | | | | T | | T | | | T |
| EX-3 | Introductory | N/A | | | | T | | | T | | | | |
| | A | ECO.Implemented | Complete | | | | T | | T | | T | | |
| EX-4 | Introductory | N/A | | | | T | | | T | | | | |
| | A | ECO.Implemented | Complete | | | | T | | T | | T | | |
| | (?) | Unassigned | | | | | | | T | T | | | |
| EX-5 | Introductory | N/A | | | | T | | | T | | | | |
| | A | ECO.Released | Released | | | | T | | T | | T | | |
| | (B) | ECO.CCB | Review | | | | | | T | | | | |
| | (C) | ECO.Submitted | Submit | | | | | | T | | | | |
| | (D) | ECO.Pending | Pending | | | | | | T | T | | | |

# Applying Create User Criteria

Create User criteria statements restrict users to reading only those objects they have created. The Create User feature allows a user without a read privilege to create an object. To implement Create User criteria, you need to first make both of the following visible:

- The **Page Two** tab of that class (see Showing, Hiding, or Renaming a Tab on page 64).
- The **Create User** attribute field on the **Page Two** tab of an Agile PLM class (see Showing or Hiding an Attribute on a Tab on page 84).

Once you've done this, Agile PLM clients automatically display the creator's name in the **Create User** field of the **Page Two** tab *for that class*. Any user who has a Create privilege can now open a blank form *for that class* and create an object. In this way, users don't need to have a basic Read privilege to create objects. However, this privilege limits users to reading only pending or preliminary objects they've created.

For users who have the Create privilege, you can expand the Read privilege to include all objects

they've created, regardless of the state, by creating a Read privilege mask for the subclass with a reusable criteria that specifies Page Two.Create User criteria conditions. In addition to making the **Page Two** tab visible and making visible the **Create User** field on the tab, you need to:

- Create a reusable criteria that specifies the Page Two.Create User attribute, typically Page Two.Create User Equal to $USER.
- Create a privilege mask that specifies the reusable criteria you created.

For example, for a user who creates ECOs, create a new privilege mask as follows:

- **Name** — Read ECOs
- **Description** — Read-Only ECOs created by this user
- **Privilege** — Read
- Select or create a reusable criteria that specifies:
  - **Type:** ECO
  - **Attribute:** Page Two.Create User
  - **Match If**: Equal To $USER

You can read the Create User attribute into the **Manufacturers** tab for both Manufacturers and Manufacturer Parts by making it visible on these tabs. Remember to first make the **Page Two** tab and its Create User attribute visible for the class.

# Using Privilege Mask Criteria Variables

You can use the variables listed and described in the following table when creating a reusable criteria for use as a privilege mask criteria. In the Create Criteria dialog box, some variables appear on the **Attribute** list, and others can be entered in the **Value** field. See Agile PLM Variables on page 14 for a complete list of Agile PLM variables.

| Variable | Description |
|---|---|
| $CHECKOUTUSER | The user who checked out the object's attachment. |
| $LATEST | The latest revision of an object; more specifically, the latest released revision for a released item, or the latest pending revision for an unreleased item (with Pending changes); $LATEST applies only to the Read and Modify privilege. |
| $CREATEUSER | The user who created the object.<br>To specify the Create User as an attribute (for example, Create User equal to $USER), select Page two.Create User in the **Attribute** field when you are defining a reusable criteria. See also Applying Create User Criteria on page 246. |
| $PARTNER | The current supply chain partner company. |
| $CURRENTREV | The current revision of an object you have selected from the rev list, or the revision of an object you are currently viewing. |

# Chapter 18

# Other User Settings

**This chapter includes the following:**


- User Monitor ..... 249
- Deleting and Undeleting Users ..... 250


# User Monitor

The User Monitor window lists the users that are presently logged in to the Agile PLM system. It displays the following information about each logged-in user.

| Table column | Description |
|---|---|
| **User Name** | The first and last name of the logged in user. |
| **User ID** | The login username of the user. |
| **Host** | Indicates the user's host. |
| **Login Time** | The time the user logged in. |

## Refreshing the User Monitor Window

The User Monitor window is not dynamic; it shows the user session information at the moment that the window was displayed. To see the most current information, click **Refresh** to update the information displayed in the window.

## Terminating a User Session

You can use the User Monitor window to terminate a user session.

**To terminate a user session in Java Client:**

1. Under **User Settings,** double-click **User Monitor**.
2. Select one or more users whose sessions you want to terminate.
3. Click the **Terminate Session** button.

   The users sessions are not terminated immediately. It may take up to a minute to terminate active user sessions. When you click the **Terminate Session** button, users can continue working for one minute. When the session is terminated, the login window is displayed in the user's browser.

# Deleting and Undeleting Users

You cannot either “hard delete” or “soft delete” a user that is associated with any other object. A newly created user who is not associated can be deleted from the **Users** node. You will then find the user object on the **Deleted Users** node.

This process can be reversed: on the **Deleted Users** node, select a deleted user and click the **Undelete** button. The user object is restored to the **Users** node; you will, however, have to close and re-open the **Users** node to see this user, this cannot be accomplished with the **Refresh** button.

If someone leaves the company, or will no longer be using Agile PLM, or you need to prevent a user from logging in to the Agile PLM system, disable the user object by setting the Status property to Inactive. The user remains on the **Users** node, and the name still appears on existing workflows, escalations, and so forth, but the user will no longer appear in the Address Book for other users to select. You can re-set the user to Active status at any time.

For users you have made Inactive, you can use Global Replace to locate and replace him in all the places in the system that his name is used (see Global Replace on page 22).

## Deleting a User Group

You can delete a user group that is no longer needed. If you delete a user group to which users have been assigned, you can do so without removing the users who were in the user group: when you delete the user group, user assignments to that user group are also deleted.

When you delete a user group, it is removed from the Agile PLM address book. You cannot delete a user group if it is in use in these cases:

- It is on any routable object signoff list in Agile PLM clients.
- It is listed in any workflow “Notify” properties (in the Workflows window).
- It is used as a designated escalation person for any users, user groups, and partners.

**Note** When you remove a user from a user group, use the Global Replace tool to determine all the places in the system that user is used (see Global Replace on page 22).

**To temporarily delete a user group in Java Client:**

1. Under **User Settings,** double-click **User Groups**. The User Groups window appears.
2. Select a user group name in the list.
3. Click the **Delete** button. You are prompted to confirm the deletion.
4. Click **Yes**. The user group no longer appears on the User Groups window.

**To temporarily delete a user group in Web Client:**

1. Click **Tools and Settings > Address Book > User Groups**. The User Groups page appears.
2. Click a user group name in the list. That user group page appears.
3. From the **Actions** menu, choose **Delete**. You are prompted to confirm the deletion.

4. Click **Yes**. The user group no longer appears on the User Groups page.

## Hard-Deleting a User Group

When you delete a user group, it is moved to the **Deleted User Groups** node. Although it does not appear in the address book, it still exists in the database. This is referred to as a "soft-deleted" user group. To permanently remove a user group, you can delete it from the **Deleted User Groups** node. This is referred to as a "hard-deleted" user group.

| | |
|---|---|
| **Note** | A hard-deleted user group is permanently removed from the database and cannot be restored or undeleted. |

**To hard-delete a deleted user group in Java Client:**

1. Under **User Settings,** double-click **Deleted User Groups**. The Deleted User Groups window appears.
2. Click to select the user group you want to hard-delete, and click the **Delete** button.

   The user group no longer appears in the Deleted User Groups window.

## Undeleting a User Group

From the **Deleted User Groups** node, you can view user groups that have been deleted from the Agile PLM system. It is useful to know which user groups have been deleted because the user group may have been assigned as an approver for routable objects. This allows you to maintain a complete "audit trail" of a product's entire history.

A user group may be "undeleted." By undeleting a deleted user group, you restore it to active use, and it appears in the address book again.

**To undelete a user group in Java Client:**

1. Under **User Settings,** double-click **Deleted User Groups**. The Deleted User Groups window appears.
2. Click to select the user group you want to undelete.
3. Click the **Undelete** button.
4. The user group's name is added to the User Groups window. You might have to click the **Refresh** button to see the group listed in the window.

# Chapter 19

# SmartRules

**This chapter includes the following:**


- Modifying SmartRules Settings..... 253
- SmartRules Defined..... 254


The Agile PLM SmartRules are documented in SmartRules.

# Modifying SmartRules Settings

**To modify a SmartRules setting:**

1. Under **System Settings,** double-click **SmartRules**. The SmartRules window appears.
2. At the rule you want to change, select a setting from the drop-down list. The list of possible settings can include the following values:
   - **Allow** — The system allows the process to occur.
   - **Disallow** — The system prevents the process from occurring.
   - **Warning** — The system warns the user that the process is occurring.
   - **Remove** — Removes all approvals/rejections on the **Workflow** tab when the routable object returns to the status where first rejected.
   - **Retain** — Keeps all approvals/rejections on the **Workflow** tab when the routable object returns to the status where first rejected.
   - **Display** — Displays the object in the table.
   - **Display with Just Description** — Displays the object with its subclass and description. The part number and other values read from the object are not displayed.
   - **Display with Just Part Num and Rev** — Displays the object with its subclass, part number, and revision. The description and other values read from the object are not displayed.
   - **No Display** — Does not display the object in the table. No warning is given.
   - **No Display with Warning** — Does not display the object in the table. Issues the following on the tab: “There are x objects not displayed due to insufficient user privilege.”
3. When you have finished, click **Save**.
4. If you wish to revert to the original settings or start over, click **Close**. You will be prompted whether you want to close the window without saving; click **Yes** if you do or **No** if you do not.

# SmartRules Defined

## Allow Deletion of Attachments

Settings: Yes, No (default)

When set to Yes, upon removing an attached file from a business object Attachments tab, if the file folder is not in use by any other business object (that is, no versions of the file folder are referenced by a business object), the file folder is automatically soft-deleted.

A user with the appropriate search capabilities can run the File Folder Recycle Bin Search and hard-delete (or undelete) the soft-deleted file folder.

## Auto-Installs from PPM

Settings: Allow (default), Disallow, Warning

Allows or disallows automatic installation of Microsoft Project (2002 and 2003) DLLs into a user's system registry. Installation of these DLLs enables seamless publishing using an Agile menu within Microsoft Project. It should be set to Disallow if your company does not want any applications to be automatically installed, or if you prefer to set up an Agile–MS Project connection systemwide.

## Auto Publish Quote History

Settings: Allow (default), Disallow, Warning

Allows or disallows RFQ or response-line data to be published to objects in the Quote History subclass.

## BOM Multi-Level Recursion

Settings: Allow (default for non-Oracle DB systems, and for upgrading customers), Disallow (default for Oracle DB systems)

For systems based on Oracle DB, this SmartRule validates whether a BOM recursion can occur, that is, when a subassembly includes one of its parent items from an upper level of the BOM tree structure on its own BOM. (A recursive BOM structure, if carried to its logical conclusion, would repeat indefinitely).

The validation happens when a change order (ECO, SCO, and MCO) advances to Released, or when Release Audit is performed, or when the change tries to advance when current status needs to Pass Release Audit (assuming SR setting is Disallow).

**Note** Because non-Oracle DB systems do not allow or check for recursion, this SmartRule has no effect; these systems should see the default setting is Allow.

**Important** The PG&C and PCM solutions do not support BOM recursion. For companies using either of these solutions, set this SR to Disallow.

## Change Status Approver Rejected Change

Settings: Allow, Disallow, Warning (default)

Determines whether to allow movement of the routable object to the next status or warn the user that an approver has rejected the change. It affects all Review or Released status types.

If you require the ability to determine if a routable object should be allowed to proceed when it does not meet the SmartRule, set it to Disallow. A user with Override privilege can advance the routable object in the workflow. Please see the Caution about this property in Override.

**Note** To make sure that a routable object is not autopromoted through the workflow after a user has rejected it, set this SmartRule to “Disallow.” Setting it to “Warning” will, in some cases, allow the routable object to be autopromoted through the workflow even when that is not the intended behavior.

## Change Status No Response Change

Settings: Allow, Disallow, Warning (default)

Determines whether to allow movement of the routable object to the next status when an approver has not responded with either an approval or a rejection. It affects all Review or Released status types and applies only to changes that are being manually moved to the next level. Autopromotion requires that each approver either approve or reject the routable object.

## Change Status Observer Rejected Change

Settings: Allow, Disallow, Warning (default)

Determines whether to allow movement of the routable object to the next status or warn the user that an observer has rejected it. It affects all Review or Released status types.

If you require the ability to determine if a routable object should be allowed to proceed when it does not meet the SmartRule, set it to Disallow. A user with Override privilege can promote the routable object. Please see the Caution about this property in Override on page 219.

## Check In Attachment With Different File Extension

Settings: Allow, Disallow (default), Warning

Determines whether an attachment file that is being checked in can have a different file extension than the file that was checked out.

**Note** This rule is overridden when an item’s attachment is Latest-x and the file extension in the attached file folder is changed and is therefore different than the extension of the item’s checked-out file.

## Copy Files to Rev

Settings: Copy, Copy with Warning, Disallow, Reference (default), Reference with Warning

Controls automatic copying of files to a new revision. The system checks this SmartRule only when a pending change is created and not when attachments are added. The settings work as follows:

Copy – For the item's new pending revision, Agile creates a new file folder and creates a new copy of the attachment file, which is placed in the new file folder.

Reference – Agile uses the existing file folder and creates a new reference to it on the item's new pending revision Attachments tab.

Disallow – The new item pending revision is created with no attachments.

Copy with Warning – The same as the Copy setting, with the option of choosing no attachments for the new pending revision.

When you add items to the pending ECO, you are presented with a warning dialog that includes a row for each item you are adding.

To add the item and to copy the attachments, check both the Add checkbox and the Attachments checkbox.

To add the item with no attachments, check only the Add checkbox.

Reference with Warning – The same as the Reference setting, with the option of choosing no attachments for the new pending revision.

When you add items to the pending ECO, you are presented with a warning dialog that includes a row for each item you are adding.

To add the item and to reference the existing attachments, check both the Add checkbox and the Attachments checkbox.

To add the item with no attachments, check only the Add checkbox.

**Note** If you set this rule to Disallow, files that were added to the Introductory revision after the object was associated with a change (that will result in released Rev A) will not be copied to subsequent revisions. Also, when a new Change is created, all the files that were added to the Introductory revision won't be added to the newly created revision.

## Demotion Approvers Removal

Settings: Retain (default), Remove

Defines whether ad hoc (not the default) approvers and observers will be removed from the list when the routable object is rejected and returned (that is, demoted) to a previous status. It affects all Review and Released status types in all workflows.

Retain: When a workflow is returned to a previous status (including Pending), the ad hoc approvers

are retained and they will see the change when it advances through the workflow again.

Remove: When a workflow is returned to a previous status (including Pending), the ad hoc approvers are removed – they will not see the change again unless the originator or change analyst adds them back.

## Demotion Signoff Removal

Settings: Retain (default),Remove

Defines whether approvals/rejections already recorded will be removed from the routable object's Workflow tab when it is rejected and returned (that is, demoted) to a previous status. It affects all Review and Released status types in all workflows.

Retain: When a workflow is returned to a previous status (including Pending), the signoffs that are already recorded are retained.

Remove: When a workflow is returned to a previous status (including Pending), the signoffs are removed and approvers will have to sign off again when the change advances through the workflow.

## Detailed TimeSheet Entry

Settings: Disallow (default), Allow

Controls the display of the Timesheet tab in Product Portfolio Management.

Disallow: The Timesheet tab and all associated actions are disabled.

Allow: The Timesheet tab is displayed next to the My Assignments tab for users to enter Actual Hours against a task or activity. When the Timesheet is enabled, data entry for Actual Hours on the Team tab is disabled.

## Discovery Object Tables

Settings: Warning (default), No Display

Controls the display of information about an object on the Change History, Where Used, Affected Items, Manufacturers, BOM, or Pending Changes tables when the user doesn't have the Discovery privilege for the object. Possible settings are Warning (the default) and No Display.

If the Discovery Privilege property in the Database node is disabled, then so is this SmartRule.

| | |
|---|---|
| **Note** | When this setting is changed, users must exit the Agile Java Client and log in again before the change takes effect. |

## Discovery Results Table

Settings: Warning (default), No Display

Controls the display of information about the Results table when the user does not have the Discovery privilege for an object in the table.

If the Discovery Privilege property in the Database node is disabled, so is this SmartRule.

| | |
|---|---|
| **Note** | When this setting is changed, users must exit Java Client and log in again before the change takes effect. |

## Display BOM Tables

Settings: No Display with Warning (default), Display, Display with Just Description, Display with Just Part Num and Rev, No Display

Controls the display of information about an item in the BOM table when the user does not have the Discovery privilege for the item.

| | |
|---|---|
| **Note** | When this setting is changed, users must exit Java Client and log in again before the change takes effect. |

## Display Structure Tables

Settings: Display (default), No Display

Controls the display of model structures that are created by objects of the Designs class.

## Duplicate Find Numbers

Settings: Allow, Disallow, Warning (default)

Controls the use of duplicate find numbers when users add items to a BOM.

This SmartRule has been expanded to dictate whether or not the same find number can be used in a site-specific portion of a BOM as is used in the global portion of a BOM. It also dictates whether or not the same find number can be used within a site-specific portion of the BOM.

For example: If the Duplicate Find Numbers SmartRule is set to Disallow, then:

Find Number 1 cannot be used for Part 123 on the Milpitas BOM if Find Number 1 is used for Part 234 on the Global BOM.

Find Number 1 cannot be used for Part 123 on the Milpitas BOM if Find Number 1 is also used for Part 234 on the Milpitas BOM.

The same find number can be used in more than one site-specific portion of the BOM regardless of the Duplicate Find Numbers SmartRule setting.

For example: Find Number 1 could be used for Part 123 on the Singapore section of the BOM and Find Number 1 could be used for Part 234 on the Milpitas section of the BOM. The setting of the Duplicate Find Numbers SmartRule does not affect this behavior.

If the Duplicate Find Numbers SmartRule is set to Allow, any item can have the same find number as another item regardless of what site section of the BOM they are on.

## Duplicate Ref Des

Settings: Allow, Disallow ,Warning

Controls whether duplicate reference designators can be used on a BOM. This SmartRule has been expanded to dictate whether or not the same reference designator can be used in a site-specific portion of a BOM as is used in the global portion of a BOM. It also dictates whether or not the same reference designator can be used within a site-specific portion of the BOM.

For example: If the Duplicate Ref Des SmartRule is set to Disallow, then:

Reference Designator 1 cannot be used for Part 123 on the Milpitas BOM if Reference Designator 1 is used for Part 234 on the Global BOM.

Reference Designator 1 cannot be used for Part 123 on the Milpitas BOM if Reference Designator 1 is also used for Part 234 on the Milpitas BOM.

The same reference designator can be used in more than one site-specific portion of the BOM regardless of the Duplicate Ref Des SmartRule setting.

For example: Reference Designator 1 could be used for Part 123 on the Singapore section of the BOM and Reference Designator 1 could be used for Part 234 on the Milpitas section of the BOM. The setting of the Duplicate Ref Des SmartRule does not affect this behavior.

If the Duplicate Ref Des SmartRule is set to Allow, any item can have the same reference designator as another item regardless of what site section of the BOM they are on.

## Effectivity Date Order

Settings: Allow, Disallow, Warning (default)

Controls whether a new rev can be released with an effective date that is earlier than one or more old revisions. This SmartRule also applies to the effectivity dates for site-specific items on the Affected Items table.

## Effectivity Gap

Settings: Allow, Disallow, Warning (default)

Controls gaps between the obsolete date of an existing part and the effective date of its replacement part on affected items. This SmartRule also applies to the effectivity dates for site-specific items on the Affected Items table.

## Effectivity Overlap

Settings: Allow, Disallow, Warning (default)

Controls overlaps in the effective dates of an existing part and the obsolete date of its replacement parts on an affected item. This SmartRule also applies to the effectivity dates for site-specific items on the Affected Items table.

## Enable Addition of Activities and Gates to Completed

## Activities

Settings: Allow, Disallow (default)

Controls ability to add new data from PPM Activities or Gates to completed (Released) Activities.

## Enable Comments for Deleting, Archiving & Delegating

Settings: Disallow (default), Allow

Controls whether comments can be recorded while performing Delete, Archive, or Delegate actions in Product Portfolio Management.

## Force Commodity and Part Family to be Identical

Settings: Yes (default), No

This rule applies only to customers that have both the PCM and PG&C solutions. Otherwise, you can ignore it.

The rule controls whether legacy Part Family objects (in PG&C solution) and new Commodity objects (in PCM solution) are to be treated by the system as Commodity objects that are shared by both solutions. The default is Yes.

If you want to treat Part Families and Commodities differently, set the rule to No. However, you must also configure the Commodities class to make the rule effective. For more information, see Part Groups: Configuring Part Families.

## Items Released First

Settings: Allow, Disallow, Warning (default)

Controls whether an item can be released when the item's BOM contains unreleased items. This SmartRule is pertinent to multiple sites: when a "parent" item is released, the rule checks the BOM components for site association and whether the BOM components are also released for that particular site.

## Many Items per PSR

Settings: Allow, Disallow (default), Warning

Allows or limits the association of items (parts or documents) with PSRs (problem reports or NCRs).

## Many QCR per PSR

Settings: Allow, Disallow (default), Warning

Allows or limits the association of QCRs (CAPAs or audits) with PSRs (problem reports or NCRs).

## Multiple Items Per Manuf Part

Settings: Allow, Disallow, Warning (default)

Controls whether a manufacturer part can have multiple item parents.

If set to Disallow or Warning (the default), the system checks items as the change order's Workflow performs a Release Audit. This function has been enhanced; for more information, see Release Audit, or Audit Current Status for Conditions to Release.

## Negative Value For Material Price Adder Fields

Settings: Allow, Disallow (default), Warning

Controls the entering of “negative prices” in the Material Price Adder attribute.

## Negative Value For Material Price Fields

Settings: Allow, Disallow (default), Warning

Controls the entering of “negative prices” in the Material Price fields.

## Negative Value For Non-Material Price Fields

Settings: Allow, Disallow (default), Warning

Controls the entering of “negative prices” in Non-Material Price fields.

## Notify Resource Pool Owner for Proposed Programs

Settings: Allow (default), Disallow

Controls the sending of notifications to resource pool owners for programs in Proposed state.

## Overlap Price Line Effectivity Periods

Settings: Allow, Disallow (default), Warning

Controls overlapping effectivity periods of price lines on price objects and PCOs.

## PSR Contains Items and Related PSRs

Settings: Allow, Disallow (default), Warning

Controls whether a PSR can be associated with both affected items and related PSRs (Allow), or only affected items or related PSRs (Disallow).

| | |
|---|---|
| **Note** | There is no business logic between items on the Affected Items tab and items on the Related PSR tab (that is, an item listed on Related PSR will not be copied to the Affected Items of the parent PSR). |

There is business logic between PSRs and QCRs, so if a PSR is associated with a QCR, items from both the PSR's Affected Items tab and its Related PSR tab will automatically populate the QCR's Affected Items tab.

## Redline Attributes

Settings: Allow, Disallow (default)

Controls the display in Web Client of Redline Title Block tab for Change Orders and Manufacturer Orders.

## Redline Manufacturers On Change Order

Settings: Allow (default), Disallow

Controls whether users are allowed to redline the Manufacturers tab from an ECO. If the setting is Disallow, all buttons in the Redline Manufacturer table are disabled. Possible settings are Allow (the default) or Disallow.

## Release QCR With Un-resolved Items

Settings: No Display (default), Warning

Controls whether a QCR can be released either manually or through auto-promotion even though all items on the Affected Items tab have not been associated with a Change.

If you set it to No Display, the SmartRule allows a QCR to be released even though all its affected items do not have an associated Change. Such a QCR will also pass a release audit without a warning.

If you set it to Warning, the SmartRule warns users that a QCR is being released even though all its affected items do not have an associated Change.

## Released Rev Required

Settings: Allow, Disallow, Warning (default)

Controls whether a routable object can be unreleased if has an affected item that is on the BOM of a released assembly.

## Unrelease Change Order

Settings: Allow, Disallow, Warning (default)

Controls whether change orders can be unreleased.

## Unrelease Change Request

Settings: Allow, Disallow, Warning (default)

Controls whether change requests can be unreleased.

## Unrelease Deviation

Settings: Allow, Disallow, Warning (default)

Controls whether deviations can be unreleased.

## Unrelease Manufacturing Order

Settings: Allow, Disallow, Warning (default)

Controls whether manufacturer orders can be unreleased.

## Unrelease Stop Ship

Settings: Allow, Disallow, Warning (default)

Controls whether stop ships can be unreleased.

## Workflow Matching Criteria Attribute Modification

Settings: Allow, Disallow (default), Warning

| | |
|---|---|
| **Note** | None of the choices result in the workflow being altered. |

A user may modify an attribute by giving it a value that does not match the workflow's matching criteria. This SmartRule governs whether the system accepts the changed attribute.

| | |
|---|---|
| **Note** | If this SmartRule is set to Warning or Disallow, any attempt to modify attributes of the routable object that are specified in the workflow's matching criteria results in an immediate warning or prevention of the modification. However, if the workflow's matching criteria includes a reusable criteria that specifies an item's attribute, the user will be allowed to modify the value of that item attribute to a value that does not match the reusable criteria. No warning will be given. If, after that item attribute has been modified, the user then modifies the routable object in any way, the user is warned that the routable object no longer matches the workflow's matching criteria. |

## Zero Value For Material Price Fields

Settings: Allow, Disallow, Warning (default)

Controls the use of "0" as a value for the Material Price field.

## Zero Value For Material Price Adder Fields

Settings: Allow, Disallow, Warning (default)

Controls the use of "0" as a value for the Material Price Adder field.

# Zero Value For Non-Material Price Fields

Settings: Allow, Disallow, Warning (default)

Controls the use of “0” as a value for the Non-Material Price field.

# Chapter 20

# Viewer and Files

**This chapter includes the following:**


- Introduction ..... 265
- Viewer & Files Properties ..... 265
- Options for Purging and Retaining Files ..... 266
- File Association tab ..... 269
- Printing Banners and Watermarks ..... 271


## Introduction

You can double-click **Viewer and Files** under **System Settings** to open the Viewer & Files window. You can set viewing preferences for files that are attached to Agile PLM objects.

## Viewer & Files Properties

The following table lists the properties on the **General Information** tab of the Viewer & Files window; elaborations about some of the properties follow.

| Property | Description | Default |
|---|---|---|
| **AutoPurge Enabled** | Determines whether old versions of attachments should be deleted from the Agile File Manager. | No |
| **File Version Keep** | Dictates the number of versions to keep for archival purposes; used with AutoPurge Enabled set to Yes. Whether or not these versions include redlined versions depends on the settings of AutoPurge Enabled and Purge Redlined Files; see Note in text below. | 3 |
| **Purge Redlined Files** | Determines whether files with redlines are deleted, depending on the setting of AutoPurge Enabled. Overrides the File Version Keep preference. | No |
| **Purge Historical Reports Unaccessed (in days)** | Dictates the number of days that a Historical Report is not accessed before it is automatically purged. | 30 |
| **Directly Use These Thumbnail Image Types** | This setting provides a list of file types (listed as the extension names) that will be directly converted to thumbnails. | bmp, gif, jfif, jpe, jpeg, jpg, and png<br>These are the default image types and the user should not remove or add |

| Property | Description | Default |
|---|---|---|
| | | to these. |
| **CAD Revision BOM Attribute** | This field lets the administrator enable a text field in the Documents class, which permits assigning Revision letters (A, B, etc.) to CAD modeling BOMs (structured BOMs). The process to set up this field is described in Structured File Types and CAD Revision BOM Attribute on page 268. | — |
| **Upload File Size Limit (in MB)** | Defines the maximum size (in MB) of uploaded files for Java Client, Web Client, Portal Client, and SDK.<br>The default value, 0, indicates "No Limit"; any non-negative integer sets the size limit in MB. Users get error message in Java & Web clients if a file is too large.<br>Files that successfully upload (before a failure occurs) remain uploaded, other files (after a failure occurs) will not be uploaded. | 0 (= No limit to file size) |

# Options for Purging and Retaining Files

The following settings determine the circumstances under which files are purged or retained on the Agile PLM system and on users' local computers. Purging and retaining take place only during checkout and checkin routines.

## AutoPurge Enabled

Old versions of attachments can be automatically deleted from the Agile File Manager when new versions are checked in; the system does this after the checkout or checkin routine. If **AutoPurge Enabled** is set to Yes, old versions are deleted. To specify how many versions of the files you want to keep, use the **File Version Keep** field. See File Version Keep on page 267 and Purge Redlined Files on page 267 below.

If **AutoPurge Enabled** is set to No, old versions of attachments are kept on the Agile PLM system.

| | |
|---|---|
| **Caution** | There is no operation to restore an autopurged file. AutoPurge is *not* an archive function. |

**To enable or disable AutoPurge:**

1. Under **System Settings,** double-click **Viewer & Files**. The Viewer & Files window appears.
2. In the **Auto Purge Enabled** dropdown list, select No (disabled) or Yes (enabled).
3. When you are finished, click **Save**.

# File Version Keep

Use this property to specify how many old versions of the files you wish to keep. For example, if the value is set to 3, then the three most recent versions of the file are archived. (The latest version is also kept.) Use this property when **Auto Purge Enabled** is set to Yes; if **Auto Purge Enabled** is set to No, all versions are kept. See AutoPurge Enabled on page 266 above and Purge Redlined Files on page 267.

**To specify the number of attachment versions to keep:**

1. Under **System Settings,** double-click **Viewer & Files**. The Viewer & Files window appears.
2. In the **File Version Keep** field, type a number. The number indicates how many attachment versions you want to keep, in addition to the current version.
3. When you are finished, click **Save**.

# Purge Redlined Files

The **Purge Redlined File** field determines whether redlined files are automatically deleted along with files without redlines, depending on the setting of **Auto Purge Enabled**.

With **Auto Purge Enabled** set to Yes and **Purge Redlined File** set to Yes, the value of **File Version Keep** is the number of versions that will be kept. All earlier redlined files are purged. With **Auto Purge Enabled** set to Yes and **Purge Redlined File** set to No, the value of **File Version Keep** is the number of versions (either redlined or non-redlined) that will be kept. All earlier redline files are also kept.

See AutoPurge Enabled on page 266 and Purge Redlined Files on page 267.

**To set the** Purge Redlined File **property:**

1. Under **System Settings,** double-click **Viewer & Files**. The Viewer & Files window appears.
2. In the **Purge Redlined File** drop-down list, select No (disabled) or Yes (enabled).
3. When you are finished, click **Save**.

# Directly Use These Thumbnail Image Types

This setting provides a list of file types (listed as the extension names) that will be directly converted to thumbnails in Web Client without requiring the use of the view server because the conversion is done directly by the browser and occurs immediately when the file is attached. This setting also is a list of file types that can be used for the Replace function of manually specifying an image file to use for the thumbnail. The administrator can add supported image file types to this list. Be sure to have only one file type (entered in as an extension name) per line.

The default list of file types included in this field are:

bmp

gif

jfif

jpe

jpeg

jpg

png

For more information regarding Thumbnails, see *Getting Started with Agile PLM,* Configuring a File Manager on page 352, and Preferences Defined on page 371.

**Important** Thumbnail images display in Web Client only.

## Thumbnail generation options

The file types listed in **Directly Use These Thumbnail Image Types** need to also be checked in the **Auto Generate Thumbnail** column of the **File Association** tab in order to be successfully rendered in the browser. It is possible to have no file types listed in **Directly Use These Thumbnail Image Types** and only have the file types checked in the **Auto Generate Thumbnail** column and successfully have those types rendered in the browser, but it takes much longer because it is using the view server.

**Note** It is necessary to restart the file server to successfully render these images once the settings have been changed.

For more information see Specifying Supported File Types for the Agile Viewer.

# CAD Revision BOM Attribute

This value identifies the attribute that is configured as part of Agile's EC CAD Connectors to hold the "As Saved" revision.

**To set the CAD Revision BOM Attribute:**

1. Double-click **Viewer & Files**.
2. From the CAD Revision BOM Attribute dropdown list, select the BOM attribute that you previously configured for holding the CAD Revision, as part of your EC installation. Click **OK**, then click **Save**.

If you wish to use the CAD Revision BOM Attribute to provide fixed BOM resolution for the viewer even though you are not using an Agile EC CAD Connector, you must manually store the current value of the Revision field, for each object in the BOM, in the BOM attribute you designate for "CAD Revision".

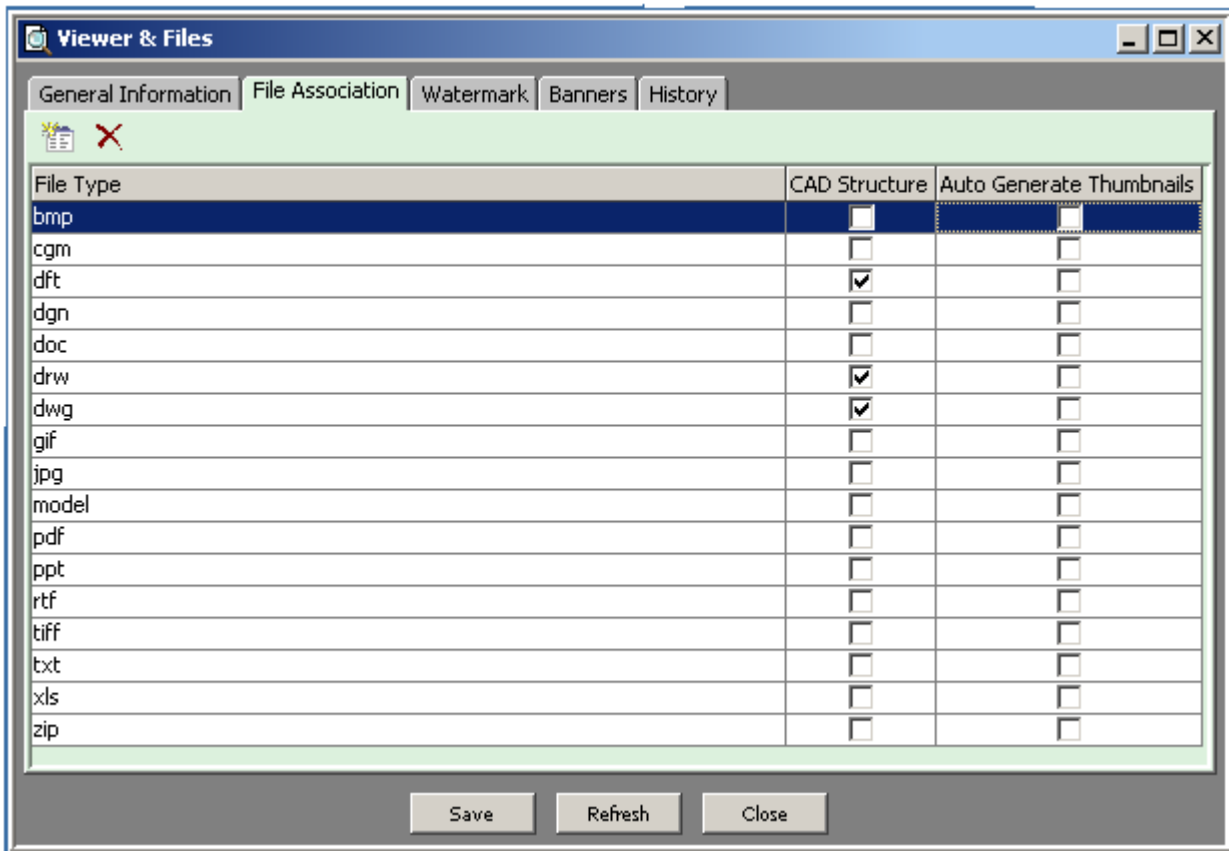
**Note** If CAD Revision BOM Attribute field is left "blank," the system uses standard Agile BOM resolution to determine how to traverse the structure to find files. Standard resolution is considered "floating", in that child objects can "rev" forward even after the parent object is released.

# File Association tab

## Specifying Supported File Types for AutoVue for Agile

In the **File Type** list on the **File Association** tab, you can select the types of files that users can view with the AutoVue for Agile. By selecting CAD Structure support then that particular file type will have the Viewer will render the structure. By selecting Auto Generate Thumbnails then that particular file type will have the Viewer render a thumbnail image in Agile.

**Note** The user needs to be assigned the ViewFile privilege to view attachments.




**Note** All of the Auto Generate Thumbnails check-boxes are un-checked by default.

For more information regarding Thumbnails, see *Getting Started with Agile PLM,* Configuring a File Manager on page 352, Directly Use These Thumbnail Image Types on page 267 and Preferences Defined on page 371.

The “2D” file formats in the following table are automatically listed in the **File Type** list in Administrator. Current customers may also have additional file types listed after upgrading.

| File extension | File type |
|---|---|
| BMP | Microsoft bitmap |
| CAL | CALS Group IV |
| CGM | Computer Graphics Metafile |
| DGN | Intergraph |
| DOC | Microsoft Word |
| DWG | AutoCAD Drawing File v 14, 2000 |
| DXF | AutoCAD Drawing Exchange Format |
| GIF | Graphic Interchange Format |
| GP4 | a neutral 2D viewable format |
| JPG, JPEG | Joint Photographic Experts Group |
| PCX | Microsoft Paintbrush |
| PDF | Adobe Portable Document Format |
| PLT | HPGL & HPGL 2 |
| PPT | Microsoft PowerPoint |
| PS | Adobe Postscript |
| SLDDRW | SolidWorks 2D |
| TIF, TIFF | Tagged Image File Format |
| TXT | Text |
| XLS | Microsoft Excel |
| ZIP | Zip |

AutoVue for Agile is the integration of Oracle's AutoVue visualization tool with Agile PLM. For an updated list of file formats supported by AutoVue, see http://www.oracle.com/technology/documentation/autovue.html.

File types are listed by extension. The **View** button is enabled by default, but if a user selects an attached file that has an unsupported filename extensions, the system will throw an error.

**Caution** If you are the administrator of Agile PLM on a Chinese, Hebrew, or Swedish operating system, Agile recommends that you remove the Supported File Types property of all file-type extensions. This is because the AutoVue for Agile is not supported on those operating systems. For detailed information about using Agile PLM with international operating systems, contact the Agile support Web site.

**To add or delete a file to the Supported File Types list:**

1. Go to **System Settings > Viewer & Files > File Association** tab.

2. To delete a file type from the **File Type** list, select that line and click **Delete**.
3. To add a file type to the **File Type** list, click the **New** button, and then enter its name on the new line. You can also use Ctrl-C and Ctrl-V to copy and paste a list from one of the other file type fields or from another application.
4. When you are finished, click **Save**.

# Printing Banners and Watermarks

The Viewer & Files window includes a **Watermark** tab and a **Banners** tab. The settings on these tabs determine whether banners and watermarks appear on printouts of files attached to Agile PLM business objects, and the content of those banners and watermarks. These banners and watermarks appear on files printed with the AutoVue for Agile.

The only way to have no print appear is to make sure the Banner and Banner Text pairs (and Attribute Watermark/Attribute Watermark Text) are *both* blank: delete the text in the appropriate text field. For example, if you make Top Left Banner blank, you will then see an unwanted “Printed by:” because this was not deleted from Top Left Banner Text.

The following table lists the watermark properties; these can be set on the **Watermarks** tab.

| Property | Description | Default Text or Setting |
|---|---|---|
| **Attribute Watermark** | The value of this attribute appears in the watermark as text after the Attribute Watermark Text. | [*Select from dropdown list. This becomes* $ATTRIBUTENAME:] |
| **Attribute Watermark Text** | Text that appears before the text of the value in the Attribute Watermark. | [*None: enter your own text, including* colon *if appropriate*.] |
| **Back Rev Watermark Text** | Watermark text that appears if the item object is not the latest revision. This property is available only for the Parts and Documents Banner & Watermark attributes. | Back Revision |
| **Name** | Lists all the Agile PLM business classes for selection | [*one for each business class*] |
| **Pending Rev Watermark Text** | Watermark text that appears if the Item object is a pending revision. This property is available only for the Parts and Documents Banner & Watermark attributes. | Pending Revision |
| **Print Watermark** | Turns printed watermarking on (Yes) and off (No). | No |

The following table lists the banner properties; these can be set on the **Banners** tab.

| Property | Description | Default Text or Setting |
|---|---|---|
| **Name** | Lists all the Agile PLM business classes | [one for each business class] |

| Property | Description | Default Text or Setting |
|---|---|---|
| | for selection | |
| **Bottom Left Banner** | Banner appears at the bottom of the printout, on the left, after the Bottom Left Banner Text. | [object number] |
| **Bottom Left Banner Text** | Text that appears before the text that is the value of the Bottom Left Banner. | $ATTRIBUTENAME: |
| **Bottom Middle Banner** | Banner appears at the bottom of the printout, in the middle, after the Bottom Middle Banner Text. | *Item*: "Rev [*letter*] of Latest Released Revision"<br>Empty for other classes |
| **Bottom Middle Banner Text** | Text that appears before the text that is the value of the Bottom Middle Banner. | *Item*: "Rev:"<br>Empty for other classes |
| **Bottom Right Banner** | Banner appears at the bottom of the printout, on the right, after the Bottom Right Banner Text. | [Status for workflow class; Lifecycle Phase for non-workflow class] |
| **Bottom Right Banner Text** | Text that appears before the text that is the value of the Bottom Right Banner. | "Status:" [or]<br>"Lifecycle phase:" |
| **Top Left Banner** | Banner appears at the top of the printout, on the left, after the Top Left Banner Text. | [*Print User*] |
| **Top Left Banner Text** | Text that appears before the text that is the value of the Top Left Banner. | "Printed By:" |
| **Top Middle Banner** | Banner appears at the top of the printout, in the middle, after the Top Middle Banner Text. | — |
| **Top Middle Banner Text** | Text that appears before the text that is the value of the Top Middle Banner. | — |
| **Top Right Banner** | Banner appears at the top of the printout, on the right, after the Top Right Banner Text. | [*Print Date*] |
| **Top Right Banner Text** | Text that appears before the text that is the value of the Top Right Banner. | "Printed On:" |

Here are a few points about the watermark and banner settings:

- All the properties except Print Watermark can be filled with your own text.
- For any specific attachment, only one setting is available per banner or watermark property.
- You cannot create multiline banners.
- The variable "$ATTRIBUTENAME:" inserts the name of the attribute that is selected in the Attribute Watermark property or the banner properties.

- *Within* the Banner Text and Attribute Watermark Text properties, it is possible to enter your own text *before* the "$ATTRIBUTENAME:" variable, but not after.

# Banner and Watermark Properties

When you set values that specify text to appear in printed banners and watermarks, you select from a dropdown list of variables and attributes (fields that appear in Agile PLM clients). The following table lists and describes the choices.

| Variable in dropdown list for property | Default printed value and restriction |
|---|---|
| **Effective Date – Obsolete Date** | *Effective from: [effective date] to: [obsolete date]*<br>Available for items only: Effective Date and Obsolete Date are derived from the **Affected Items** tab of the change that created the specific rev of the item. |
| **Page Number** | Page *[page number]*<br>Also specifies the region of an image when a specific region is printed. |
| **Page Number and Page Count** | Page *[page number]* of *[total pages]*<br>Also specifies the region of an image when a specific region is printed. |
| **Print Date** | *[print date]* |
| **Print User** | *[user]* |
| **Rev of Latest Released Revision** | *[rev letter of latest released revision]*<br>Available for items only. |
| **Watermark Text** | The value specified for the Watermark Attribute property, regardless of whether that property is enabled.<br>Available as a variable for banners only. |
| **Attribute** | *[attribute name].[attribute value]*<br>Many selections: Attachments.*[value], Change History.[value], Page Two.[value], Title Block.[value], Cover Page.[value], General Info.[value]* |

**To print watermarking and specify watermark text:**

1. Under **System Settings,** double-click **Viewer & Files**. The Viewer & Files window appears.
2. Click the **Watermark** tab to bring it forward.
3. Double-click the watermark you want, and update the property setting for the class.
4. Select Yes in the **Print Watermark** dropdown list to enable watermark printing.
5. To select watermark text, in the **Attribute Watermark** the dropdown list, select an attribute. The value of this attribute will appear as watermark text when **Print Watermark** is set to Yes.
6. To specify custom text for back revisions and pending revisions, in the **Back Rev Watermark Text** or the **Pending Rev Watermark Text** field, type the text that you want to appear in the watermark or accept the default text.
7. When you are finished modifying the **Watermark** tab, click **Save**.

# Specifying Banner Text and Location

**To specify a banner to print on Agile PLM object printouts:**

1. Under **System Settings,** double-click **Viewer & Files**. The Viewer & Files window appears.
2. Click the **Banners** tab to bring it forward.
3. For each banner you want to appear (Bottom Left, Middle, or Right; or Top Left, Middle, or Right), select an attribute or variable from the drop-down list. The value of this attribute or variable will appear in the banner.
4. If you want to remove the default text or enter new text in the banner, place your cursor in the corresponding text field of the banner, for instance, **Bottom Left Banner Text**, and delete or alter the text. Since this text appears *before* the value in **Bottom Left Banner** property, you may end this text with a colon.
5. To change a text property, in the corresponding text field of the banner, for instance, **Bottom Left Banner Text**, delete or alter the text, type the new text or variable in its place.
6. When you are finished modifying the **Banners** tab, click **Save**.

# Chapter 21

# Notifications

**This chapter includes the following:**


- Overview of Notifications ..... 275
- Default Notifications ..... 276
- Event-based Notifications ..... 277
- Working with Default and Event-based Notifications ..... 281


# Overview of Notifications

Agile PLM has the capability to send notifications to users, either when the user is required to take action, or to notify the user that various actions have taken place.

With the introduction of the **Event Management** framework, there are now two types of notifications in PLM. The **Notifications** node stores both types:

- *Default notifications* can be modified but only in a limited way; for this reason they are sometimes referred to as *notification templates*.
- *Event-based notifications* are admin-*user–created* and permit much more flexibility and creativity in how and when notifications are sent by the system.

The default notifications are triggered by the system based on system- or end-user–generated actions that have been captured in the notification templates. Default notifications have an uneditable Object Type property. They cannot be used by Event Subscriptions, that is, unless you perform SaveAs on the default notification and proceed to configure a Notification mask.

Event-based notifications can be distinguished by their configurable Object Type property. They can be triggered by Event Subscriptions that are configured in the **Event Subscribers** node.

## Preliminary Admin Settings for Notifications

It is necessary for the administrator to enable a database setting before *any* notifications are sent. In **Server Settings** > **Database**, ensure that the setting of the **Notification Enabled** property is Yes. If this property is set to No, PLM does not send notification emails or messages to the users in the system.

An *individual* notification is sent only if it has been enabled by setting its **Enabled** attribute to Yes. Depending on the **Notification Type** property setting for the notification, the notification appears in the user's Inbox, is sent by email, or both. The **Enabled** and **Notification Type** attributes can be set as needed for each notification.

A user can receive notifications with links back to Java Client or Web Client; more explicitly, the User preference Preferred Client (under **User Settings > Users >** <*any user*> > **Preferences** tab) determines which client is automatically opened to the object to which the notification's link points. Since Agile users working in the PPM, PCM, and PG&C solutions operate only in Web Client, it is

recommended that the Preferred Client be set to Web Client for those users.

To fully configure email notification in Agile PLM Clients, please see Configuring Web Client Notification on page 347 and Configuring Java Client Notification on page 348.

# Default Notifications

Default notifications are listed in the **System Settings** > **Notifications** node. Default notifications are not and cannot be associated with any events configured by the administrator in the **Event Management** node; however, a Default notification can be saved to create a new notification that can be associated with an Event Subscription. See Creating a Notification Using SaveAs on page 282 for more information.

## Attributes of Default Notifications

| Default notification attribute | Description |
|---|---|
| **Name** | Name of notification; not editable |
| **API Name** | API Name is non-editable in Default notifications |
| **Object Type** | Object Type is non-editable in Default notifications |
| **Enabled** | Yes = the notification is enabled, so it can be sent (by the system).<br>No = the notification is disabled, it cannot be sent. |
| **Priority** | Default is Regular; list includes High and Low |
| **From** | Select a user from the list who will be listed in the "From:" field of the notification. Only one user can be selected for the From attribute.<br>The defaults are variables: $AGILE is always present on the dropdown list; $ORIGINATOR and $SENDER are often present; on routable objects, the appropriate routing manager variable is present (for example, $CHANGE ANALYST). |
| **To** | To: is always empty on Default notifications. |
| **Notification Type** | Select Email, Email and Inbox, or Inbox from the dropdown list. |
| **Subject** (and **Data**) | By including Data tags when you compose the subject line, you can include references specific to the triggering event, such as Change Number and Originator.<br>See Entering Subject and Body Text on page 281. |
| **Body** (and **Data**) | By including Data tags when you compose the body text, you can include references specific to the triggering event, such as Change Number and Originator.<br>See Entering Subject and Body Text on page 281. |

$NOTIFY picks up only the default Notify users from manual routing. In this case, when Change Status is brought about by autopromotion (for example, when a Relationship is triggered), the default Notify users are not picked up. This is a difference between how Default notifications and Event-based notifications process change status.

# Event-based Notifications

Event-based (or "admin-created" or "user-defined") notifications are created in the **Notifications** node. A notification created by the administrator has much more flexibility than the default notifications. Event-based notifications support **Page Three** attributes (if any) for subclass notification.

Event-based notifications are the Handler in those Event Subscriptions that are dedicated to notifications. (See Event Management on page 301 and Handler Types on page 307.) For an example of a Notification mask configured into an Event Subscription, see Notify when Affected Items Table is Modified on page 345.

Notifications can also be called by the Agile SDK and a Script PX handler. (See Invoking Notifications Programmatically on page 280) For an example of generating a notification through the SDK, see Notify Create User when Item is Incorporated on page 340.

When a notification is triggered in the application, a Scheduled Notification is tracked in the **History** tab of the object. It shows the triggered Event and Notification as well as the $NOTIFY users that may receive the notification.

In an admin-created notification, $NOTIFY picks up the "Notify" users entered by end-user in the Notify field in routing slips, Signoff dialog box, and Comment dialog box; it also picks up the "Checkbox" users in Signoff dialog box and Comment dialog box.

In addition, if the Default notification is used for Reminders, it picks up the Approvers to be reminded; if the Default notification is used for Escalations, it picks up the Escalation persons of the Approvers.

To find a list of only Event-based notifications that are created by the administrator go to **System Settings** > **Event Management** > **Event Handler Type** > **Notifications** > **Where Used** tab.

**Note** Unlike Default notifications, admin-created notifications can be deleted. See Deleting a Notification on page 280.

**Important** For notification-specific Event Subscriptions, the object type of the Event mask must be appropriate for the object type of the Notification mask. Either the notification applies to all objects of the Object Type (subclass), or, the notification is derived from the Object Type (base class and class).

For example, if the Notification mask specifies Change Orders (a class), the Event mask must specify the class or ECO (subclass) or any other subclass derived from the Change Order class.

# Attributes of Event-based Notifications

| Event-based notification attribute | Description |
|---|---|
| ***General Information tab attributes*** | |
| **Name** | Name of Notification mask |
| **API Name** | Name of the object that is used by external processes to identify the object. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name once you have specified the Name of the Notification mask. See API Name on page 19. |
| **Object Type** | The object type is used to limit the notification to actions on objects of the type specified. This required field is selected from a dropdown list of Base Class, Class, and Subclass options.<br>The object type selected also dictates the available From options and available attributes (**Cover Page, Page One, Page Two, or Page Three**) to be added as tags for the Subject and Body fields in the notification. It also dictates some of the variables available in the "To" options (for example, $CHANGEANALYST).<br>Object Type can be changed in the Create Notification (Mask) dialog for user-defined notifications. Object Type is non-editable once the Notification mask has been created.<br>(See "Important" note just above this table.) |
| **Enabled** | Yes = the notification is enabled, so it can be sent (by Event Subscription or even programmatically, i.e., by SDK).<br>No = this notification is disabled, it cannot be sent. |
| **Priority** | Default is Regular; list includes High and Low. |
| **From** | Select a user from the list who will be listed in the “From:” field of the notification. Only one user can be selected for the **From** attribute.<br>The defaults are variables: $AGILE is always present on the dropdown list; $ORIGINATOR and $SENDER are often present; on routable objects, the appropriate routing manager variable is present (for example, $CHANGE ANALYST). |
| **To** | Select a user from the list who will be listed in the "To:" field and receive the notification. The variable $NOTIFY is available, as well as specific recipient users from the Address Book.<br>**Note** If the object type is a routable object then the available list of recipients will also include workflow users. The list of users may include users that will not receive notifications because of lack of privileges. |
| **Notification Type** | Select Email, Email and Inbox, or Inbox from the dropdown list. |

| Event-based notification attribute | Description |
|---|---|
| **Subject** (and **Data**) | You may compose a subject line for the notification.<br>By including Data tags when you compose the subject line, you can include references specific to the triggering event, such as Change Number and Originator.<br>See Entering Subject and Body Text on page 281. |
| **Body** (and **Data**) | You may compose body text for the notification.<br>By including Data tags when you compose the body text, you can include references specific to the triggering event, such as Change Number and Originator.<br>See Entering Subject and Body Text on page 281. |
| ***Where Used tab attributes*** | |
| **Name** | Names of Event Subscribers (masks) that point to the notification |
| **Description** | Description of Subscriber masks that point to the notification |
| **Enabled** | The setting of Enabled for the Subscriber masks |
| **Event** | The Event mask that is associated with the Subscriber masks |
| **Event Handler** | The current notification's name. This is always the Notification currently opened because the Notification functions as the Handler for the associated Subscriber masks. |
| **Trigger Type** | The trigger type for a notification is always Post. See Trigger Type: Pre and Post on page 320. |
| **Execution Mode** | The execution mode for a notification is always Asynchronous. See Execution Mode on page 321. |
| **Order** | Disabled; the column does not display data because the property does not apply to notifications. |
| **Error Handling Rule** | Disabled; the column does not display data because the property does not apply to notifications. |

# Working with Events-based Notifications

The following tasks apply to Events-based notifications, not Default notifications.

## Creating a Notification

This is the procedure to create a Notification mask. It can then be named by a Subscriber mask that binds it to an Event mask to become a complete Events-based notification.

**To create a Notification mask:**

1. Under **System Settings,** double-click **Notifications**. The Notifications window appears.
2. Click **New**.
3. Enter a name in the **Name** field.
4. Enter a name or click the **Autonumber** button to generate the required API name.
5. For the required **Object Type** field, choose a Base Class, Class, or Subclass type of object.
6. Click **Next**.
7. Set the **Enabled** field to Yes or No.
8. Set **Priority** to High, Low, or Regular.
9. In the **From** field select from the dropdown list of the available values for the object type selected, such as $AGILE or $CHANGEANALYST, depending on Base Class, Class, or Subclass.
10. In the required **To** field select from the Address Book of Users and User Groups who the notification will be sent to.
11. Set the **Notification Type** as Email, Email and Inbox, or Inbox only.
12. To insert a data tag (or data variable) into the **Subject** or **Body** field, click **Add Data Tag**. For more information, see Entering Subject and Body Text on page 281.
13. Click **Finish**.

## Deleting a Notification

Notification masks can be deleted only if they are not used by Subscribers. Default notifications cannot be deleted; they can be enabled and disabled.

**To delete a user-defined notification:**

1. Under **System Settings,** double-click **Notifications**. The Notifications window appears.
2. Use the filter to find the specific notification you want to delete.
3. Select the notification and click **Delete**.

   You can also double-click the notification and then click **Delete**. By doing so, you can look through the settings and properties to be sure you want to delete the notification.

**Note** If you delete a Notification mask that has already sent notifications to users, then the subject of the notifications in the users' inboxes will change to appear as null, reflecting that the notification is deleted.

# Invoking Notifications Programmatically

SDK and Scripting allows notifications to be sent out programmatically. A notification invocation uses the following parameters, whose use is explained:

- **SendNotification**: the program module or method used to send out a notification.

- **Notification Name**: the name of the notification.
- **Urgent**: whether the notification is to be flagged as Urgent or not; the value can be True or False.
- **Addressee List**: list of users who may be notified; this set of values is appended to the list of people listed in the "To" field of the notification. This list of Addressees is only picked up if $NOTIFY is assigned in the "To" field.
- **Comment**: refers to the comment passed to the Comment data tag, if assigned in the notification template.
- **Object**: an object of the type corresponding to that associated with the notification in its definition. For example, an ECO object needs to be passed to a notification tied to a notification associated with the ECO subclass (it can also be passed to a notification tied to the Change Orders class or Changes base class). This object is used to derive the values of variables used in the notification Subject, Body, and To fields.

For more information about sending notifications using Agile SDK, see Chapter 9, "Subscribing to Agile PLM Objects" in the *Oracle Agile PLM SDK Developer's Guide*.

# Working with Default and Event-based Notifications

The following tasks apply to both Default notifications and Events-based notifications.

## Editing a Notification

**To edit a notification:**

1. Under **System Settings,** double-click **Notifications**. The Notifications window appears.
2. Filter the notifications by Name or Object Type.
3. Double-click the name of the notification you want to edit. Its window appears.
4. Edit the attributes in the appropriate manner: select from a dropdown list or enter text. Refer to the table found above.
5. To insert a data tag (or data variable) into the **Subject** or **Body** field, click the **Add Data Tag** button. For more information, see Entering Subject and Body Text on page 281.
6. Click **Save**.

## Entering Subject and Body Text

Subject and Body fields are required to be populated. Each Default notification includes text for the subject and body of the message. You can use the default text, or you can enter a text message that you want to use.

During the editing or creation of a notification, when you enter the text you want to appear in the subject line and body of the notification, you can add data tags that refer to pertinent information for that notification.

The available tags are dictated by the object type associated with the notification.

For example, for an ECO Status Promotion, Approvers and Change Analyst notification, you might want the subject line to say:

> [Cover Page.Change Type] [Cover Page.Number] has been moved from [From Status] to the [To Status] of [Cover Page.Workflow] workflow.

**To include the From Status in the subject line:**

1. Click to position the cursor in the appropriate place in the subject message.
2. Select From Status in the **Data** field dropdown next to the **Subject** field.
3. Click Add **Data Tag**. The data tag [From Status] appears in the **Subject** field.
4. As you continue to enter text in the subject line, you can add more data tags.

Similarly, use the **Data** field next to the **Body** field to add data tags to the body text.

# Creating a Notification using SaveAs

From existing notifications you can perform a SaveAs function to create a new notification with a new name and configure the values according to your needs.

**Important** All Default notification templates that are not bound to an object type (for example, Transfer Authority, Create, or Modify) as well as those for Product Cost Management cannot use the SaveAs feature.

When you create a notification using SaveAs, the object type of the original notification cannot be changed for the new one. The new notification is automatically associated with the object type of the original notification.

This is the only way a Default notification can be recast as a Notification mask and be used as an Event-based notification.

Also, if the source notification is a Default notification, then the **To** field in the new notification is unpopulated and can now be selected.

**To create a new notification using SaveAs:**

1. From an open notification click **SaveAs**.
2. Enter a new name and autogenerate the API Name.
3. On the **General Information** tab enter a To recipient.
4. Click **Save**.

Use the instructions in "Editing the Notification" to tailor the new notification as desired.

# Chapter 22

# Full Text Search

**This chapter includes the following:**


- Full Text Search Limitations ..... 283
- Full Text Search Properties ..... 284
- Indexing Attachment Files ..... 285


From the **Full Text Search** node, you specify how the search is done and manage the indexing of attachment files. The information on indexing in this chapter applies only to indexing files. Indexing of database information for Web searches is always done synchronously.

When a user searches for attachment file content, the search results list each object that contains files matching the search criteria. Only the latest versions of attachments are listed. For more information about setting up searches and finding data in Agile PLM, see *Getting Started with Agile PLM*.

| | |
|---|---|
| **Note** | Regarding the Agile multi-language capability, Full Text Search is supported for attachment files in Japanese, Traditional Chinese, Simplified Chinese, German, and French localized versions. |

# Full Text Search Limitations

Full Text Search has the following known limitations:

- Full Text Search follows existing roles and privileges. Users who do not have privileges for objects do not see these objects in search results.
- Users who do not have Field-level Read for fields do not see these fields in the search results.
- Full Text Search does not support searches for the following:
  - Synonyms
  - URL attachments
  - Viewer (Cimmetry) redline data
- For non-English text in documents, Full Text Search is supported only for attachment files, not for Stem searches or Concept searches.
- Attachment files that can be returned by a full text search must be created and saved in DOC, PPT, PDF, TXT, HTML, XLS, XML or RTF.
- Files on any Agile File Management file vault are available to Full Text Search, and are indexed based on the Administrator settings.
- If an object contains multiple files (that is, multiple documents are attached to the object), even if only one of the files contains the search text, the entire object is returned. The file that contains the text is not indicated, and you may need to look through all attachment files to find

the appropriate one.

# Full Text Search Properties

When you double-click the **Full Text Search** node, the Full Text Search window appears, with **General Information** and **History** tabs. There is more information about the settings for indexing attachment files in the next section.

| Property | Editable ? | Default | Notes |
|---|---|---|---|
| **Stem Searching** | Yes | Disabled | Finds variations of a word. From "run" it could find *ran*, *runs*, *running*. |
| **Concept Searching** | Yes | Disabled | Finds words that fit the same general concept. From "sports" it could find *running, track*, *basketball*, and the like.<br>**Important:** Concept searching is not supported on Microsoft SQL Server. |
| **Indexing** | Yes | Synchronous | Indicates when words in a file will be available for full text search:<br>Manual – indexes files when a user clicks the **Index Attachments** button only.<br>Scheduled – indexes files according to the specified schedule—see next table.<br>Synchronous – indexes files as soon as a file is added.<br>See Indexing Attachment Files on page 285. |
| **Index File Type** | Yes | doc, ppt, pdf, txt, html, rtf, xml, xls | Makes possible selective indexing of specified files. A comma separates each file type, which are not case-sensitive. The default values cover a range of common file types. If you narrow the field to include one or two file types for a specific search, be sure to re-populate the field with your preferred file types upon completion of the search. |
| **Last Indexed** | No | N/A | Date the system was last indexed. |
| **Next Scheduled Index** | No | System-generated if scheduled indexing is running | Indicates the time the next scheduled index will occur. Only appropriate if indexing is "Scheduled." |
| **Recurrence** | Yes | N/A | Indicates how frequently indexing occurs. Activated when indexing is "Scheduled." |

The following additional attributes appear when you click the Recurrence field:

| Property | Editable? | Default | Notes |
|---|---|---|---|
| **[Once a Day] or [Every X Hours Y Minutes]** | Yes | No | Specifies the frequency that indexing occurs. |
| **Starting At** | Yes | 0:00:00 am | Indicates the starting time of the schedule |
| **Ending At** | Yes | 0:00:00 am | Indicates the ending time of the schedule. This option will be disabled if recurrence = Once A Day. |
| **Days** | Yes | M–F | Indicates which days the schedule will take effect |

# Indexing Attachment Files

There are some best practice recommendations for setting the system for indexing attachment files.

## Using FileLoad and Indexing New Files

If the Indexing property is set to Synchronous and a large number of files are added to the File Management file vault using Agile FileLoad, the indexing process can take a long time. It has been found that setting this field to Manual increases the loading speed for a large number of files. Or, if the files being added tend to be large documents, set the Indexing property to Scheduled and set up an indexing schedule for once every 3 or 4 hours. If most of the new files are relatively small, the Indexing property can be set to Synchronous.

## Index Attachment Button

Use the **Index Attachment** button on the toolbar (**Admin > System Settings > Full Text Search**) to manually index new attachments that are not yet indexed. The **Index Attachment** button is available only when indexing is set to Manual.

## "Stop" Words

There are short, common words called “stop” words that are ignored by the system in searching against attachment files. So, a search for “the mouse” will return only matches for “mouse”. You don’t need to know the words unless you run a lot of searches; the list of words is in “Using Quick Search to Find Attachment File Content” in the Searches chapter in *Getting Started with Agile PLM*.

# Chapter 23

# My Assignments

**This chapter includes the following:**


- Overview ..... 287
- Columns ..... 287


## Overview

This node configures some aspects of what your users see when they click the **My Assignments** tab in Web Client's home page. (If a user's Preferred Inbox View user preference is set to My Assignments, it will already be displayed when that user opens Web Client.) You can set the order of the columns, and you can rename the column names; the default columns are called Name, Status, Due Date, % Complete, Related To, Actual Hours, and Flag.

## Columns

**To change the order of the columns:**

1. Under **System Settings**, double-click **My Assignments**. The Configure My Assignment Table opens.
2. Click the **Order Columns** icon. The Order dialog opens.
3. Select a column name and use the **Up** or **Down** buttons to move that name in the list. ("Top-to-bottom" in the list is equivalent to "left-to-right" in the user interface.)
4. When you are finished, Click **OK**. The dialog closes and you see the numbers in the Order column reflect the new order. A user will have to open or re-start Web Client to see the columns in the revised order.

**To change the name of a column:**

1. Under **System Settings**, double-click **My Assignments**. The Configure My Assignment Table opens.
2. Click in the field of the Column Name that you want to change. The field becomes editable.
3. Enter the new name. You can change other column names in this procedure.
4. When you are finished, click **Save**. A user will have to open or re-start Web Client to see the new column names.

**To create a new column:**

1. Under **System Settings**, double-click **My Assignments**. The Configure My Assignment Table opens.
2. Click **New**. A new row (representing the eventual column in Web Client user interface), is created.
3. Enter a name in the Column Name field.
4. In the Column Data Source field, use the dropdown dialog to select a data source for the

column. You can choose a Class – Activities or Gates – and enter a Find value, using the **Down** or **Up** buttons to "find next" or "find previous," respectively. When you have selected a data source, Click **OK**.

5. Use the **Order Columns** icon to adjust the order of the new column with the others (and to activate the **Save** button). When finished, click **Save** to complete the procedure. A user will have to open or re-start Web Client to see the new column.

**To remove a column:**

1. Under **System Settings**, double-click **My Assignments**. The Configure My Assignment Table opens.
2. Select the row that you want to remove. (It is easiest to click in Data Type or Order column of the row.)
3. Click the **Remove Column** icon. The row will be removed.

| Note | Default Columns cannot be removed, they can only be made non-visible by unselecting the checkbox in the Visible column. |
|---|---|

4. When you are finished removing columns, click **Save**.

# Chapter 24

# Unit of Measure (UOM)

**This chapter includes the following:**


- Creating Categories of Measure and Units of Measure..... 289


**Note** If you do not see the **UOM** node under **System Settings**, modify the Administrator privilege so that the Applied To list includes **UOM**.

The **UOM** node allows you to define different units of measure for use in Agile PLM. Units of measure are important for Product Governance & Compliance (PG&C), where you must measure the weight or quantity of restricted substances contained in your products.

You can define a new measure, and then specify the units used for that measure. You can also specify units of measure for any existing measure defined in your Agile PLM system. For example, the Dram unit of measure could be added to the Weight measure. In avoirdupois weights, 16 drams equal 1 ounce.

Each measure has a standard unit. All unit of measure values are normalized to the standard unit using a conversion factor. If you define a new unit of measure, you must specify the conversion factor relative to the standard unit. For example, if the standard unit of the Weight measure is Gram, the conversion factor for Ounce is 31.1034.

## Creating Categories of Measure and Units of Measure

The following tasks describe how to create a new category of measure and new units of measure.

**To create a new category of measure:**

1. Under **System Settings**, double-click **UOM**. The Measures window opens.
2. Click the **New** button. The Create a New Measure dialog box appears, referring to a new category of measure, just as Weight and Quantity really are kinds or categories of measurement.
3. Enter the **Name**, **API Name**, **Description**, and **Range Match Factor** for your new category of measure.

   The Range Match Factor expands the range of search results for a particular unit of measure that you search for. Its purpose is to adjust for inexact conversions to the standard unit for a measure.
4. Click **OK**.

**To create a new unit of measure:**

1. Under **System Settings**, double-click **UOM**. The Measures window opens.
2. Double-click a category of measure in which your new UOM belongs. The Measure: <*Name*> window appears.

3. Click the **UOM** tab.
4. Click the **New** button. The Create a New Unit of Measure dialog box appears.
5. Enter values for fields described in the following table. When you have finished Click **OK**.

| Unit of Measure Attribute | Description |
|---|---|
| **Name** | Name of the unit of measure. |
| **API Name** | Unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. |
| **Abbreviation** | Standard abbreviation for the unit of measure. |
| **Conversion Factor** | Factor used to convert values to the standard unit. (See "Is Standard" description.) |
| **Is Exact Conversion** | Select **Yes** if the specified conversion is exact. |
| **Is Standard** | Select **Yes** if the unit is the standard for your company. Only one unit of measure can be the standard.<br>The first UOM that you create in a new category of measure must be the standard, whose conversion factor must be 1. All subsequent unit of measure values are normalized to the standard unit using the specified conversion factor.<br>**Note** Once a unit of measure is set as a standard and compliance related data is created in Agile, it is recommended to not change the standard UOM. Changing the UOM will affect searches and compliance calculations. |
| **Enabled** | Whether the unit of measure is enabled. Only enabled units of measure appear to the end-users of Agile PLM clients. |

# Chapter 25

# Other System Settings

**This chapter includes the following:**


- Company Profile ..... 291
- Currency Exchange Rates ..... 291


## Company Profile

The **Company Profile** node provides a single location for storing information about your company.

**To define the company profile:**

1. Under **System Settings**, double-click **Company Profile**. The Company Profile window opens.
2. Enter the name, address, phone and fax numbers, URL, corporate currency, and the display name.
3. Click **Save**.

## Currency Exchange Rates

From the **Currency Exchange Rates** node, you manage the Currency Exchange Rates table for in-system currency conversion. The list of available currencies is determined when Agile PLM is installed. For information about maintaining the list of available currencies, contact your Oracle Consulting – Agile Practice representative. The table in the Currency Exchange Rates window displays the currencies in use and their exchange rates for the corporate base currency.

| Note | Currency exchange rates affect costs in two solutions: Product Portfolio Management and Product Cost Management. |
|---|---|

Currency conversion rates are used to convert currency values to the selected corporate currency. For instructions on how to select the corporate currency, see Company Profile on page 291.

Currency conversion is not an automatic process. Your system can be configured to have money fields on **Page Two** or **Page Three** of items. If you change the currency for a money field (for example, from USD to GBP), the field value is not automatically recalculated for the new currency.

**To add a currency:**

1. Under **System Settings**, double-click **Currency Exchange Rates**. The Currency Exchange Rates window opens.
2. Click the **New** button. The Add Currency dialog box appears.
3. Select one or more currencies in the **Choices** list and use the right-arrow to move them to the **Selected** list.

An additional dialog box opens for entering conversion rates.

4. Type the appropriate values in all **Conversion Rate** fields. Enter rates as decimal values, for example, 1.04 or 0.96.
5. Click **OK**. The new currency exchange rates appear in the list.

**To modify the exchange rate for a currency:**

1. Under **System Settings**, double-click **Currency Exchange Rates**. The Currency Exchange Rates window opens.
2. Double-click a currency to update. The Currency window opens.
3. In the **Conversion Rate** field, type a decimal value.
4. Click **OK**. The new currency exchange rates appear in the list.

**Note** To import updated currency exchange rates from a Microsoft® Excel file or a delimited text file, choose **Tools > Import**. For more information*, see Getting Started* with *Agile PLM*.

**To view historical rates for a currency:**

1. Under **System Settings**, double-click **Currency Exchange Rates**. The Currency Exchange Rates window opens.
2. Double-click a currency. The currency window opens.
3. Click the **Historical Rates** tab. The **Historical Rates** tab displays a list of conversion rates for the currency by date. You cannot edit historical currency rates.
4. Click **Close**.

# Chapter 26

# Dashboard Management

**This chapter includes the following:**


- View Dashboard Tabs ..... 293
- Adding Dashboard Tabs ..... 293
- Deleting Dashboard Tabs ..... 293
- Adding Tables to Optional Tabs ..... 294
- Adding Charts to Optional Tabs ..... 294
- Adding Custom (URL) Process Extensions for Optional Tables ..... 297
- Reordering Table Rows in a Tab ..... 297
- Displaying Optional Tabs ..... 297


The Dashboard Tab View privilege is used to configure Dashboard tabs. For more information, see Dashboard Management on page 36.

## View Dashboard Tabs

To view dashboard tabs, click **System Settings > Dashboard Management** node. The list of default dashboard tabs displays. You can reorder this list using the icon, to change the order in which the tabs display in Web Client. To navigate to a tab, click on its name. You can edit the name and description of the tab, and specify whether the tab should be visible or not.

Optional tabs can be renamed and configured.

## Adding Dashboard Tabs

To add a new tab, click the icon in **Dashboard Management**. In the **Create Dashboard Tab** dialog, enter a Name, API Name and Description. In the **Visible** field, select Yes.

| | |
|---|---|
| **Note** | Although you can create an unlimited number of tabs, the tabs displayed in Web Client are restricted by the user's roles and privileges. To enable a user to view a particular dashboard tab, you must assign the Dashboard Tab View privilege to that user, and select the required tab in the **AppliedTo** property of that privilege. For further information, see AppliedTo Capability on page 228. |

## Deleting Dashboard Tabs

You can only delete dashboard tabs that you have created. Default tabs cannot be deleted. To delete a newly created Dashboard tab, within Dashboard Management, select the tab and click .

| | |
|---|---|
| **Note** | Once a tab has been accessed, you cannot delete it. Attempts to delete it will result in an error - "Object is in use." |

# Adding Tables to Optional Tabs

Once a new optional tab is created, tables need to be inserted into the tab. To add a table, double-click the new tab name in Dashboard Management and click the **Tables** tab. Click the **Create Dashboard Table** icon.

The table below lists the properties in the Create Dashboard Table for Table type.

| Property | Description | Possible Settings |
|---|---|---|
| Name | Enter the name of the table | |
| API Name | You can specify an<br>API name, but internal consistency is enhanced by accepting the system-generated<br>name. | |
| Description | Enter the description of the table | |
| View List Type | Lists the type of table type. Select Table | Chart, Table, Custom, Advanced Search |
| Dashboard Extension (DX) | Lists all the dashboard extensions created for Table type list. | |
| Visible | To enable in Web Client | Yes/ No |

## Adding Data to Tables

Double-click the new table, click on **Attributes** tab and click on the **Add an Attribute** + icon to create a new attribute. Agile currently supports Text, Numeric, Image, Date, Money, and Link type of table attributes.

**Note** Attributes need to be of a specific type which is supported by Agile. While creating a DX, the data for the DX would already be defined as currency, numeric, text, and so on.

In the Column attribute General Information tab, the **Attribute** field needs to be mapped to the attribute name mentioned in the DX. For example, if an attribute name in the DX is Mytext and the attribute type is selected as Text, the attribute field should be mapped with the attribute name Mytext.

# Adding Charts to Optional Tabs

**To create a chart type table in a newly created optional tab:**

1. In Dashboard Management, double-click the name of the new tab to open it.
2. In the **Tables** tab, click the **New Dashboard Table** icon.

3. Specify table fields and attributes as described in the table below:

| Property | Description | Possible Settings |
|---|---|---|
| Name | Enter a name for the table. | |
| API Name | You can specify an<br>API name, but internal consistency is enhanced by accepting the system-generated<br>name. | |
| Description | Enter a description for the table | (optional) |
| View List Type | Lists the type of table. Select **Chart (when you select Chart, additional options are displayed).** | Chart, Table, Custom, Advanced Search |
| Dashboard Extension | Lists all the dashboard extensions created for chart type. Select the dashboard extension you want. | |
| Visible | Select Yes to enable display in Web Client | Yes/ No |
| Chart Type | Select the type of chart you want displayed. | Area, Bar, Line, Pie, Polar, Scatter, Stacked Area, Stacked Bar, Table |
| X axis | Type the X axis label. | (optional) |
| Y axis | Type the Y axis label. | (optional) |
| Show Legend | Specify whether the chart legend should display on screen. | Yes/ No |
| Legend Position | Specify the position where the Legend should be displayed. | Bottom, default, left, right, top |
| 3D Style | Specify whether the view should be 3-dimensional. | Yes/ No |
| Header | Enter a header note if required. | (optional) |
| Footer | Enter a footer note if required. | (optional) |

**Note** For Stacked type Charts, the Chart DX can be coded with unlimited X- and Y-axis values.

**Important** In the **Dashboard Extension** field, all Chart type DXs will be displayed only if the user has the SDK license and if the DX was created as Chart type. There are two classes for creating Chart type dashboard extensions: 1. ChartDataModel, and 2. ChartDataSet. These classes should be used while creating Chart DXs. The saved Chart DX needs to be saved in the *agile_home/integration/sdk/extensions* folder and linked in the **Dashboard Extension** field. For further information on creating DXs, refer to Agile PLM SDK Guide

## Adding Advanced Search to Optional Tabs

**To create a dashboard table that uses an Advanced Search query as the data source:**

1. In Dashboard Management, double-click the name of the new tab to open it.
2. In the **Tables** tab, click the **New Dashboard Table** icon.
3. Specify table fields and attributes as described in the table below.

| Property | Description | Possible Settings |
|---|---|---|
| Name | Enter a name for the table. | |
| API Name | You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. | |
| Description | Enter a description for the table. | (optional) |
| View List Type | Select Advanced Search. | Chart, Table, Custom, Advanced Search |
| Dashboard Extension | | Disabled (not applicable for Advanced Search |
| Visible | Select Yes to enable display in Web Client. | Yes/ No |

**Note** When you select View List type as Advanced Search, the dialog box automatically displays fields applicable only to Advanced Search type table.

# Configure Table for Advanced Search

Once the table is created, double-click the new created table and click the **Configure Dashboard Table** icon in the General Information tab.

You can create a new or use a saved search. The Configure Table dialog is similar to the Advanced Search function of Agile except for a few differences as follows:

- Output Fields

- The **Output Fields** option allows you to select the available fields as the X and Y axis values. Clicking the **Output Fields** button displays the Customize Output Display dialog. Select the required fields from the Available Fields list and click the left and right arrow buttons to move field names to and from the Available Fields and Selected Fields lists.
- The **Display Chart** button allows a Chart and Table type display option. If Chart option is selected, the Configure Table dialog is displayed. You can do the following:

- Select the Chart type.
- Select the 3D style for Chart.
- Show or Hide the Legend.
- Select the position of the Legend to be displayed in the Chart.
- Select the X axis value from the drop down list (which is displayed from the Customize Output field)
- Label the header, footer, X axis and Y axis values on the chart
- Perform a mathematical function for the Data values such as Average, Count, Max, Min, and Sum.

# Adding Custom (URL) Process Extensions for Optional Tables

URL process extensions that are created with Dashboard as the integration point are listed in the Dashboard Extension field, while a Dashboard table with View Type List is created. The output of the URL process extension is displayed in the Dashboard table.

For more information, see the *Agile SDK Developer Guide*.

# Reordering Table Rows in a Tab

You can reorder the tables in a tab from **Dashboard Management** > **[OptionalTab Name]** > **Tables** > **Config Dashboard** icon . The **Config Tab** dialog is displayed in which you can select all the tables from the Available Content field and place it in the required rows. There are only three rows; by default the new tabs are created in Row 3. You can reorder the tabs in the order of the rows to be displayed.

# Displaying Optional Tabs

To ensure that all optional tabs that are newly created under **Systems Settings > Dashboard Management** display properly, go to **User Settings > Privileges > Dashboard Tab View** and enable each tab for end-users in Web Client.

Select the New icon and type in a new privilege name and description. Set **Enabled** to **Yes**. All created tabs will be displayed in the **AppliedTo > Choices** field. Select the tabs you want displayed in the Dashboard and use the right arrow icon to move these to the **Selected** area. Any tab you choose can be set as the Home page.

# Chapter 27

# Solution-specific Administration

**This chapter includes the following:**


- Documentation ..... 299
- Configurable solutions for Administrator ..... 299


## Documentation

User guides for the PLM solutions contain a chapter for the administrator that deals with specific settings and configurations for each solution, and are mandatory reading for each solution purchased by your company.

For instance, every installation of PLM includes the solution called Product Collaboration. In the *Agile PLM Product Collaboration User Guide*, there is a chapter dedicated to the Agile administrator, not the end-user.

The Agile administrator should also become familiar with the end-user documentation as the manuals for end-users contain valuable information that assist the Agile administrator in grasping what the solution is all about and what the user experience incorporates.

## Configurable solutions for Administrator

The solution-specific nodes and sub-nodes are the following:

**Product Cost Management** node folder

- Ship To Locations
- RFQ Terms and Conditions

**Product Portfolio Management** node folder

- Schedule Status
- Cost Status
- Quality Status
- Resource Status
- Default Role
- UI Configuration Data

**Agile Content Service** node folder

- Subscribers
- Destinations
- Events

Filters

Package Services

**Product Governance & Compliance** node folder

Signoff Message

Compliance Rollup Scheduling

Compliance Rollup Role Setting

Supplier Declaration Process Extension

Specification Mapping

**Note** Product Collaboration is not seen as a solution folder, but is also a solution that requires configuration by the Agile administrator.

# Chapter 28

# Event Management

**This chapter includes the following:**


- Introduction to Event Management ..... 301
- Anatomy of Event Subscriptions ..... 303
- Handler Types ..... 307
- Event Types ..... 308
- Working in Events Node ..... 312
- Working in Event Handlers Node ..... 315
- Working in Event Subscribers Node ..... 318
- Sample Event Subscriptions ..... 323
- Monitoring Events ..... 323
- Control of Event Components ..... 325
- More about Selected Event Types ..... 327


# Introduction to Event Management

Event Management allows Agile PLM customers to automate their business processes across the PLM suite of solutions. The Event Management framework provides the administrator with a flexible alternative to Process Extensions for extending Agile PLM.

This chapter describes the components of Event Subscriptions and how the administrator creates those components in the **Event Management** nodes. Two other chapters in this manual pertain to this robust new feature of Agile PLM:

- Event-based Notifications are created in the **Notifications** node and are documented in Event-based Notifications on page 277.
- Scripting and Sample Events on page 335 gives a brief introduction to the scripting capability and sample configured Event Subscriptions.

Also, *Agile PLM SDK Developer's Guide* provides technical documentation of the Event framework for the Agile administrator or assisting programmer. Event Management leverages the Agile SDK just as Process Extensions do, only with much greater flexibility and specificity to PLM business objects and user actions.

## Preliminary Configuration

### Access to the Event Management nodes

Ensure that your own User Profile has proper access to all six **Event Management** nodes via the AppliedTo property in the Administrator privilege mask.

## Global Event control

Once you can access the **Event Management** node folder, still, the Event functions are initially disabled on all 9.3 PLM installations. When the administrator has configured Event Subscriptions and wants to commence their use by the system, Event triggers are enabled by the **Enable Triggering Events** button on the Admin toolbar. Only users with the Administrator privilege mask can see and use this button.

| | |
|---|---|
| **Caution** | It is highly recommended that Event Subscriptions and their various masks are first created and tested on a PLM Test system before introducing them into the Production system. With Agile Configuration Propagation (ACP, see separate manual), fully tested Event Management configurations can easily be transferred from Test to Production systems. |

There is extensive control of the layers of Event Management; for more information, see Control of Event Components on page 325.

Once initially enabled, Event Management will start automatically when the Agile Application Server is restarted; this step does not need to occur again.

## Systemwide Preferences that pertain to Events

Several systemwide Preferences must be set to govern some aspects of Event Management: *Agile Script Log Level*, *Event Maximum Nested Levels Allowed*, and *Event Monitor Log Level*. These are detailed in Preferences Defined on page 371.

# Event Management Node Folder

These are the nodes in the **Event Management** node folder and a brief description of the purpose of each node.

***Events node –***

Create and manage Event masks.

***Event Handlers node –***

Create and manage Handler masks based on "Java PX" and "Script PX" handler types.

Notifications, the third handler type, are created and managed in **Notifications** node.

***Event Subscribers node –***

Create and manage Subscriber masks.

***Event Types node –***

Library of default Event Types (see Event Types Defined on page 309).

***Event Handler Types node –***

Library of default Handler Types (see Event Handler Types on page 307).

***Event Handler Monitor node –***

Review the status of transactions and results, such as messages or errors, from Handler transactions that have been invoked (see Event Handler Monitor on page 324).

## About Process Extensions and Notifications Nodes

With the incorporation of the Events framework, PLM has two types of Notifications and multiple types of Process Extensions. Please note the distinctions and naming conventions below.

### *Default Notifications and Event-based Notifications*

- *Default Notifications* are the "out-of-the-box" notification templates in the **Notifications** node. These Notifications are still useful, particularly before you have deployed Event-based notifications.
- *Event-based Notifications* are created in the **Notifications** node. A notification created in PLM 9.3.x has much more flexibility than a default notification. Most Event Subscriptions that are dedicated to notifications have a configured Notification mask for the Handler. See Handler Types on page 307.

### *Custom Process Extensions and Event-based Process Extensions*

- *Custom process extensions* (Custom PX) is a general name for PXs created in **Data Settings > Process Extensions** node. These PXs have limited initiation capability, for instance, they can be added to objects' **Actions** menu. It may be preferable to maintain your Custom process extensions until you have developed and tested Java PX Event Subscriptions (outside your Production system).
- *Event-based process extensions* (Event PX) are created in the **Event Management** nodes. "Event PX" is a general term to contrast with "Custom PX", but we will be referring more to the specific kinds of Event-based PXs, namely, *Java PXs* and *Script PXs*.

# Anatomy of Event Subscriptions

An *Event Subscription* is a single instance of an "administrator-configured event".

The Event Subscription provides complete instructions for an automation, including what kind of object is to be acted on, what actions will be taken, and what result or outcome will be accomplished by the system.

**An Event Subscription, configured and ready to be invoked, consists of...**

an **Event mask** – *specifies the conditions under which an automation will be invoked,* includes an Event Type, an Object Type (often but not always), and additional attributes;

and a **Handler mask** or a **Notification mask** – *describes the intended automated result,*
a Handler mask is a configured Java PX or a configured Script PX; a Notification mask is a configured Notification;

and a **Subscriber mask** – *binds a Handler to an Event,*
a Subscriber names Event and Handler/Notification masks to work in tandem with additional attributes that refine the Event Subscription's behavior;

**... and when the Event Subscription is invoked, the configured result is completed by the system.**

As end-users perform normal actions in PLM, the default Event Types may be triggered dozens of times a day. However, nothing happens unless one (or more) Event Subscription gives the information to the system about what to do when an event type is triggered.

| **Note** | Be careful how you think of "Events" and "triggered Events". It is the Event Types that are triggered by user actions; but nothing happens unless a proper combination of the three masks are in place. |
|---|---|

You cannot create "duplicate" Event masks, that is, there can be only one Event mask that names an Event Type–Object Type combination. However, Event and Handler masks are reusable, that is, they can be named by any number of Subscriber masks to create highly specific Event Subscriptions.

For every Event Subscription that you configure, you must be clear about the object type you want to affect – whether it is a subclass, or a class (and all of its subclasses), or a base class (and all of its "descendant" classes and subclasses).

| **Note** | There are several Event Types that are not specific to object types; this is the first category in the table in Event Types Defined on page 309. |
|---|---|

So a "Create Object" Event mask for Items base class is triggered for the Parts and Documents classes and, in turn, their children subclasses. A "Create Object" Event mask for Parts class trigger for all subclasses of the Parts class. A "Create Object" Event mask for Part subclass triggers for that subclass only, not for other subclasses in the Parts class.

When you create the masks, an automatic naming capability provides a name that you can accept or modify; or, you can create your own name. For example, you may prefix mask names with a business process or subclass.

| **Note** | System-provided names follow internal conventions, but the provided name may not be unique and you may need to modify the provided name. |
|---|---|

A naming system will be a benefit when you have stored many masks in your Event-node libraries. In any case, choose names that help you see at a glance what their use is.

Here is an example of names for three masks in a possible Event Subscription.

Let's say the administrator builds on the "Create Object" event type to specify "objects from the Capacitor subclass". The name of the Event mask is:

```
[Create Object  Capacitor]
```

Every Event mask names an Event Type, in this case the Create Object event type. This event type

takes an object type (most of them do), so this Event mask specifies Capacitor, a subclass of the Parts class.

Now a Handler mask is created:

```
[New Part on ECO]
```

This handler's name gives the idea that a change order will be created with the new part associated with it. Also, you can see that this Handler mask will work with any Event mask that names a subclass of the Parts class, or the Parts class itself.

Finally, a Subscriber mask that names the Event mask and Handler mask is:

```
[Create Object  Capacitor  New Part on ECO]
```

When the system recognizes that a Capacitor was just created, and this Event Subscription is invoked, the Subscriber provides the specific instructions and the system runs the Handler that produces the desired outcome:

> "Since a Capacitor was just created, now (automatically) create a new Change Order and add that new object on the ECO."

With this automation, the user does not have to create an ECO and associate it with the part; the user simply creates a part and soon after sees the system-generated ECO containing the part.

The names of the default Event Types (see Event Types Defined on page 309) give an idea of the range of automated Event Subscriptions in PLM that the administrator can create in the Event Management framework.

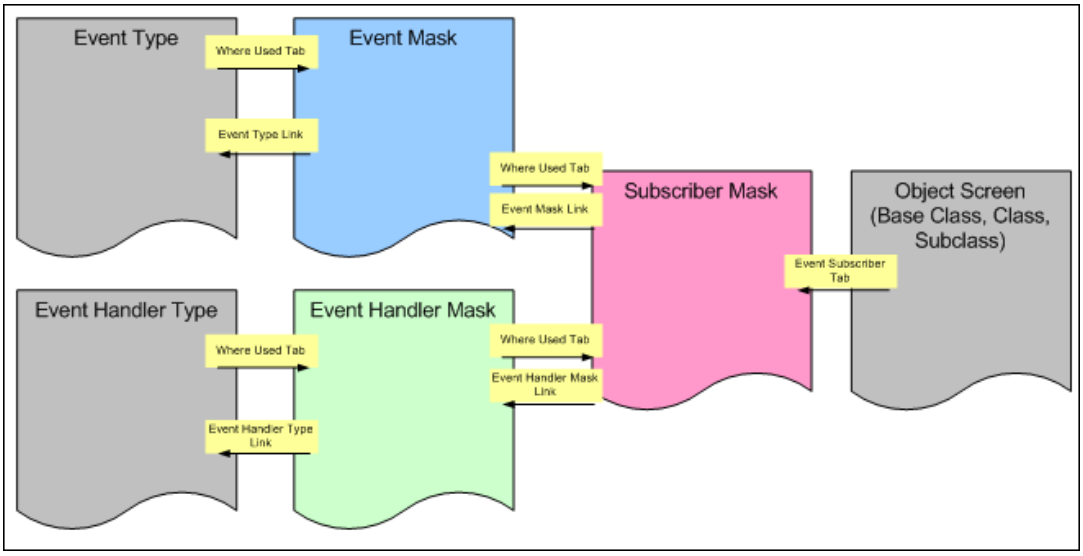
# Event Subscription Components are Interconnected

This schematic depicts the connections between individual objects that are created in **Event Management** nodes.

For example, when you open an event type from the **Event Types** node, it has a **Where Used** tab that lists all event masks that use that event type. In the other direction, when you open an event mask from **Events** node, the field "Event Type" is a live link to the named event type object.

The rightmost element "Object Screen" represents any object type – each row in **Classes** node – and each and every object type has an **Event Subscribers** tab that lists all event subscribers that are associated to this object type (because a given subscriber names an event mask that names the object type).

| **Note** | For Event-based Notifications, a Notification mask is the Handler. The distinction between Handler mask and Notification mask is simply because they are created in different nodes in Administrator, namely, **Events Handlers** node and **Notifications** node. |
|---|---|




# Event Subscribers Tab on All Object Types

The **Event Subscribers** tab has been added to every base class, class, and subclass (default and administrator-created). When you create an Event Subscription that points to an object type, you can open the object type in **Classes** and see all its associated Subscribers under this tab.

This is also the best way to see which Subscribers are active or inactive, by using a dropdown filter with these values:

- **All** – displays both Active and Inactive subscribers, per the other criteria in the filtering process.
- **Active** – a subscriber can be Active if each of the three masks are enabled: the Subscriber mask itself is enabled, the Event mask it names is enabled, and the Handler mask or Notification mask it names is enabled.

  In addition, the entire **Event Management** framework must be enabled (through **Admin** toolbar > **Enable Triggering Events** button), as well as the associated Event Type, Handler Type (all three), or Notifications (via **Server Settings** > **Database** > **Notification Enabled**).
- **Inactive** – a Subscriber mask may be enabled, but if either the Event mask or Handler mask that it names is disabled (or if base Event Type, Handler Type, or Event Management itself are disabled), the subscriber is considered Inactive; any disabled Subscriber is also Inactive.

  The Inactive Reason field is automatically populated in every subscriber displayed by filtering for Inactive. The Inactive Reason field displays these various reasons:

  - [Blank, no value] if the Subscriber is active;
  - "Event triggering disabled" if the global **Event Management** control is disabled;
  - "Event is disabled" if the Event mask is disabled;
  - "Event type disabled" if the Event Type is disabled;
  - "Handler type disabled" if the Handler Type (Java PX, Script PX, or Notification) is disabled;

- "Event handler disabled" if the Notification itself is disabled, or
- "Notifications are disabled" if global notifications are disabled.

Remember for any filtering of subscribers on a given object type, the hierarchy of base class, class, and subclass is in effect, as follows:

- If you open a base class, you see those handlers available to that base class (through subscribers).
- If you open a class, you see those handlers available to that class and its parent base class (through subscribers).
- If you open a subclass, you see those handlers available to that subclass, its parent class, and its parent base class (through subscribers).

# Handler Types

The **Event Handler Types** node contains a library of the available Handler Types. Currently there are three handler types, which are described below; first, this table clarifies some terms used with regard to Event Handler Types.

| Category of Handler Type | Handler Type | Kind of Mask | Where created & managed | Comment |
|---|---|---|---|---|
| PX-based Handler Types | **Java PX**<br>**Script PX** | Handler mask | **Event Handlers** node | When you create a Handler mask, only Java PX and Script PX are in dropdown menu for selection. |
| Notification-based Handler Type | **Notification** | Notification mask | **Notifications** node | "Create" in **Notifications** node creates a Notification mask. These can be associated with Event and Subscriber masks, or they can be triggered by Java PX or Script PX handler. |

The Handler Type dictates the form that the Handler's instructions will take. The Handler mask or Notification mask identify what to invoke when the related Event mask has been invoked.

## Process Extension–based Handler Types

### Java Process Extensions

A Java Process Extension (**Java PX** or **jPX**) is a Java class deployed on the Agile Application Server. The Java PX handler type allows you to trigger compiled Java code. Compared with Custom PXs (created and stored at **Process Extensions** node), the Events framework gives Java PXs much greater potential for your targeted automations.

Java PXs are stored on the Agile Application Server directory structure. Refer to *Agile PLM SDK Developer's Guide* for information about writing Java PX code and deploying PXs.

### Script Process Extensions

The Script PX handler type (**Script PX** or **sPX**) – a new form of process extension in Agile PLM – allows you to store code directly in the system database. The script is always visible and can be edited right in the Handler mask.

**Important** Groovy is the only scripting language that is supported in Agile PLM 9.3.x.

Scripting is introduced to the administrator in Overview of Scripting in Agile PLM on page 335, and there is substantial information for the developer about scripting in *Agile PLM SDK Developer's Guide*.

## Notification-based Handler Type

### Notification Masks

Admin-created Notification masks work in conjunction with the Event framework. Use the **Notifications** node to create Notification masks, as well as manage the Default notifications.

In an Event Subscription whose outcome is a notification, the Subscriber mask names the Notification mask, rather than a Handler mask (example: Notify when Affected Items Table is Modified on page 345). Notification masks also can be triggered by Java PX or Script PX handler (example: Notify Create User when Item is Incorporated on page 340).

Details about Handler masks, their attributes, and related business rules are taken up in Working in Event Handlers Node on page 315. Notification masks are detailed in Notifications on page 275.

## Notes on Event Handler Types Node

When you double-click to open a handler type, the **General Information** tab contains just a few attributes: Name, API Name, and Description are all uneditable.

All handler types are enabled by default, and they can be enabled and disabled (see When a Handler Type is Disabled on page 326).

On the **Where Used** tab of the handler types:

- Java PX – lists all Handler masks that are based on the Java PX handler type.
- Script PX – lists all Handler masks that are based on the Script PX handler type.
- Notifications – lists all Event-based Notification masks.

The rows in these **Where Used** tables are active links so you can "click" directly to a specific Handler mask.

# Event Types

The **Event Types** node lists all of the supported Event Types. The event types are enabled out of the box, and you can disable and re-enable them at any time. See Control of Event Components on page 325.

Use the list in **Event Types** node to see what user actions can be the starting point for automated Event Subscriptions. Because the event types name what the system is "triggering," it may also help to think of the event types as the points in PLM where Handler masks are invoked.

The Name of each event type begins with the action (or verb, for example, Approve) and a target (or noun, for example, Workflow, so, **Approve for Workflow**). If you click the Description column header, the event types sort by the *targets*, alphabetically (so, **Workflow, Approve for**), ascending or descending.

The **Where Used** tab lists all Event masks that are associated with that event type. These rows are active links so you can go directly to a specific Event mask.

# Default Event Types

The following table lists all of the event types that are available to create Event masks. The default event types can be categorized in four groups to help understand their scope:

- **Global event types** – do not apply directly to PLM object types
- **Generic Object-based event types** – apply to all PLM object types
- **Specific Object-based event types** – apply to specific PLM object types
- **Workflow-based event types** – apply to all Workflow objects

The purpose of each event type is described by what user action will trigger the event type (right column) and by the PLM base class it applies to (center column).

Information about selected event types beyond the scope of this table is presented in More about Selected Event Types on page 327.

| Event Type | Applies to these PLM base classes | Event type is triggered by this end-user Action (or system action or through SDK) |
|---|---|---|
| ***Global Event Types*** | | |
| **Scheduled Event** | (Global event types do not apply directly to PLM base classes.) | There is a Schedule Event Task that controls the execution of this event type. See Task Configuration on page 384 > **Schedule Event Task**. |
| **Export Object** | | **Action** > **Export**, or whenever the Export utility is opened. |
| **Extend Tools Menu** | | Event triggered from the Tools Menu. |
| **Transfer Authority** | | Transfer Authority is added, removed, or changed. |
| ***Generic Object-based Event Types*** | | |
| **Extend Actions Menu** | All base classes | Creates link or action, or something that can be activated from a business object's **Actions** menu. |

| Event Type | Applies to these PLM base classes | Event type is triggered by this end-user Action (or system action or through SDK) |
|---|---|---|
| **Create Object** | All base classes | An object is created; this does not include objects that are created by the "Save As" method of object creation (see **Save As Object**). |
| **Delete Object** | All base classes | An object is Deleted. |
| **Save As Object** | All base classes | An object is created by **Actions** > **Save As**, which creates a copy of the current object. |
| **Check In Files** | All base classes except Reports; also excludes Automated Transfer Orders class | A File Folder is checked in, either from direct action on the file folder or through an object's **Attachments** tab. The Replace File action is part of checkin and will not generate an **Update Table** event on the file folder. |
| **Check Out Files** | All base classes except Reports; also excludes Automated Transfer Orders class | A File Folder is checked out, either from direct action on the file folder or through an object's **Attachments** tab. From Web Client, the Checkout action creates a separate **Get File** action and event. |
| **Cancel Check Out Files** | All base classes except Reports; also excludes Automated Transfer Orders class | A File Folder checkout is canceled, either from direct action on the file folder or through an object's **Attachments** tab. |
| **Get File** | All base classes except Reports; also excludes Automated Transfer Orders class | A user requests to Get a file; this can occur through use of the **Get** button or if selecting the filename causes a Get action. |
| **Update Relationship** | All base classes | A user adds, removes, or directly edits a Relationship on an object's **Relationships** tab; editing includes either updating the Relationship Rule or editing a field on the row. |
| **Update Table** | All base classes; however, certain tables are not supported. | A user modifies a table on a business object, for example, on Items' **BOM** tab, or Changes' **Affected Items** tab. See details and exceptions in Update Table on page 329. |
| **Update Title Block** | All base classes | A user directly edits any fields on an object's **Page1**, **Page2**, or **Page3**. |
| ***Specific Object-based Event Types*** | | |
| **Change Status for Sourcing Object** | Requests For Quote, Sourcing Projects | A user performs **Action > Change Status** on a PCM-solution object, also from the **RFQ** tab of Sourcing Projects. |
| **Compliance Rollup on Object** | Items, Mfr.Parts, Part Groups, Declarations | A manual or scheduled rollup commences, e.g., when user performs **Action > Calculate Compliance** on the object, or when the declaration's workflow |

| Event Type | Applies to these PLM base classes | Event type is triggered by this end-user Action (or system action or through SDK) |
|---|---|---|
| | | changes status (manually or automatically). |
| **Incorporate Item** | Items: Parts and Documents | A user selects **Action > Item Incorporate.** |
| **Unincorporate Item** | Items: Parts and Documents | A user selects **Action > Item Unincorporate.** |
| **Purge Version Files** | File Folders | A user selects **Action > File Folder Purge.** |
| ***Workflow Object-based Event Types*** | | |
| **Approve for Workflow** | Changes, Declarations, Programs, PSRs, QCRs, Transfer Orders, Packages, File Folders | A user approves a workflow |
| **Audit for Workflow** | | A user initiates a Status Audit or a Release Audit on a workflow using **Actions > Audit Status** or **Actions > Audit Release**. (System audit does not trigger it.) |
| **Change Approvers or Observers for Workflow** | | A user adds an Approver/Observer to, or removes Approver/Observer from, the **Workflow** tab. This event type is not triggered when Approver/Observer is added or removed via Routing Slip. |
| **Change Status for Workflow** | | A workflow moves from one status to another, either by user action, or by autopromotion, or by forced change due to Relationship rule. |
| **Comment for Workflow** | | A user comments on a workflow. |
| **Escalation for Workflow** | | A workflow has been in a status awaiting signoff longer than the period defined by Review Escalation Period on the status. |
| **Promotion Failure for Workflow** | | A system-initiated workflow status promotion fails; promotion failures can occur due to autopromotions or Relationship triggers. |
| **Reject for Workflow** | | A user rejects a workflow. |
| **Reminder for Workflow** | | A workflow has been in a status awaiting signoff longer than the period defined by Reminder Period on the status. |

Additional information about selected event types is presented in More about Selected Event Types on page 327.

## Multiple Triggers from Single Action

Generally one user action in a PLM client will trigger one event type; however, there are many

actions in PLM that trigger secondary actions, and this fact accounts for how a single user action may trigger more than one event type.

For example, on an item's Actions menu is **Create Change**. When a user selects this action, the Create Object event type is triggered. Meanwhile, when PLM creates the change order associated with the original item, it automatically adds the item to the change's **Affected Items** table, and so the Update Table event type is also triggered.

Here are some additional examples of this:

- Item > **Actions** > **Create Declaration**: triggers Create Object and Update Table (the **Parts** table on the new Declaration) event types;
- QCR > **Actions** > **Create Change**: triggers Create Object and Update Relationship (the **Relationships** tab on the new change order) event types;
- Change object > **Actions** > **Bulk Change**: triggers Update Table (the **Affected Items** table on each new Change, one for each item that is "affected") and Update Table (for each **BOM** or **Manufacturers** table that is redlined on each affected item;
- Checking out a file triggers Checkout File and Get File event types;
- Checking in a file on Agile object (except File Folder) triggers Checkin File and Update Table (Attachments) event types; Modify action if the folder version of the attachment is not the latest; Update Table event type is not triggered if the folder version is set to the latest;
- Action > Create File Folder triggers Create Object and Checkout File event types;
- In Java Client, adding two URLs to a File Folder triggers two events (adding two URLs to any other business object, only one event is triggered; note also that the user can add two URLs to a File Folder in Web Client, only one event is triggered).

### User $AGILEUSER

The following three event types, when used in an Event mask, work in the background:

- Scheduled Event
- Escalation for Workflow
- Reminder for Workflow

For them, the Handler will be invoked as $AGILEUSER. By default, this user is a PLM-system user with many roles assigned, *even though it is not available in the Administrator user interface* (for example, ACS uses it). As a result, if the Handler does not have a role assigned, these "invisible" roles that are assigned to $AGILEUSER are used when running the Handler.

# Working in Events Node

The **Events** node is where you create and manage Event masks.

## Attributes of Event Masks

The table below lists attributes found on the **General Information** tab in Event masks.

All Event masks must name an Event Type. Most Event masks must name an Object Type (excepting those that name one of the "Global" event types).

**Note** System-generated names are a helpful feature when creating Event masks and Subscriber masks. The **Name Helper** button can be used: it concatenates Event Type and Object Type (as long as you select them before you manually enter a Name) to generate a Name and API Name.

Additional attributes may appear in the Create dialog as certain event types and object types are selected; these are given general mention in this table. "Additional attributes" are also noted by event type in More about Selected Event Types on page 327.

The **Where Used** tab lists all Subscriber masks that use the Event mask.

| Attribute | Description |
|---|---|
| **Event Type** | Event type categorizes what user action will trigger the event.<br>In a configured Event mask, this field becomes uneditable. "**Event Type**" itself is a live link: click it to go to the event type. |
| **Name** | Name of the Event mask. You can specify the name, but it is best if the name includes the event type and object type. If you populate Event Type and Object Type fields first, the **Name Helper** button produces a system-generated name (see Note above this table). |
| **API Name** | Name of the object that is used by external processes to identify the object. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name once you have specified the Name of the Event mask. See API Name on page 19. |
| **Description** | Enter a useful description of the Event mask. Descriptions are searchable. |
| **Enabled** | Value is Yes or No; see Control of Event Components on page 325. |
| **Object Type** | Object Type specifies what kind of business object is being acted on (by the user's task) for the event to be triggered. (Exception: when a workflow is selected as the object type of an Event mask, the system uses the object type of the first Matching Criteria property of the workflow. See More about Workflow-based Event Types on page 331.)<br>Only object types that support the Event Type may be selected.<br>Note: The Global event types take no object type: Scheduled Event, Export Object, Extend Tools Menu, Transfer Authority. |
| **Workflow** | Appears only if a workflow-specific event type is selected. See More about Workflow-based Event Types on page 331.<br>Specify a workflow in the system, or **<All>** workflows. (**<All>** is not available for the Change Status for Workflow event type.)<br>If **<All>** is chosen, Object Type attribute becomes uneditable. |
| **Status – from** | Appears if Change Status for Workflow event type is selected; choose one or multiple statuses, or **<Any>** status. |
| **Status – to** | Appears if Change Status for Workflow event type is selected; choose one or |

| Attribute | Description |
|---|---|
| | multiple statuses, or **<Any>** status. |
| **Action** | Appears and lists appropriate actions for some event types; the list of actions may also be modified by selected object type. |
| **Source Object** | Appears only for the Update Relationship event type; names one or multiple object types as source of the relationship ("relating to"). See Update Relationship on page 329. |
| **Target Object** | Appears only for the Update Relationship event type; names one or multiple object types as target of the relationship ("related from"). See Update Relationship on page 329. |
| **Frequency** | Appears only for the event type Scheduled Event; the dropdown is a scheduling utility (similar to Frequency property for Scheduled Event under **Events** of ACS). See Scheduled Event on page 327. |
| **Table Name** | Appears only for Update Table event type; list of tabs/tables pertains to selected object type. See Update Table on page 329. |

# Checklist for Creating an Event Mask

This is a checklist of questions to ask (of, perhaps, a company Change Analyst or Product Analyst) and have answered when you begin to create Event Subscriptions and, more specifically, when you create any Event mask.

- For my company's installed PLM solutions – and routine tasks performed by Agile users – what are all the PLM Actions that I want to automate the system? Are there any automations needed to supplement actions such as Create, Save As, Update Title Block, Update Table, or by other object behaviors (that are represented by the event types)?
- From a given event type being triggered, what is the result that we want to automatically happen?
- What is the proper object type that this Event mask will apply to? You may need to ask: will the Handler or Notification (the outcome of this automation) apply to a specific subclass or subclasses? Can the handler apply to all the subclasses for a class and/or base class?
- If the event involves a Change, which Workflow(s) does it apply to? Does it apply to all workflows or a specific workflow?
- Are there any Event Type–specific parameters to consider? For example, for the event type Delete Object Action, is the Handler mask going to handle soft deletes or hard deletes?

# Creating an Event Mask

**To create an Event mask:**

1. Open the **Events** node and click the **New** button. The new Event mask object appears.

2. In the Event Type field, use the dropdown arrow to select an event type. Use the information in Event Types Defined on page 309 and More about Selected Event Types on page 327 to base your choice.
3. In the Object Type field, use the dropdown arrow to select an object type.

   It is possible to select a disabled object type, so you must be clear about the status of all object types in your PLM system (the **Classes** node).

   If the event type you selected applies to workflows, the Workflow field appears: select the appropriate workflow and then, if Object Type is enabled, an appropriate object type.

   Selection of object type may have also caused the Action field to appear, in which case it is required.

**Important** If you intend to create an Event mask whose Handler is a notification (sent by email to selected users), the object type of the Event mask must be appropriate for the object type of the Notification mask. For example, if the Notification mask specifies Change Orders (a class), the Event mask must specify the class or ECO (subclass) or any other subclass derived from the Change Order class. For Update Relationship event type, the Notification object type must match the Source object type (or the common object type if there is more than one Source object type) of the Event mask.

4. At the Name field, click the **Name Helper** button to the right; the system uses the event type and object type names to generate a standard name for the Event mask.
5. Accept the auto-generated API Name. You can enter your own API name, but it is recommended to let the system's name stand for naming consistency.
6. Fill in a description.
7. In the Enabled field, accept Yes or select No.

   **Note** A disabled Event mask cannot be invoked; any Subscriber mask that names an Event mask that is disabled will itself be Inactive.

8. Click **OK** to save the configured Event mask.

# Working in Event Handlers Node

The **Event Handlers** node is where you create and manage Handler masks that are based on the Java PX or Script PX handler types.

**Note** Here are a couple of tips for quick sorting in the Filter (which becomes more useful as you accumulate Handler and Notification masks).
In Event Handlers node: if you set **Filter By** to Handler Type and **Match If** to Starts With, in the field you can simply type "j" or "s" for all Handler masks based on Java PX or Script PX handler types, respectively.
However, in **Event Handlers** node, typing "n" for Notifications does not return the Notification masks; instead, open **Event Handler Types** node, open Notifications handler type, and look under the **Where Used** tab for that list; or, open **Notifications** node itself.

# Attributes of Handler Masks

The table below lists attributes found on the **General Information** tab in Handler masks.

The **Where Used** tab on a Handler mask lists all Subscriber masks that point to the Handler mask.

| Attribute | Description |
|---|---|
| **Handler Type** | Java PX or Script PX.<br>In a configured Handler mask, this field becomes uneditable; however, "Handler Type" has become a live link: click it to go to the handler type. |
| **Name** | Name of the Handler mask: you specify the name (there is no Name Helper as in other parts of these masks). It is recommended that the name describes what the purpose or result of the Handler is. |
| **API Name** | Name of the object that is used by external processes to identify the object. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name once you have specified the Name of the Handler mask. See API Name on page 19. |
| **Description** | Enter a description that is useful; descriptions are searchable. |
| **Enabled** | Value is Yes or No; see Control of Event Components on page 325. |
| **Role** | Select one or more roles to override the user's default roles when the system executes the Handler. See Roles in Handler Masks on page 316. |
| **Event Action** | Appears if handler type is Java PX. The dropdown list allows the user to pick from a list of Java PXs that have been deployed and are available in the system.<br>Note that the completed Events-based process extension itself is deployed in the same manner that you deploy Custom PXs. |
| **Script** | Appears if handler type is Script PX; the script code is pasted into this field; for more information, see Editing Scripts in the Handler Mask on page 337 and Introduction to Scripting in Events on page 335. |

## Roles in Handler Masks

The PLM Roles that are specified in Handler masks determine what roles are taken into account by the system when running the Handler.

If no role is specified in the Roles attribute, the Handler is run with the system observing the assigned role(s) of the user who performed the original action (that triggered the Event Type that is named by the Event mask that has been invoked in the current Subscription).

Whereas, If the Roles attribute is populated, the Handler is run with the system observing only the roles specified in that field, which essentially overrides the user's roles for the actions dictated by the Handler.

With no overriding role specified in the Handler mask, an error condition will occur if that user does not have sufficient privileges to perform the action (or some partial action) dictated by the Handler.

In this way, the administrator uses the Roles attribute to guarantee that the Handler can complete all of its specified actions.

# Checklist for Creating a Handler Mask

This is a checklist of questions to ask and have answered when you create a Handler mask. These items can also serve as a communication vehicle between yourself and a programmer who is tasked to create Java-based PXs and Groovy script-based process extensions.

- Do I already have configured Event masks? What Event mask am I trying to handle?
- Is this a notification or a process extension? That is, what broad result does the Handler need to accomplish? What kind of instructions must the system have to complete a successful automation?
- If the Handler is to be a notification, has the Notification mask already been created (in **System Settings > Notifications**)?
- If the Handler is to be a process extension, what is the proper handler type for this Handler mask?
- Who is providing the Java code, SDK or Groovy script that will drive this Handler mask?
- Has the Java process extension been developed and tested? Is it properly deployed so the Java PX Handler mask will be able to find it?
- What Roles are needed to run the handler mask? Will the user performing the PLM action always have the right Roles?

# Creating a Handler Mask

**To create a Handler mask:**

1. Open the **Event Handlers** node and click the **New** button. The new Handler mask object appears.
2. In the Event Handler Type field, choose Java PX or Script PX.

   Remember that while Notifications are categorically a handler type, Notification masks are created in **Notifications** node and thus are not offered as a choice when creating a Handler mask.

   If you select Script PX, the Script field appears with this default Groovy script:

   ```
   import com.agile.agileDSL.ScriptObj.IBaseScriptObj
   // add other import statements here
   void invokeScript(IBaseScriptObj obj) {
   //script body starts here.
   }
   ```
3. In the Name field, enter a name.
4. Accept the auto-generated API Name.
5. In the Role field, enter the roles that will permit the Handler to be run as. If a role is specified here, it overrides the original user's roles when the Handler is executed. See Roles in Handler Masks on page 316.

6. Enabled should be set to Yes if you want this Handler mask to be available for Event Subscriptions.
7. Script field: paste in the script for this handler. (You can create the Handler mask without providing a script yet; in this case it is better to set Enabled to No.)
8. Click **Validate Script** to make sure the script will run properly.
9. Click **OK** to save the configured Handler mask.

# Working in Event Subscribers Node

The **Event Subscribers** node is where you create and manage Subscriber masks.

The Subscriber binds (by naming) an Event mask and a Handler mask, and provides concrete "How" and "When" information about the automated outcome with such fields as Trigger Type (Pre- or Post), Execution Mode (Synchronous or Asynchronous), Order and Error Handling Rule; all of these are defined later in this section.

You can create more than one Subscriber mask that names a given Event mask. A second Subscriber mask naming the same Event mask would name a different Handler mask. In this case, when this Event fires, more than one Handler will be executed. The Subscriber can specify the order for the system to run the Handlers (Order), as well as how to handle error cases (Error Handler Rule).

**Note** Here are a couple of tips for quick sorting in the Filter (which becomes more useful as you accumulate Handler and Notification masks).
In **Event Subscribers** node, if you set **Filter By** to Handler Type and **Match If** to Starts With, in the **Value** field you can simply type "j" or "s" to return all Subscribers with a Handler mask based on, respectively, Java PX or Script PX handler types. In this node, you also can enter "n" and Subscribers with a Notification mask are returned (unlike in the **Event Handlers** node).

## Filtering in Dialogs during Create Subscriber

When you create a subscriber mask, two dialogs offer similar filtering attributes, which are mentioned here for comparison.

- **Event** [Mask] field dropdown arrow brings up the **Select Event** [Mask] dialog, in which you can **Filter By**:
  - Name [of Event masks]
  - Event Type
  - Object Type
- **Event Handler** [Mask] field dropdown arrow brings up the **Select Event Handler** [Mask] dialog, in which you can **Filter By**:
  - Name [of Handler masks or Notification masks]
  - Handler Type
  - Object Type

In this second dialog, if **Filter By** is Handler Type, and **Match If** is Starts With, in the **Value** field this dialog returns the appropriate Handler or Notification masks when you type "j" (Java PX Handler masks), "s" (Script PX Handler masks), or "n" (Notification masks).

It may be said explicitly (as these filters imply) that the selectable Event mask and selectable Handler mask must be compatible. This point may help explain why some Event masks or Handler masks do not show up as you construct the Subscriber mask. Other compatibility concerns come up with Subscriber attributes such as Trigger Type and Execution Mode; they are suggested in the definitions of Subscriber attributes below and in sections after the table.

# Attributes of Subscriber Masks

The table below lists attributes found on the **General Information** tab in Subscriber masks.

Subscriber masks also have a **Monitor** tab, which provides a view of the row for the current Subscriber mask from the **Event Handler Monitor** (described in Event Handler Monitor on page 324).

| Attribute | Description |
|---|---|
| **Name** | Name of the Subscriber mask: it can be system-generated the Event (mask) and Event Handler (mask) fields are given values |
| **API Name** | Name of the object that is used by external processes to identify the object. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name once you have specified the Name of the Subscriber mask. See API Name on page 19. |
| **Description** | Enter a description that is useful; descriptions are searchable |
| **Enabled** | Value is Yes or No; see Control of Event Components on page 325 |
| **Event** | Name of Event mask bound by this Subscriber mask: use dropdown arrow to bring up **Select Event** dialog to filter and choose Event mask.<br>In a configured Subscriber mask, although this field remains editable (because you can modify the Subscriber to point to another Event mask), "**Event**" has become a live link: click it to go to the designated Event mask itself. |
| **Event Type** | Appears (uneditable) in the saved Subscriber mask from the selected Event mask |
| **Object Type** | Appears (uneditable) in the saved Subscriber mask from the selected Event mask |
| **Event Handler** | Name of Handler mask bound by this Subscriber mask: use dropdown arrow to bring up **Select Event Handler** dialog to filter and choose Handler mask.<br>In a configured Subscriber mask, although this field remains editable (because you can modify the Subscriber to point to another Handler mask), "**Event Handler**" has become a live link: click it to go to the designated Handler mask itself. |
| **Handler Type** | Appears (uneditable) in the saved Subscriber mask from the selected Handler mask |
| **Trigger Type** | **Pre** – invokes the Handler before the action (described by the event type named in the Event mask) occurs.<br>**Post** – invokes the Handler after the action (described by the event type named in the Event mask) occurs. "Post" is generally the more applicable setting.<br>See Trigger Type: Pre and Post on page 320 and Trigger Types and |

| Attribute | Description |
|---|---|
| | Execution Mode on page 321. |
| **Execution Mode** | **Asynchronous** – the transaction completes and the interface control is returned to the user while the Handler is executed in a different thread in the background.<br>**Synchronous** – the user waits for interface control to return until after the handler completes and then handler is executed in the same thread.<br>If Trigger Type is Pre, Execution Mode can only be Synchronous.<br>See Execution Mode on page 321. |
| **Order** | A number (integer) that sets the sequence of synchronous events.<br>Applies to synchronous Subscribers only.<br>See Order and Execution Mode on page 321. |
| **Error Handling Rule** | *The basic definitions of* **Continue** *and* **Stop** *are elaborated below.*<br>**Continue** – If there is an error during the execution of a Handler with synchronous execution mode, the error is ignored.<br>**Stop** – If there is an error during the execution of a Handler with synchronous execution mode, the original action and the Event Subscription is ceased.<br>*When Execution Mode is Synchronous:*<br>▫ When the Handler is Pre and Synchronous – **Continue** allows remaining Handlers and the transaction to continue; **Stop** ceases remaining handlers and the transaction.<br>▫ When the Handler is Post and Synchronous – **Continue** allows remaining Handlers to continue; **Stop** ceases remaining synchronous Handlers.<br>*When Execution Mode is Asynchronous:*<br>▫ When Execution Mode is Asynchronous, the **Error Handling Rule** is disabled; the Event Subscription runs, the PLM transaction has already occurred, so the user sees a successful result. |

## Trigger Type: Pre and Post

Pre and Post are settings for the Subscriber mask attribute Trigger Type. The familiar prefixes "pre" and "post" are used to mean, respectively, before and after. In configuring Event Subscriptions, you must choose whether you want the Event Subscription to be run before or after the transaction.

- **Pre**(-event) trigger types signal a point just **before** the system starts the transaction for the PLM action that the user requested. A Subscriber with a Pre trigger type can be used to prepare objects or data for a subsequent, upcoming action; or to validate objects or data.
- **Post**(-event) trigger types signal a point just **after** the system completes transaction for the PLM action that the user requested. A Subscriber with a Post trigger type can be used for notifications, reminders, escalations, or to perform some auditing tasks based on an earlier action (such as tasks accomplished by Pre-event Subscribers); or external-system integration tasks.

Notification masks are handled only by Post-type Event Subscriptions.

This discussion only defines these terms; you may consult the schematic called "Triggering and Processing an Agile PLM Event" in Chapter 24 of *Agile PLM SDK Developer's Guide*.

## Execution Mode: Synchronous or Asynchronous

Event handlers can be run either asynchronously or synchronously.

Generally, *asynchronous* means “not occurring at the same time” and *synchronous* means “occurring at the same time”.

In Event Management, an asynchronous Handler means the Handler is executed in a separate thread, and the original thread does not wait for the Handler to finish (that is, it is not blocked). A synchronous Handler is executed in the same thread as the initiator, and it waits for the Handler to finish.

### *Trigger Types and Execution Mode*

The choice of the handler's Execution Mode varies based on the choice of Trigger Type:

- Pre-event triggers support only those handlers that can be run synchronously.
- Post-event triggers support handlers that can be run both asynchronously and synchronously.

  Notifications are always Post trigger type and Asynchronous execution mode.

**Note** Be sure to read the nuances about Pre, Post, and Synchronous in "Error Handling Rule" in the above table.

### *Order and Execution Mode*

For synchronous Subscribers, either pre or post, the Order value is used to determine the sequence of calling handlers.

For example, if there are three Subscriber masks that trigger based on a user creating a part. These could have been triggered because of Event masks that name the object types Items (base class), Parts (class), or Part (subclass).

The system evaluates all three Subscribers together: it sorts first by values for Order, so if you entered 1, 2, and 3, respectively, in these three Subscribers, the system knows which one to act on first, and so on.

If Order has no values or the same value, the system resorts to selecting Subscriber names in alphabetical order to determine the sequence of calling Handlers.

Again, you may consult the schematic called "Triggering and Processing an Agile PLM Event" in Chapter 24 of *Agile PLM SDK Developer's Guide*.

# Checklist for Creating a Subscriber Mask

Here are some questions to be answered when you set up a Subscriber mask. The attributes in the Subscriber mask may have been already considered when you set up the Event and Handler masks.

- Do I have all the relevant Subscribers to support my Event masks and Handler masks?
- What Event mask is being named in this Subscriber mask? What is the event type that you are interested in?
- What Handler mask or Notification mask is being named in this Subscriber mask? What Handler do you want invoked when the event type of interest is triggered?
- Which Trigger Type should be used, Pre or Post? Why do you want the Handler to invoke Pre or Post?
- Which Execution Mode should be used, Asynchronous or Synchronous?
- What is the Order in which the Handler mask should be invoked?
- What Error Handling Rule should be used?
- Am I certain that all elements of this Event Subscription are enabled? Can I verify that this Subscriber mask is Active? (See Event Subscribers Tab on page 306.)

# Creating a Subscriber Mask

**Note** System-generated names are a helpful feature when creating Event masks and Subscriber masks. The **Name Helper** button can be used: when you create a Subscriber mask, the system concatenates selected Event mask name and Handler mask name (as long as you select them before you manually enter a Description, in which case the system uses the Description to generate the name, with a limit of 150 characters).

**To create a Subscriber mask:**

1. Open the **Event Subscribers** node and click the **New** button. The new Subscriber mask object appears.
2. At the Event field, use the dropdown arrow to bring up Select Event dialog.

   Use **Filter By** and **Match If** criteria (see Filtering in Dialogs during Create Subscriber on page 318) to return Event masks, and select one.
3. At the Event Handler field, use the dropdown arrow to bring up Select Event Handler dialog.

   Use **Filter By** and **Match If** criteria (see Filtering in Dialogs during Create Subscriber on page 318) to return Handler masks or Notification masks, and select one.

**Important** If you are creating an Event Subscription whose outcome is a notification (sent by email to selected users), the object type of the Event mask must be the same as the object type of the Notification mask. This is a good time to verify that this condition is met by the Event mask (designated in step 2) and the Notification mask (designated in this step).

4. At Name field, click the **Name Helper** button to the right; the system uses the Event mask and Handler mask names to generate a standard name, as well as the API Name. If the system-generated name is not sufficient, you may change it.
5. A new Subscriber mask is enabled by default; you can set Enabled field to No, but of course it

will not run the Subscription until it is enabled.

6. At Trigger Type field, select Pre or Post.
7. At Execution Mode field, select Asynchronous or Synchronous.

   If Trigger Type is Post, Execution Mode field is automatically set to Asynchronous.

   Post and Asynchronous are the correct settings for notification-specific Event Subscriptions.
8. If appropriate, at Order field, give an order number.
9. Similarly, if an Error Handling Rule is required, select one.
10. Click **OK** to save the configured Subscriber mask.

# Sample Event Subscriptions

The next chapter introduces scripting for the administrator and presents five sample Event Subscriptions, including the script, as appropriate.

- Update Item Description on page 339

  This Event Subscription updates an item's Description when a user updates the **Title Block** of any business object from one of all the Parts-class subclasses.
- Notify Create User when Item is Incorporated on page 340

  This Event Subscription creates a notification and sends it to the user who created that part or document when an item – an instance of any subclass in the Parts or Documents subclass, and therefore of the Items base class – is incorporated. Note that this notification is generated by instructions from a script (a Script PX handler), rather than a Notification mask created by the administrator (as in last example below).
- Set Change Analyst on page 341 when ECO changes status

  This Event Subscription automatically sets the Change Analyst when an ECO changes status to Submitted.
- Add Document to New Part on page 343

  This Event Subscription automatically creates a Document and adds it to the Assembly BOM any time a part is created.
- Notify when Affected Items Table is Updated on page 345

  This Event Subscription automatically sends a notification (created in **Notifications** node) any time the **Affected Items** table of a Change is updated.

# Monitoring Events

The **Event Management** framework provides several means to monitoring handlers and subscribers, or troubleshooting problems with triggered events:

- **Event Handler Monitor** node is a monitor of event activities; it is detailed below;
- An Event-specific log file; see Event Log File on page 325;

- Runtime error messages are presented in an error dialog; and,
- Execution of Java PX, Script, or Notification is logged on the object's **History** tab.

# Event Handler Monitor

The **Event Handler Monitor** keeps track of which Handlers have been run as a result of events being triggered. The administrator can see how many handlers have been invoked and how many are still running. For each Handler, the administrator can see when it was invoked, what Event mask was involved, the Handler's current status, when it finished, explicit return messages from the Handler, or exceptions if something is wrong.

The systemwide Preference **Event Handler Logging** can be set to log all invoked Handlers (*All*), to log only errors (*Error*), or turned off (*Off*).

When you open **Event Handler Monitor** node, use the filter bar (table search) to return existing event masks. The **Filter By** attributes in the **Event Handler Monitor** node are:

- Event [that is, name of Event mask]
- Event Handler [that is, name of Handler mask or Notification mask]
- Event Subscriber [that is, name of Subscriber mask]
- Status
- Object

## Attributes of Event Handler Monitor

This table defines the Event Handler Monitor attributes. Note that each Subscriber mask has a monitor view that shows essentially the same information, only for Handlers named by that Subscriber.

| Attribute | Definition |
|---|---|
| **Event Type** | The event type of the triggered Event mask |
| **Event** | Name of the Event mask |
| **Handler Type** | The handler type of the designated Handler mask |
| **Event Handler** | Name of the Handler mask |
| **Event Subscriber** | Name of the Subscriber mask |
| **Trigger Type** | Whether the Event Subscription is to be run before or after the transaction |
| **Execution Mode** | Whether the Handler is to be run synchronously or asynchronously |
| **Object** | The specific business object on which the Event was triggered |
| **Start Time** | When the Handler begins to run, displayed in the user's Time Zone |
| **Stop Time** | When the Handler has finished running, displayed in the user's Time Zone |
| **Owner** | The end-user whose action triggered the Event |
| **Status** | These are the values for Status and what they mean:<br>**Running** – handler is in progress |

| Attribute | Definition |
|---|---|
| | **Completed** – handler has finished<br>**Failed** – handler did not finish<br>**Queued** – waiting to be run, only applies to asynchronous handlers<br>**Unknown** – handler failed but the reason is unknown<br>**Submit for Notification** – status for Notification handler |
| **Message** | The result message that is returned by the handler mask |
| **Thread ID** | ID of thread for which the Handler runs |
| **Event Count** | Tracks the level of "nested" events, that is, events triggered by an event (not a user's action). The allowed maximum of nested events is defined in the systemwide preference "Event Maximum Nested Levels Allowed". |

## Event Log File

In addition to the Event Handler Monitor, Agile PLM also provides an Event Log file, **SOAEvent.log**. It records all event-related activities occurring in the application.

This log file is configured in **Log.xml**, which is established at Install time. The "log level" can be configured from log.xml. Like other Agile log files, SOAEvent.log resides on the File Server.

Log.xml is found in **<AgileInstallFolder> \ AgileDomain \ config**

## Object History Tab

A business object's **History** tab will indicate such information as: in the Action column, what user-defined action called the process extension; in the Details column, what the Handler name is and any returned message (if an event was triggered on the object). Scheduled Notification is also tracked in an object's **History** tab.

# Control of Event Components

The following subsections specify the "stop controls" in **Event Management** nodes.

In any given Event Management node, multiple rows can be selected and disabled or enabled with one click. The details that follow all come with an implicit warning to be careful about what you are disabling or, for that matter, enabling.

**Important** It is highly recommended that Event Subscriptions and their various masks are first created and tested on a PLM Test system before introducing them into the Production system. With Agile Configuration Propagation (ACP, see separate manual), fully tested Event Management configurations can easily be transferred from Test to Production systems.

# Global Event Control

The Admin toolbar has a button called **Disable Triggering Events**. When it is clicked (and you respond to the prompt), no Events are triggered from that point on, until the same button – which changed to **Enable Triggering Events** – is clicked. At that point, event triggering resumes.

The **Disable Triggering Events** button is, like the other buttons on the Admin toolbar, enabled only for those users who have been assigned the Administrator privilege mask.

When Triggering is disabled, all currently running transactions will be completed, but no new events are triggered. The button only disables triggering, that is, it does not impact administrator configuration of events.

**Note** The **Disable/Enable Triggering Events** button creates a persistent setting: on startup, the system begins with this setting in the mode used when the server was stopped.

Any indirect changes in the setting of this button – which includes changes caused by switching database, or being enabled/disabled by another administrator – does not take effect until logout and log back in to Java Client.

# When an Event Type is Disabled

If you disable an Event Type, the system does not change the Enabled property of the Event masks that use that event type; however, as a result, each and every Event mask of the disabled event type is effectively inactive – which renders inactive any Subscriber that names it.

So with one click you could effectively "shut down" all *Get File*-based events, for example.

You can still create Event masks of that event type while it is disabled.

If you re-enable that event type (by selecting its row and clicking the **Enable** button), the system resumes triggering for that event type. Subscribers that name such Event masks may be active again.

# When a Handler Type is Disabled

If you disable a handler type, the system does not change the Enabled property of any Handler masks that use that handler type; however, each and every Handler mask of the disabled handler type is effectively inactive – which will render inactive any Subscriber that names that Handler.

By disabling any handler type, you are therefore effectively preventing the execution of the Handler with that handler type. So you could effectively "shut down" all *Java PX*-based events, or all *Script PX*-based events, or all *Notification*-based events with one click.

You can still create Handler masks of that handler type while it is disabled; they simply will not be executed.

If you re-enable that handler type (by selecting the row and clicking the **Enable** button), the system resumes responding to all Handler masks based on the handler type. Subscribers that name such Handlers may be active again.

## When an Event Mask is Disabled

If you disable an Event mask, Subscribers that name such Event masks are then inactive.

You can still modify a disabled Event mask; it simply will not respond when invoked.

If you re-enable that Event mask, the system resumes responding to it. Subscribers that name such Event masks may be active again.

## When a Handler Mask is Disabled

If you disable a Handler mask, all Subscribers that name that Handler are then inactive.

You can still modify a disabled Handler mask.

If you re-enable that Handler mask, the system resumes responding to it. Subscribers that name that Handler may be active again.

## When a Subscriber Mask is Disabled

If you disable a Subscriber mask, that Subscriber mask is immediately inactive and cannot be executed.

You can still modify a disabled Subscriber mask.

If you re-enable that Subscriber mask, the system resumes responding to it; and the Subscriber mask is active.

# More about Selected Event Types

This section adds information about selected event types that is not captured in the table in The 29 Event Types on page 309.

## More about Selected Global Event Types

### Extend Tools Menu

Extend Tools Menu event type is a direct parallel to the existing capability for legacy PXs to appear in the client's **Tools** menu. Use of this event type will place the Event mask in the **Tools** menu.

There is no relevance to Pre and Post conditions, so this event type only triggers on Post.

The event mask's Name is what appears in the **Tools** menu. Note that the listing of PXs in the **Tools** menu is alphabetical, with legacy PXs and new event mask Names sorted together.

### Scheduled Event

Scheduled Event allows for the execution of any Event Subscription at a time you select.

All standard fields in event masks are available. Selecting Scheduled Event adds the **Frequency**

field, similar to Scheduled Event in ACS with the same settings. Object Type is Not Applicable.

Only Post event is possible for this event type.

The timestamp (Time that is displayed) is in GMT time zone.

# More about Selected Generic Object-based Event Types

## Extend Actions Menu

Extend Actions Menu event type is a direct parallel to the existing capability for legacy PXs to appear in an object's **Actions** menu. Use of this event type will place the Event mask in the correct object's **Actions** menu.

There is no relevance to Pre and Post conditions, so this event type only triggers on Post.

The event mask's Name is what appears in the **Actions** menu.

## Create Object

Besides the **Create** menu in Web Client, there are several other ways to initiate object creation: through the Actions shortcuts (as in an Item > **Actions** > **Create Change**); through a Table action; through the Agile SDK; or through Agile Web Services.

Use of Save As to create objects is covered by a separate event type.

| | |
|---|---|
| **Note** | When a user creates a Project by means of Create Program from Template action, the Save As Object event type is triggered, not Create Object event type. |

Entering Required Fields during the "Create" process is not considered an "Update" action, and so does not trigger the Update Title Block event type.

## Delete Object

The "Delete" action occurs in Java Client through the **Delete** button or, in Web Client, by **Actions > Delete**. This action can also be called by SDK or Web Services.

When an object is first deleted, it is recognized as an Initial, or "soft", delete. The object is not yet removed from the database and can be retrieved through Recycle Bin searches. After opening the object from a Recycle Bin search, the user can select Delete again to trigger a final delete ("hard" delete).

*Additional Attributes*

- **Delete Type** – Soft, Hard

  For an Agile business object that does not support hard delete, there is no risk in selecting Hard since the event itself will never trigger.

  RFQs class supports only Hard delete. RFQ Responses class does not support either type of delete.

## Get File

The Get File event type is triggered in the following cases:

- on a business object's **Attachments** table, user hits **Get** button
- on a business object's **Attachments** table, user clicks a filename link:
  - if the file is a viewable file, Agile Viewer is launched; no event type is triggered.
  - If the file is not a viewable file, the system performs the Get File action and the Get File event type is triggered; in general, from Java Client the Get File event type is triggered on File Folder objects, while from Web Client it is triggered on business objects.

Also, in Java Client, the number of Get File triggers depends on whether the "Download the files in one ZIP file" option is selected or not: if selected, one Get File trigger fires; if this option is not selected, multiple Get File triggers are fired.

## Update Relationship

The Update Relationship event type is triggered whenever a Relationship is added, removed, or directly edited on a business object's **Relationships** tab. Editing can include editing any field on the row, including the Relationship Rule.

The "Relationship" use case is treated as a discrete event type (compared to Update Table) because the target object is significant, that is, any PLM object can be in Relationship to any other PLM object.

This event type is not triggered due simply to the Relationship Rule being met. For example, let the rule for an ECO be: set an ECR to Closed status when the ECO moves to its Implemented status; so, when the ECO moves to Implemented, the condition of the rule is met, and Agile PLM attempts to move the ECR to Closed. If the ECR is moved to Closed, then the Change Status event type is triggered; whereas, if the move to Closed fails, then the Promotion Failure for Workflow event type is triggered.

*Additional Attributes*

- **Source Object Type** – select from list of object types that can trigger this event type when a user modifies the Relationships tab.
- **Target Object Type** – select from list of object types that can be updated as a result of the Relationships tab update.

  **Important** To set up a Subscriber mask for a Notification handler when the Event mask is based on Update Relationship event type, the object type of Source Object Type and the object type of the Notification mask must match. If multiple Source object types under the same class hierarchy are selected, the Update Relationship-based Event mask can be bound to a Notification mask whose object type is common among all selected Source object types.

## Update Table

Not all tabs of all objects support the Update Table event type.

### *Inclusions*

- Update Table event type captures user-initiated object table edits. In general, this includes Items > **BOM** table, Changes > **Affected Items** table, **Attachments** table, and so forth.
- Mass Update from a file folder object causes Update Table to be called for the relevant actions. Different events are triggered for each object's **Attachments** tab.
- A Changes > **Redline** table for **BOM**, **Manufacturers**, or **Attachments** triggers an event on the Item, not on any Changes table.
- For PCM-specific, PPM-specific, or "Common Services" objects.
  - Supported in **Attachments** only.

### *Exclusions*

- The **Relationships** table warrants its own event type, as noted above.
- In general, Update Table event type excludes those kinds of tables that are Read Only, for example, all **Where Used** and all **History** tables.
- Secondary table updates are also excluded, for instance, **Changes** and **Quality** tables, because the primary action is taking place elsewhere.
- Modifications to the **BOM** and **Manufacturers** tables do not trigger an update when an ECO is released – the trigger is used on the Changes > **Redline** tables or on **Change Status**.

*Additional Attribute*

**Table** – This multilist field contains all relevant Tables that are valid for the object and that are enabled for Events; tables are listed once they are implemented for Events. All implemented tables display for the object, even if they are not made Visible for users within Administrator.

There is no **<All>** table action. As a multilist field, customers can select multiple values.

## Update Title Block

The Update Title Block event type is triggered when a user directly edits any fields on an object's **PageOne** (that is, **Cover Page**, **General Information**, or **Title Block**), **PageTwo**, or **PageThree**. It does not trigger based on system-written values, for example, releasing a Change Order to update Lifecycle Phase.

The new Attribute Change Management feature of Change Orders and Manufacturing Orders are treated consistently with a **BOM** or **AML** update. Update Title Block event type is invoked only because of direct data entry, not because the Change was released with Attribute Change Management.

Redlining the **Title Block** on a Change Order or Manufacturer Order will trigger the Update Title Block event on the Item, not on the Change. A property of the Event Context Object indicates whether the update is occurring as a Redline; see *Agile PLM SDK Developer's Guide*.

LDAP updates do not invoke the Update Title Block event type.

# More about Selected Specific Object-based Event Types

## Change Status for Sourcing Object

The PCM solution does not use the standard Agile workflow. Change Status for Sourcing Object event type captures various changes to the statuses of PCM objects.

*Additional Attributes*

- **Action** – In general, this is the list of Lifecycle Phases for PCM Objects.
  - List for Requests for Quote
  - List for Sourcing Projects

# More about Workflow-based Event Types

When a workflow is created by the administrator, the only object type that can be chosen is a base class. Events are designed to be more definable than that; the system therefore uses the workflow's *Matching Criteria* property.

For all of the Workflow-based event types, the *Workflow* property is added to the Event mask. This property follows these rules:

- For all but Change Status for Workflow event type, the *Workflow* value can be **<All>**; when **<All>** is selected, *Object Type* property is disabled.
- When, however, a class-based workflow is chosen, if that workflow's *Matching Criteria* property has one value, the system uses that value as the *Object Type* of the Event mask.
- If the workflow's *Matching Criteria* property has multiple values, the system uses them to provide a dropdown list from which the administrator selects the *Object Type* for the Event mask.

## Approve for Workflow

*Additional Attribute*

**Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Audit for Workflow

Besides a user starting a Status Audit or a Release Audit, the Audit for Workflow event type can be initiated through SDK or Web Services. (For more information, see *Agile PLM SDK Developer's Guide*.) System-generated audits that occur during status changes do not trigger this event. Any failure information is included in the Event Context Object; see *Agile PLM SDK Developer's Guide*.

*Additional Attributes*

- **Type** – Status (audit), Release (audit)
- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Change Approvers or Observers for Workflow

There is no "affected status" for this event type.

*Additional Attributes*

- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section
- **Action** – Add Approver or Observer, or, Remove Approver or Observer

## Change Status for Workflow

Besides a Change Status in a workflow taking place because of user action, this event type recognizes Change Status in these cases: autopromotion within the server; when a Relationship Rule is matched which forces a Change Status; when “If Rejected, Set Status To” status property is used to change status. This event type triggers whenever a workflow moves from one status to another.

The Pre-event is before the object exits the "From" status. The Post-event is after the object arrives in the "To" status. This event type is structured in this way because there is no valid condition between statuses – the Change Status transaction is a "black box" to the programmer.

*Additional Attributes*

- **Workflow** – value cannot be **<All>** or blank; there is a single list of valid workflows for the selected object. Once a workflow is chosen, populating *Object Type* property observes the rules stated at start of this section.
- **Status – From** – list of statuses from the selected workflow; **<Any>** is valid and indicates any status
- **Status – To** – list of statuses from the selected workflow; **<Any>** is valid and indicates any status

## Comment for Workflow

*Additional Attribute*

**Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Escalation for Workflow

Since this event type is predicated on a system action that must complete (regarding the status property Review Escalation Period), the Pre-event is not available, so Trigger Type is Post only.

*Additional Attribute*

- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Promotion Failure for Workflow

Since this is a system action which must complete, the Pre-event is not available, so Trigger Type is

Post only.

*Additional Attribute*

- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Reject for Workflow

*Additional Attribute*

- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

## Reminder for Workflow

Since this event type is predicated on a system action that must complete (regarding the status property Reminder Period), the Pre-event is not available, so Trigger Type is Post only.

*Additional Attribute*

- **Workflow** – populating this property and *Object Type* property observes the rules stated at start of this section.

# Chapter 29

# Scripting and Sample Event Subscriptions

**This chapter includes the following:**


- Overview to Scripting in Agile PLM ..... 335
- Using Groovy Scripting Language ..... 336
- Logging of Scripts ..... 337
- Checklist for Specifying Script-based Event Subscriptions ..... 338
- Sample Event Subscriptions with Scripts ..... 338

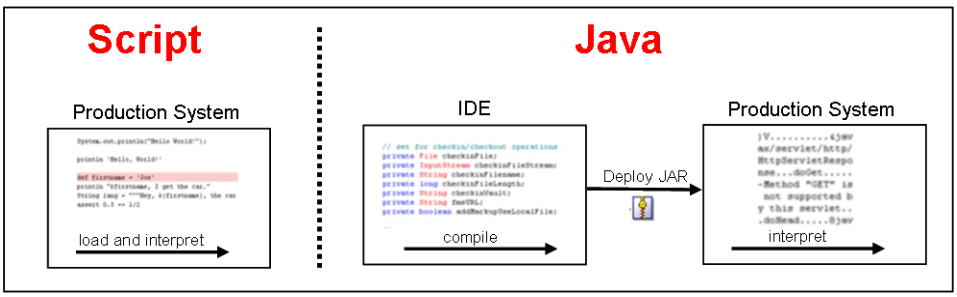

## Overview to Scripting in Agile PLM

This chapter provides an introduction to the Scripting capability, which permits scripted instructions of any length and complexity in Event Handlers. The scripting capability is fully embedded in the **Event Management** framework, with a server-side scripting engine that runs inside the PLM server (Agile Application Server). The script code is stored in a Large Character field in the database. You have access to the Script APIs and the Agile SDK, so you can write SDK programs.

The use of the Agile SDK is a key aspect to exploiting the potential of automated Events. The SDK programmer will find technical information about PLM Events in the *Agile PLM SDK Developer's Guide*.

The sample Event Subscriptions later in this chapter are models for the administrator's understanding of configured Event Subscriptions; they can help a non-programming administrator become familiar with scripting in Groovy (however, the samples are not specific "tutorials" in script-writing). While it is not necessary that the Agile-system administrator be able to write scripts for Handlers, if you do have the assistance of a programmer, you must be able to describe to that person what you want to achieve with the Event Management framework in Agile PLM.

The source code should be available in the production system at runtime, so script code is delivered in plain text, not in object code.



# Using Groovy Scripting Language

To contribute to Script PX Handler masks, the script-writer must know how to write scripts in Groovy, how to use the Script API, and how to use Agile Java SDK.

Ideally, the contributing programmer must understand the basics of Event Subscriptions and Handler masks. (A programmer who has not read the previous chapter, "Event Management", should start with Handler Types on page 307 and Working in Event Handlers Node on page 315.)

A scripting language is an excellent means to write user-readable and user-modifiable programs that perform simple operations and control the execution of other programs.

Groovy is a dynamic language for the Java platform. Its many features are inspired by languages such as Python, Ruby, and Smalltalk, making them available to Java developers using a Java-like syntax. With Groovy's simplified syntax, your code becomes easy to read and maintain.

These and other factors led to the selection of Groovy as the supported scripting language for the Agile PLM Event Management framework. Groovy is the only scripting language that is supported in Agile PLM 9.3.x.

There are many resources for learning Groovy, both in print and electronic media.

## Choosing Script-based Handlers

Scripts are normally dynamically typed (type checked at runtime) instead of statically typed (type checked while compiling). You should use or specify Script-based Handlers over Java-based Handlers in these cases:

- whenever there is a premium on user- (that is, administrator)-readable programs,
- cases where the business logic will need to be updated without explicit compiling or deploying,
- and especially for rapid prototyping of business logic; once validated, you may want to convert a Script PX to Java PX for better performance.

## Steps to Develop or Update Handler Masks

The steps that are needed to develop or update Handler masks is another factor to consider.

**To develop a Script PX Handler Mask:**

1. Define requirements (see Checklists below).
2. Write Groovy script.
3. Associate completed Script PX Handler mask into an Event Subscription.
4. Test the Handler.

**To develop a Java PX Handler Mask:**

1. Define requirements (see Checklists below).
2. Write Java code.

3. Compile code and test PX.
4. Deploy PX on Agile Application Server.
5. Associate completed Java PX Handler mask into an Event Subscription.

**To update a Script PX Handler Mask:**

1. Revise the Groovy script.

**To update a Java PX Handler Mask:**

1. Update the Java process extension.
2. Compile code and test PX.
3. Deploy revised PX on Agile Application Server.

## Editing Scripts in the Handler Mask

A Handler mask based on the Script PX handler type is an effective script editor.

On the **General Information** tab of a Handler mask, you can copy-and-paste your script for direct editing in Administrator.

Some editing features are available, particularly the **Validate Script** button, which performs a syntax check of script code, such as proper syntax with parentheses, "Equal" signs, and simple text errors.

The **Where Used** tab shows all Subscriber masks that use this script-based Handler mask.

The **History** tab displays the editing history of the script, such as its creation and instances of modification.

# Logging of Scripts

There are several different types of logging of scripts, which are activated and logged differently.

- Systems Logging
  - The systemwide Preference "Agile Script Log Level" controls the amount of information that is written to the Script PX log.
- Object-related Scripts
  - Explicit logs are written into server-side LOG4J trace files.
  - There is one log file for each PLM user. The log filename resembles: ***<userID>*_agileScript.Log**.
  - Runtime error messages give information to the administrator.
  - Object-related script execution is logged in the object's **History** tab.
- Event Handler Monitor
  - The **Event Handler Monitor** provides information about handlers being run. See Event Handler Monitor on page 324.

- The systemwide Preference "Event Monitor Log Level" controls the level of script logs to the Handler Monitor.

# Checklist for Specifying Script-based Event Subscriptions

In the "Event Management" chapter, checklist questions were listed in the sections about creating Event masks, Handler masks, and Subscriber masks. Some of these questions are collected below for your use in describing desired Script PX handlers.

- **Event Mask:**
  - From a given Event being triggered, what is the result that you want to automatically happen?
  - Will the Handler mask apply to a single subclass? Will the Handler apply to multiple subclasses from a single class, but not all of its subclasses?
  - If the event involves a Change, which Workflow(s) does it apply to? Does it apply to all workflows or a specific workflow?
  - Are there any Event Type–specific parameters to consider? For example, for the event type Delete Object Action, is the Handler mask going to handle soft deletes or hard deletes?
- **Handler Mask:**
  - What broad outcome does the Handler need to accomplish? What kind of instructions must the system have to complete a successful automation?
  - If the Handler is to be a process extension, what is the proper handler type for this Handler mask?
  - Who is providing the Java code, SDK or Groovy script that will drive this Handler mask?
  - What Roles are needed to run the handler mask? Will the user performing the PLM action always have the right Roles?
- **Subscriber Mask:**
  - What Handler mask is being named in this Subscriber mask? What Handler do you want invoked when the Event of interest is triggered?
  - Which Trigger Type should be used, Pre or Post? Why do you want the Handler to invoke Pre or Post?
  - Which Execution Mode should be used, Asynchronous or Synchronous?
  - What is the Order in which the Handler mask should be invoked?
  - What Error Handling Rule should be used?

# Sample Event Subscriptions with Scripts

The following sample Event Subscriptions provide models of the various masks in several use cases. They demonstrate how a Groovy script drives certain Subscription outcomes.

Event Management has been deployed as a working "out-of-box" feature, but It is not assumed that

the PLM administrator knows – or has to learn – how to write scripts in Groovy. There is more technical documentation, including samples, about Agile Events and Scripting in *Agile PLM SDK Developer's Guide*. The full potential of Events can be realized with the assistance of a programmer working with that manual.

| **Note** | The solution-based PLM user guides may contain information about how the solution can be enhanced by the Event framework. This would be in the administrator configuration chapter or appendix (for example, in the *PG&C* and *PCM User Guides*). |
|---|---|

These sample events and Groovy scripts have been tested, but they are not guaranteed to work "verbatim" in any customer system without scrutiny and modifications.

# Update Item Description

In this case, when a user updates the Title Block of any business object from one of all the Items base-class (grandchild) subclasses, the item's Description field is updated. This can be done with the primary business objects of any of the PLM solutions, for instance, auto-updating the Content objects of a Project in PPM.

In this sample, the action that triggers this Event Subscription is when a user updates the Title Block of a part. If Description is set with, say, "Testing", and the user modifies Text07 to "SOA" and Text08 to "Project" and Saves, this Event Subscription replaces the existing Description value with "Testing SOA Project". The Handler's execution status can be verified from the Handler Monitor.

## Event mask: `[Update Title Block  Items]`

- **Event Type**: Update Title Block
- **Object Type**: Items

## Handler mask: `[Update Item Description]`

- **Handler Type**: Script PX
- **Script**:

```
import com.agile.agileDSL.ScriptObj.IBaseScriptObj
// add other import statements here
import com.agile.api.ItemConstants


void invokeScript(IBaseScriptObj obj) {
    //script body starts here.
    // get old item description value
    oldItemDescription =
    obj.getValueByAttId(ItemConstants.ATT_TITLE_BLOCK_DESCRIPTION);


    // get the first field from pageTwo
    field1 = obj.getValueByAttId(ItemConstants.ATT_PAGE_TWO_TEXT07);
    // get the second field from pageTwo field2 =
    obj.getValueByAttId(ItemConstants.ATT_PAGE_TWO_TEXT08);
```

```
    // set the new description
    obj.setValueByAttId(ItemConstants.ATT_TITLE_BLOCK_DESCRIPTION,
    oldItemDescription + " " + field1 + "¯" + field2);

    // get the new description
    newItemDescription =
    obj.getValueByAttId(ItemConstants.ATT_TITLE_BLOCK_DESCRIPTION);

    // log to Event Handler Monitor
    obj.logMonitor( "Old description: " + oldItemDescription + "| " +
    "New description: " + newItemDescription);
}
```

### Subscriber mask: `[Update Title Block Items Update Item Description]`

- **Event**: `[Update Title Block  Items]`
- **Event Handler**: `[Update Item Description]`
- **Trigger Type**: Post
- **Execution Mode**: Synchronous
- **Order**: 0
- **Error Handling Rule**: Continue

## Notify Create User when Item is Incorporated

In this case, when an item – an instance of any subclass in the Parts or Documents classes, and therefore of the Items base class – is incorporated, a notification is created and sent to the user who created that part or document.

More specifically, the action that triggers this Event Subscription is a user creates a part, and from the **Actions** menu selects **Incorporate**.

The item changes to Incorporated, and a notification is sent to the part's creator. The Handler's execution status can be verified from the Handler Monitor.

Note that this notification is generated by instructions from a script (a Script PX handler), rather than a Notification mask created by the administrator (as in last example in this set).

### Event mask: `[Incorporate Item]`

- **Event Type**: Incorporate Item
- **Object Type**: Items

### Handler mask: `[Incorporated Item]`

- **Handler Type**: Script PX
- **Script**:

```
import com.agile.agileDSL.ScriptObj.IBaseScriptObj
// add other import statements here
import com.agile.api.ItemConstants

void invokeScript(IBaseScriptObj obj) {
//script body starts here.

    // get create user
    createUser =
    obj.getValueByAttId(ItemConstants.ATT_PAGE_TWO_CREATE_USER)

    // send notification
    sendToList = [createUser];
    obj.sendNotification("Notify On Incorporate", false, sendToList,
    "Comments: send from Script handler");

    // log to Event Handler Monitor
    obj.logMonitor( "Notify On Incorporate sent to: " + sendToList);
}
```

### Subscriber mask: `[Incorporate Item Incorporated Item]`

- **Event**: `[Incorporate Item]`
- **Event Handler**: `[Incorporated Item]`
- **Trigger Type**: Post
- **Execution Mode**: Synchronous
- **Order**: 0
- **Error Handling Rule**: Continue

## Set Change Analyst

This Event Subscription is a model for automatically setting the Change Analyst when a workflow changes status.

More specifically, the action that triggers this Event Subscription is a user creates an ECO and sets the product lines to one from this list: Capricorn, Leo, Pisces, Scorpio, Taurus, Test, Virgo. The field for Change Analyst is left unfilled.

When the system (or a user manually) changes the status to Submitted, the Change Analyst for the ECO is set according to the product lines. The Handler's execution status can be verified from the Handler Monitor.

### Event mask: `[Change Analyst for Workflow]`

- **Event Type**: Change Status for Workflow
- **Workflow**: Default Change Orders

- **Object Type**: Change Orders
- **Status – From**: Pending
- **Status – To**: Submitted

## Handler mask: `[Set Change Analyst]`

- **Handler Type**: Script PX
- **Script**:

```
import com.agile.agileDSL.ScriptObj.BaseScriptObj;
import com.agile.agileDSL.ScriptObj.AgileDSLException;
import com.agile.api.ChangeConstants;

/**
* Sets the change analyst upon setting an ECO to submitted.
*/
BaseScriptObj invokeScript(BaseScriptObj obj) {
    def PRODUCTLINES_ATTID =
    ChangeConstants.ATT_COVER_PAGE_PRODUCT_LINES;
    def CHANGEANALYST_ATTID =
    ChangeConstants.ATT_COVER_PAGE_CHANGE_ANALYST;
    def CAMap = [
        'Capricorn':'user1',
        'Leo':'user2',
        'Pisces':'user3',
        'Scorpio':'user4',
        'Taurus':'user5',
        'Test':'user6',
        'Virgo':'user7'
        ]

    def toStatus = obj.getToStatus();
    def workflow = obj.getWorkflow();
    def changeAnalyst = obj.getValueByAttId(CHANGEANALYST_ATTID);

    // update change analyst based on the product lines if the next
    status is CCB and CA list is not filled yet
    if(workflow == "Default Change Orders" && toStatus == "Submitted" &&
        (changeAnalyst == null || changeAnalyst == "") ){
        def productLines = obj.getValueByAttId(PRODUCTLINES_ATTID);
        if(productLines == null || productLines == "") throw new
        AgileDSLException("Product Line cannot be null, please fill
        in!");
    else
        obj.logMonitor("Current productlines are " + productLines);
    def productLinesList = productLines.tokenize(";")
        // Set change analyst based on the product lines
    changeAnalyst = CAMap.subMap(productLinesList).values().min();
        obj.setValueByAttId(CHANGEANALYST_ATTID, changeAnalyst);
        obj.logMonitor("||Set change analyst to " + changeAnalyst);
    }
}
```

### Subscriber mask: `[Change Analyst for Workflow Set Change Analyst]`

- **Event**: `[Change Analyst for Workflow]`
- **Event Handler**: `[Set Change Analyst]`
- **Trigger Type**: Post
- **Execution Mode**: Synchronous
- **Order**: 0
- **Error Handling Rule**: Continue

## Add Document to New Part

This Event Subscription is a model for automatically creating a Document and adding it to the Assembly BOM any time a part is created. It could be set up for any single subclass from the Parts class, as it is here, or for any object in all the Parts-class subclasses (by populating Object Type field with Parts Class).

More specifically, the action that triggers this Event Subscription is a user creates a part in Web or Java client.

When you click on the part's **BOM** tab, you see a document has been added to the part and the part has been changed to an assembly. The Handler's execution status can be verified from the Handler Monitor.

### Event mask: `[Create Parts]`

- **Event Type**: Create Object
- **Object Type**: Parts

### Handler mask: `[Add Doc to New Part]`

- **Handler Type**: Script PX
- **Script**:

```
import com.agile.agileDSL.ScriptObj.IBaseScriptObj;
import com.agile.agileDSL.ScriptObj.AgileDSLException;

import com.agile.api.IAdmin;
import com.agile.api.IAgileClass
import com.agile.api.IAgileSession;
import com.agile.api.IAutoNumber;
import com.agile.api.IItem;
import com.agile.api.INode;
import com.agile.api.ITable;
```

```
import com.agile.api.ItemConstants;
import com.agile.px.EventActionResult;
import com.agile.px.IEventAction;
import com.agile.px.IEventInfo;
import com.agile.px.ICreateEventInfo;
import com.agile.px.EventConstants;
import com.agile.px.ActionResult;

import java.text.SimpleDateFormat;
import java.util.*;

/**
*Post event of create item action, create document and add the document
into BOM table of assembly.
* It's mainly using SDK
*
*/

void invokeScript(IBaseScriptObj obj) {
   IAgileSession session = obj.getAgileSDKSession();
   IEventInfo req = obj.getPXEventInfo();

   try {
      int eventType=req.getEventType();
      int triggerType=req.getEventTriggerType();
      if(eventType!=EventConstants.EVENT_CREATE_OBJECT &&
      triggerType!=EventConstants.EVENT_TRIGGER_POST)
         throw new AgileDSLException("The PX is only applicable for
         post create event");
      ICreateEventInfo info=(ICreateEventInfo)req;
      Integer newSubClassId=info.getNewSubclassId();

      IAdmin admin=session.getAdminInstance();
      IAgileClass partclass=admin.getAgileClass(newSubClassId);
      IAgileClass
      docclass=admin.getAgileClass(ItemConstants.CLASS_DOCUMENT);

      // get the part object
      IItem part = (IItem) session.getObject(partclass,
      info.getNewNumber());
      if(part==null)
         throw new AgileDSLException("The part is not
         created:"+info.getNewNumber());
      if(partclass.getSuperClass().getId().equals(ItemConstants.CLASS_P
      ARTS_CLASS)){
      IAutoNumber docNumber = docclass.getAutoNumberSources()[0];
      //create a new document
      IItem doc = (IItem) session.createObject(docclass, docNumber);
      //add the document into BOM table of the new part.
      ITable tab = part.getTable(ItemConstants.TABLE_BOM);
      tab.createRow(doc);
      obj.logMonitor("Succeed to add document '"+  doc.getName() +"' to
      assemlby '"+part.getName()+"'");
         }
```

```
        else
        obj.logMonitor("The PX is only applicable for creating part
        object.");
        } catch (Exception e) {
        e.printStackTrace();
        throw new AgileDSLException(e);
    }
}
```

### Subscriber mask:

- **Event**: [Create Parts]
- **Event Handler**: [Add Doc to New Part]
- **Trigger Type**: Post
- **Execution Mode**: Synchronous
- **Order**: 0
- **Error Handling Rule**: Continue

## Notify when Affected Items Table is Updated

This event is a model for automatically creating and sending a notification (the Notification mask has been created in the **Notifications** node) any time the **Affected Items** table of a Change is updated.

It could be set up for any single subclass from the Change Orders class, or for any object in all the Change Orders-class subclasses (by populating Object Type field with Change Orders Class), or for any object in all the Changes base class (by populating Object Type field with Changes Base Class).

In this case, when the system modifies the **Affected Items** table (this is always an automatic process) of an instance of the subclass (and object type) ECO, a notification is created and sent to the users named on it. The Notification's execution status can be verified from the Handler Monitor.

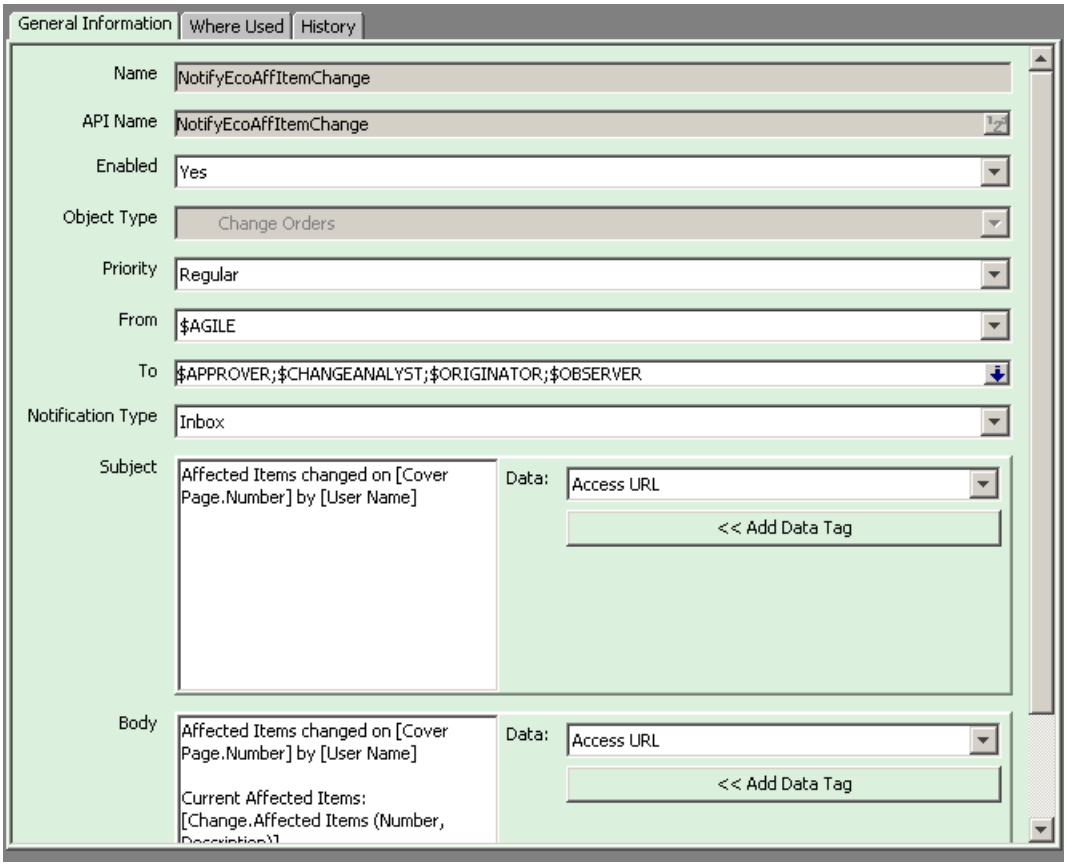
### Event mask: **[Update Table ECO]**

- **Event Type**: Update Table
- **Object Type**: ECO
- **Table Name**: Affected Items

### Notification mask: **[Notify ECO AffItem Change]**

- **Object Type**: Change Orders

- Here is a screenshot of the Notification mask:




## Subscriber mask: `[Update Table ECO Notify Item ECO AffItem Change]`

- **Event**: `[Update Table  ECO]`
- **Event Handler**: `[NotifyItemECOAffItemChange]`
- **Trigger Type**: Post
- **Execution Mode**: Asynchronous
- **Order**: —
- **Error Handling Rule**: —

# Chapter 30

# Locations and File Management

**This chapter includes the following:**


- Configuring Web Client Notification ..... 347
- Configuring Java Client Notification ..... 348
- File Management in Agile PLM ..... 348


The Locations window displays information about the location of the application and various servers in your Agile PLM configuration.

On the **General Information** tab of the Locations window, you can specify the following properties with appropriate URLs:

- Web Server URL
- Java Client URL.
- Online Help Manuals URL

**Note** These properties should only be changed by you, the administrator, with previous confirmation from your IT people.

Properties that are also seen on the **General Information** tab but are not editable were populated when the Agile PLM system was installed:

- Application Server URL
- DB Server Hostname
- DB Server Username
- Name (an informal name of the database server)

## Configuring Web Client Notification

Agile PLM users whose Preferred Client property is set to Web Client will receive automatic email notifications that contain a link to the Web Client. You can specify the Web Client location in the **Web Server URL** setting.

**Important** Agile's Product Cost Management (PCM), Product Governance & Compliance (PG&C), and Product Portfolio Management (PPM) solutions operate only in Web Client. Users who work primarily in those areas should have their Preferred Client set to Web Client.

**To configure Web Client notification by modifying the Web Server URL:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.

2. Enter your Web Server URL in this format: http://Web **Server:<port>/Agile/PLMServlet**

| **Note** | The Web Server URL is case-sensitive.<br>Ports are usually configurable. A port listed herein is either an industry-standard port or a default port. If the Web server uses the standard HTTP port (80), the port is not needed in the URL. |
|---|---|

3. When you are finished, click **Save**.

# Configuring Java Client Notification

Agile PLM users whose Preferred Client property is set to Java Client will receive automatic email notifications that contain a link to the Java Client. You can specify the Java Client location in the **Java Client URL** setting.

| **Note** | Although you can access the Java Client through a proxy server, the Java Client URL must be the location where Java Client is deployed on the application server for email notifications to work. |
|---|---|

**To configure Java Client notification by modifying the Java Client URL:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.
2. Enter the Java Client URL appropriate for your application server

   **Oracle Application Server:**
   http://<appserver>:8888/JavaClient/start.jsp

   **WebLogic Application Server:**
   http://<appserver>:7001/JavaClient/start.jsp

| **Note** | The Java Client URL is case-sensitive. |
|---|---|

3. When you are finished, click **Save**.

# File Management in Agile PLM

Administrator settings on the **Server Settings > Locations > File Manager** tab are important in relation to attachment files and where originals and copies are found. This is documented in File Management in Agile PLM on page 348.

The Checksum internal utility is a security measure against improper handling of files. Reset privilege and Checksum Computation preference are described in Handle File Checksum on page 350.

## Agile File Management Server

There are two main components to Agile File Manager (AFM): the file server and the file vault. When a file is added to Agile, it is assigned an internal Agile identifier (ID) number by the file server and added to the file vault. It is not stored in the file vault under its original filename: mapping information is maintained in the Agile PLM database, and the filename is modified per Filename Prefix systemwide preference.

When a user requests a file through an Agile PLM client (which requires privileges such as Get, View, or Checkout), the request is routed to the file server, which looks up the file's ID, retrieves the file from the file vault, and sends it to the user.

### File Vault

The file vault contains all file attachments stored by AFM. It is recommended that a separate server or storage device be designated as the Agile File Vault to store attachments or files. This is particularly useful when taking advantage of application clustering because each server in the cluster needs access to the same file vault.

### Using FileLoad and Indexing New Files

If the Indexing property is set to Synchronous and a large number of files are added to the File Management file vault using Agile FileLoad, the indexing process can take a long time. It has been found that setting this field to Manual increases the loading speed for a large number of files. Or, if the files being added tend to be large documents, set the Indexing property to Scheduled and set up an indexing schedule for once every 3 or 4 hours. If most of the new files are relatively small, the Indexing property can be set to Synchronous.

## Distributed File Management

Due to the geographically dispersed nature of the global enterprise, multiple AFM servers can be deployed in a distributed configuration for efficient distribution of product content. A Distributed File Management (DFM) setup allows you to manage files efficiently at remote locations. Deploying DFM servers reduce download time by placing Agile PLM files closer to where they are needed, and allowing users configure which file manager to use. Agile PLM supports a large number of DFM servers.

### Optimized Replication Systemwide Preference

When there are more than two File Managers, this systemwide preference (**Preferences** node) allows the PLM system to keep track of the File Managers that contain each file. During replication, the requesting FM only contacts other FMs that contain the file, and downloads a given file from the nearest FM (which is determined by shortest 'ping' to all the FM servers). Set to Enabled or Disabled, the latter can be set to troubleshoot problems with the optimizations.

### Using the AutoVue for Agile Server in a Distributed Environment

If AutoVue for Agile is used, an AutoVue for Agile Server should be installed locally with each DFM server. The local viewer server can be installed on the same machine as the DFM. If local users are accessing Agile PLM from outside the firewall, a proxy is recommended in the DFM configuration.

### How Distributed File Manager Works

Agile's file management servers have a peer-to-peer relationship. When a user requests a file, the request is directed to that user's configured file manager (see Configuring a File Manager on page 352). If the file is found, it is served to the user. If the file is not found at that location, the FM sends a request for the file to its peer file servers. The peer file server who has the file sends it back. The local file server saves it to its vault and serves it to the user.

A DFM scenario might be as follows: the AAS is installed in San Jose, California, and a DFM server is installed in Tokyo, Japan. A user from the Tokyo site selects a file attachment to view. The DFM in Tokyo determines that the file is located in San Jose and downloads the file, copies it to the file vault in Tokyo, and serves the file to the user through the View Server. Another user selects the same file attachment. Since the file now resides in the Tokyo DFM vault, it is served directly to the user.

You can install multiple file vaults, or DFM servers. Multiple file vaults are defined on the **Server Settings > Locations > File** Manager tab. On the **Preferences** tab for each Agile PLM user, you can specify the preferred file server for that user.

**Note** If your Agile PLM system is using multiple DFM servers, your users must use the full domain URL when logging in. If a user does not use the full domain URL, she will be asked to provide login information again when he performs a file operation.

# File Management Security

There are two components to file security: server security and client access security. There is also a utility called Handle File Checksum in Java Client that permits users to be warned of problems with improperly accessed files and a means to fix them.

## Server Security

Content in Agile file vaults must be protected from deletion or modification by unauthorized users. Agile recommends allowing access only to Agile PLM administrators. System users who access files through the clients do not need Add and Read privileges to the Agile file vault or file directory because Agile file servers retrieve files for users, this task is not performed by individual users.

## Client Access Security

Whether you access files from Java or Web Clients, your files are secure. When the client is run behind the firewall, the files are transferred behind the firewall, which secures your files from outside intervention. When clients access files from outside the firewall, Secure Sockets Layer (SSL) communications protocol is supported.

## Handle File Checksum

File Checksum is a feature in Agile that warns users when they encounter a file – say, an attachment to a business object – that may have been accessed improperly. In both Java and Web Clients, the **Attachments** tab of business objects and the **Files** tab of file folders have a field called **Has Checksum Error** . When a file on an attachment row has a checksum error, the Has Checksum Error field displays the same symbol.

In support of the Checksum feature, the administrator can manage file validation with a systemwide preference and a privilege mask:

- **Checksum Computation** systemwide preference sets whether to enable or disable checksum computation for attachment files. Note that the default setting is Disabled, which does not allow for this security feature to function. (It is possible that server speed is improved when this preference is set to Disabled, but that may not be as important as enabling the Checksum functionality.)

- **Reset File Checksum** privilege mask, which is built on the Reset privilege. All users who have been assigned this privilege mask will be notified of any file with a checksum problem. (Note that this privilege mask is not included with any out-of-box Agile role.) You can add the privilege mask to any existing role; at first, however, it may be safest to add it to specific individual's role, or perhaps to your Administrator role and to your User Administrators.

  The notification provides such information as the offending file's filename, the file folder(s) to which it is associated, the File Manager that hosts the file, and the file's location.

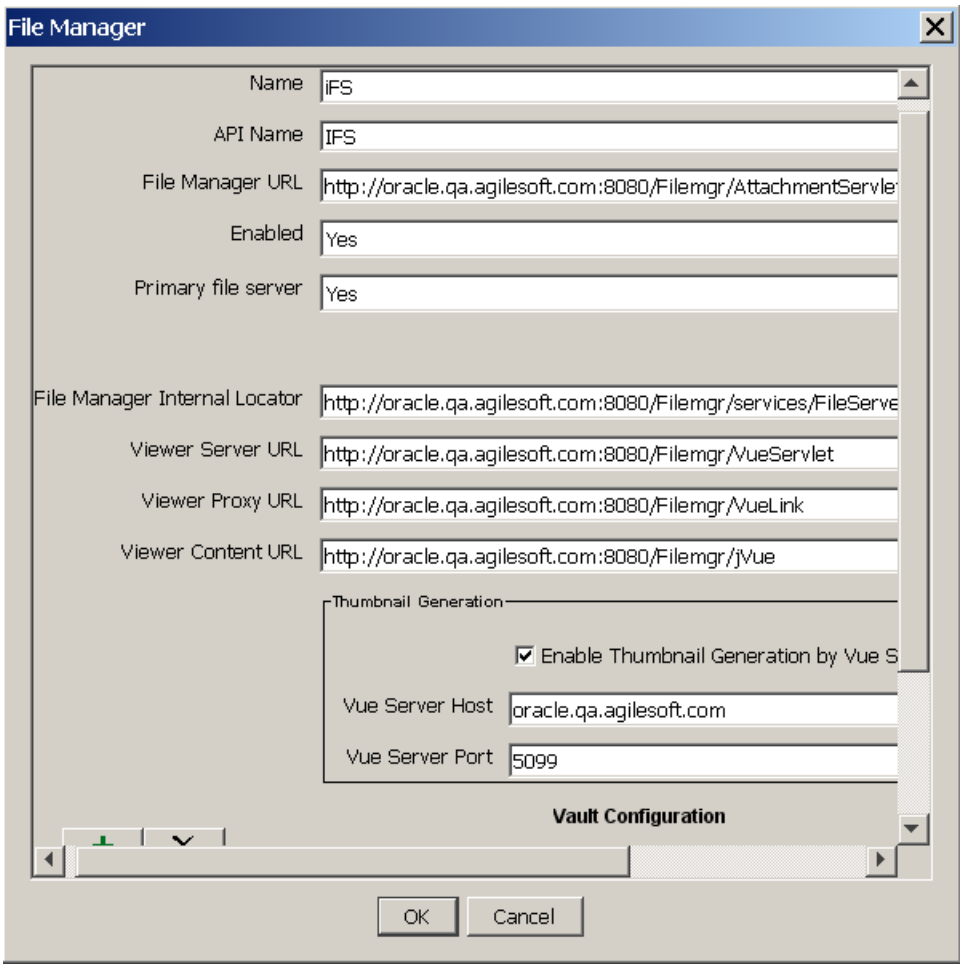
Besides the Reset File Checksum privilege, you will need to assign Read and Modify privileges for file folders to enable users to resolve checksum errors.

Checksum errors are resolved at the level of the file folder. Use the notification link to open the file folder, then click the ! in the offending file's row and use the dialog to resolve the error. You can click one of three options:

1. **Delete the file** – Deletes the file attachment itself across the Agile system. Use this option when you believe that the file is corrupted and irrecoverable, even across DFMs.
2. **Delete only invalid files** – This option can be used to attempt to fix the checksum issue:
   - Deletes the file from File Managers where the file status is Invalid.
   - Resets the checksum flag.
3. **Upload a new file to replace the problematic file in iFS vault** – Replaces the current file with a new file and resets the checksum to the value of the new file in the database. The file's ID in the database and the filename in the vault remain unchanged.

# Configuring a File Manager

On the **Server Settings > Locations** > **File Manager** tab, you can define multiple file managers.




When you click the **File Manager** tab and either click the **New** button or double-click one of the property fields, you see the first four fields in the table below. Click the **Advanced** button to see the other fields. From the advanced view, you can click the **Standard** button to return to the standard view.

**Important** You must re-start all the File Servers that have been added when you make changes on the File Manager tab, or change Direct Image File type for Thumbnail, or create a new one, for changes to take effect.

If a URL is specified in the documentation or release notes, copy the upper- and lower-case characters exactly, including the full domain name specified.

| Field | Description |
|---|---|
| Fields in the “Create a File Manager” dialog and File Manager “Standard” view | |

| Field | Description |
|---|---|
| **Name** | The informal name of this instance of File Manager. Use a descriptive name that is easy for you and your user administrators to understand, especially in a DFM (multiple) server configuration. This name will appear in the Preferred File Server dropdown list on each user's Preferences tab. |
| **API Name** | Unique systemwide identifier for objects and can be autogenerated. You can specify an API name, but internal consistency is enhanced by accepting the system-generated name. |
| **File Manager URL** | The file server URL that Agile Web Client will connect to. The format is: http://<proxy/loadbalancer>:<port>/<fileserver_virtual_path>/AttachmentServlet |
| **Enabled** | Enables or disables the file management server. If you want to create the file manager now and enable it later, select No. |
| **Primary File Server** | Select Yes or No from the dropdown list to make this server the primary file server or not. The primary file server should be co-located with the application server, that is, in the same LAN.<br>Setting a different file manager to be "primary" – setting this field to Yes when another file manager is already the primary – will automatically re-set this field in the previous primary file server to No. |
| Click the **Advanced** button to view and edit these fields: | |
| **File Manager Internal Locator** | The file application server URL for the File Manager or other component application (such as Agile SDK) to use. The format is:<br>http://<File *Manager host>:<port>/<fileserver_virtual_path>/services/FileServer* |
| **Viewer Server URL** | The viewer server (jVue server) URL that Agile Web Client will connect to. The format is: http://<proxy/loadbalancer>:<port>/<fileserver_virtual_path>/VueServlet |
| **Viewer Proxy URL** | The viewer proxy URL that the Viewer Server uses to communicate with the File Manager. If local users are accessing Agile PLM from outside the firewall, a proxy is recommended in the DFM configuration. The format is:<br>http://<File *Manager host>:<port>/<fileserver_virtual_path>/VueLink* |
| **Viewer Content URL** | The location from where the Viewer Applet is downloaded. The format is: http://<proxy/loadbalancer>:<port>/<appserver_virtual_path>/jVue |
| **Thumbnail Generation** | This area of the File Manager Advanced dialog contains a few settings:<br>▫ **Enable Thumbnail Generation by Vue Server** checkbox: check to enable "Thumbnail" graphics<br>▫ **Vue Server Host**: <*Vue Server host*><br>▫ **Vue Server Port**: <*Vue Server port*><br>**Note** Enabling Thumbnail generation with these settings is also enabling metafile generation. |

**To create a new File Manager:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.
2. Click the **File Manager** tab.
3. Click **New**. The Create a File Manager dialog appears.
4. Fill in the fields for **Name**, **API Name**, **File Manager URL**, and **Primary File Server**, which are described

in the table above.

| Note | If your company is using AutoVue for Agile, an AutoVue for Agile Server should be installed locally with each DFM server. |
|---|---|

5. The **Enabled** field must be set to Yes for that server to be put into active use.
6. Click the **Add New Vault** button to add and configure a new file vault.
7. When you have defined the new file manager, click **OK**.

   Once you have created this new instance of a file manager, you can open it and add other server definitions, as described in the next task.

## Creating a Custom File Vault

If you are using Java Client to upload references to files stored in a Custom file vault, you must set up the vault first. A Custom vault is “Read Only” and is not available to users to check out and modify attachments; it is used for uploading files using Agile FileLoad (which is documented in *Agile PLM Import & Export Guide*).

**To set up a Custom file vault:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.
2. Click the **File Manager** tab.
3. Double-click the entry to display the File Manager dialog box.
4. Click the **+** button to add a new vault.
5. In the Vault Type field, select Custom.
6. In the Description field, type a description of the vault.
7. In the Base Storage Directory field, enter the primary location where the files are stored. See next subsection
8. In the Purge Directory field, enter the primary location where the purged files are moved to. See next subsection.
9. Click **OK**.
10. Restart the Agile File Manager.

### *Base Storage Directory and Purge Directory Fields*

The Base Storage Directory default location is \files. The location can be a shared network storage directory, such as a Storage Area Network (SAN).

| Important | Do not specify a mapped location. Instead, specify the actual machine name and directory, like this: |
|---|---|

\\fileserver\files

The Purge Directory field indicates where purged (deleted) files are automatically moved. The default location is agile_home\files\purge. The location can be a shared network storage directory, such as a Storage Area Network (SAN).

**Important** Do not specify a mapped location. Instead, specify the actual machine name and directory, like this:

\\fileserver\files\purge

## Modifying a File Manager Definition

**To modify a File Manager definition:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.
2. Click the **File Manager** tab.
3. Double-click the row of the file manager you want to modify. The File Manager dialog appears.

   Be sure to navigate to the correct server field – use the **Advanced** button as needed.
4. When you have finished your modifications, click **OK**.

## Deleting a File Manager Definition or Removing a File Server

**To delete a file manager definition:**

1. Under **Server Settings,** double-click **Locations**. The Locations window appears.
2. Click the **File Manager** tab.
3. Double-click the row of the file server you want to modify. The File Manager dialog appears.

   Be sure to navigate to the correct server field – use the **Advanced** button as needed.
4. You can simply delete the reference to the server in the field – the file server definition.

   To remove a configured server, select a row and click the **Remove Selected Vault** button. If you have rights to that server, it will be removed.
5. When you are finished, and click **OK**.

# Setting the Preferred File Manager for a User

If your Agile PLM system uses Distributed File Management, each user should specify the preferred file manager to use. For best performance, specify a file server in the same location as the user. For example, users in the United States should select a file manager located in the United States and not one located in China.

**To set a user's preferred file manager in Java Client:**

1. Under **User Settings,** double-click **Users**. The Users window appears.
2. Double-click the user's name in the list. That user's window appears.
3. Click the **Preferences** tab.
4. For **Preferred File Manager**, select a local file manager from the list.
5. Click **Save**.

**To set a user's preferred file manager in Web Client:**

1. Click **Tools and Settings > Administration > Users**. The Users page appears.

2. Click the user's name in the list. That user's page appears.
3. Click the **Preferences** tab.
4. Click **Edit**. The fields become editable.
5. For **Preferred File Manager**, select a local file server from the list.
6. Click **Save**.

# Purging Older Versions of Files

If AutoPurge is enabled for your Agile PLM system (see Viewer and Files on page 265), the Agile File Manager automatically purges old versions of files. A background thread starts every day at 6:00 am local time that determines what files should be purged. If there are such files, a folder named by date and time is created in the purged files location and the files are moved into it. If no files need to be purged, the dated folder is not created. You can change the default interval between the hard-coded time and the time that the purge actually takes place. This interval is set in Attachment Purging Task; see Task Configuration on page 384.

| | |
|---|---|
| **Note** | The Agile File Manager does not permanently remove purged files from the system. Your company's IT personnel must handle that task. |

## Use Case for Attachment Purging Task

If there happened to be server failure during the time that was configured for attachment purging, the task would not run that day. For example, if the task runs at 6:00 am every day but the server shut down between 5:00 am and 9:00 am, the task would not run because the Task Lookback Window is set to 120 minutes, and 9:00 am is a greater value of minutes from 6:00 am. The workaround is to temporarily set Task Lookback Window to a value that is larger than 180 minutes (in this case), say, to 200 minutes. Then the task will run when the server is restarted.

In general, the value for Task Lookback Window (**Server Settings** > **Task Configuration> <any task> > General Info** page) should always bigger than the value for Task Interval to make sure that no event is skipped.

# Chapter 31

# Database

**This chapter includes the following:**


- Configuration of Database Settings ..... 357
- Database Properties ..... 357


## Configuration of Database Settings

From the **Database** node you can view and configure Agile PLM's systemwide database settings.

**To modify a Database setting:**

1. Under **Server Settings**, double-click **Database**. The Database window appears.
2. In any editable field, enter a new value or select from the dropdown list.
3. When you have finished, click **Save**.
4. Restart the Agile Application Server to activate the change.

**Caution** If you change the **Notification Enabled** setting, it takes effect immediately. Changing any other Database setting requires that you restart the Agile Application Server for the new setting to take effect. All active users' connections will be terminated when the Agile Application Server is restarted.

## Database Properties

The following table lists Database properties.

| Property | Description | Default |
|---|---|---|
| **Discovery Privilege** | Enabling Discovery Privilege allows all Discover privilege masks for this Agile PLM system to function.<br>If Disabled, all Discover privilege masks and Discovery-related SmartRules are disabled. If you are not using Discovery functions, you can improve your system's performance by setting this property to Disabled. | Enabled |
| **GUID** | The unique ID for your company's Agile PLM database. This property is not editable. | (GUID) |
| **Name** | The name of the main Agile PLM database; you can rename the node, using up to 29 letters. | Agile Database |

| Property | Description | Default |
|---|---|---|
| **Notification Enabled** | Whether email notifications are sent to users.<br>If this property is disabled (= No), all other email notification settings are ignored.<br>This setting is especially important with regard to **Event Management**: all "event-based" notifications will *not* be active if Notification Enabled is set to No. | No |
| **Schema** | Database "username," or the name of the schema, that is used to log in to the database. This property is not editable. | (Schema name) |
| **Server** | The name of the current Agile PLM database server. This property is not editable. | (Server name) |
| **DB Version** | The version of Agile PLM database being used. This property is not editable. | (DB version) |
| **Mail From Domain** | Domain name for email notification, for example, mydomain.com. | (None) |
| **Mail From User** | The From email address that is used in Agile PLM email notifications. The value must be a valid email address, for example, AgileAdmin@mydomain.com. | (None) |
| **Mail Server External** | The name of the computer running your external email server. | (None) |
| **Mail Server Internal** | The name of the computer running your internal email server. | (None) |
| **Customer ID** | The unique ID for your company. This property is not editable. | (ID number) |
| **Year Cycle – Start Month** | First month of your company's fiscal year. | (Set during installation) |
| **Year Cycle – Start Day** | First day of your company's fiscal year. | (Set during installation) |

# Chapter 32

# LDAP

**This chapter includes the following:**


- Configuring WebLogic Server for LDAP with Agile PLM ..... 360
- Creating Users in an LDAP Directory ..... 365
- Enhanced LDAP ..... 366
- Synchronizing LDAP and Agile PLM ..... 366
- LDAP-controlled User Properties ..... 367
- Agile LDAP Configuration ..... 367


Many enterprises use the Lightweight Directory Access Protocol (LDAP) system, as well as a dedicated LDAP server, to create their user accounts. The **Server Settings** node folder now includes an **LDAP** node. When you open the **LDAP** node, the LDAP Configuration Editor appears. The administrator can change LDAP settings (especially the search filter), preview the results, and save the changes, all without having to re-start the server.

**Note** The LDAP node may not be visible in your out-of-box Administrator tree. If your company does not use an LDAP system, the node is not needed. The node is made visible through the AppliedTo capability; see Administrator Privilege and the AppliedTo Capability on page 229.

Agile PLM supports LDAP authentication through the Agile Directory Server Integration Module. You can integrate Agile with your existing directory server to manage your users in one place. This approach can be fully integrated into Agile PLM, for these supported directory servers:

- Oracle Internet Directory Server
- Microsoft Active Directory Server
- Sun Java System Directory Server

If you chose to manage your user accounts through a directory server (instead of the database) during installation, then all new users (except supplier users) are added, and certain user attributes are configured, only through the directory server. Supplier users are authenticated by the database and managed by the supplier administrator.

**Note** Agile Administrator has the capability of integrating aspects of your PLM system with Single Sign-On (SSO) capability. With SSO configured and deployed for your PLM system, a user that has signed in to the system once (for instance, through the corporate portal) is not prompted again by a "login" dialog (see Appendix A, Single Sign-On on page 413).

# Configuring WebLogic Server for LDAP with Agile PLM

Oracle Application Server supports the Agile-certified LDAP servers; however, Oracle WebLogic Server requires further configuration to support them, as detailed in the tasks below.

## Configuring WebLogic with Oracle Internet Directory

**To configure WebLogic with an Oracle Internet Directory Server:**

1. Open the Administration console page by typing the following URL:

   http://localhost:<port_number>/console

2. In the left pane, choose **Lock & Edit**.
3. In the left pane, choose **agileDomain > Security Realms > AgileRealm > Providers > Authentication**.
4. Click **New** and enter a name, such as "OracleInternetDirectory" for the OID authenticator and select **OracleInternetDirectoryAuthenticator** and click **OK**.
5. Click on the newly created OracleInternetDirectory authenticator and change the value of the Control Flag to **SUFFICIENT** and then click **Save**.

| **Important** | The following is a sample of the values and settings needed for your own configuration. Be sure to use the applicable settings for your specific company's needs. |
|---|---|

6. Select provider specifics from the current page and fill in the following entries with your relevant values:

- `User Name Attribute: cn`
- `Propagate Cause for Login Exception: Check the Box`
- `Principal: cn=orcladmin`
- `Host: <Machine Name where the OID running>`
- `User Object Class: person`
- `All Users Filter: (Specify the User Filter Name or leave blank objectclass=person)`
- `User Search Scope: Subtree`
- `All Groups Filter: <Specify the Group Filter Name or leave blank>`
- `Static Member DN Attribute: uniquemember`
- `Group from Name Filter: (& (cn=%g)(objectclass=groupofuniquenames))`
- `Static Group DNs from Member DN Filter: (&(uniquemember=%M)(objectclass=groupofuniquenames))`
- `Use Retrieved User Name as Principal: (check the box)`

- Results Time Limit: 0
- Cache TTL: 60
- Dynamic Group Name Attribute: uniquemember
- Credential: <Password given for OID login>
- Confirm Credential: <Password given for OID login>
- Group Search Scope: subtree
- Group Base DN: cn=Groups,dc=agile,dc=agilesoft,dc=com
- Dynamic Group Object class: (empty)
- User from Name Filter: (&(cn=%u)(objectclass=person))
- Cache Size: 32
- Dynamic Member URL Attribute: (labeledurl)
- SSLEnabled: (empty)
- Cache Enabled: (check the box)
- Connection Retry Limit: 1
- Connect Timeout: 0
- Parallel Connect Delay: 0
- User Dynamic Group DN Attribute: cn
- Static Group Name Attribute: cn
- User Base DN: cn=Users,dc=agile,dc=agilesoft,dc=com
- Follow Referrals: (check the box)
- Port: 389
- Ignore Duplicate Membership: (uncheck the box)
- Static Group Object Class: groupofuniquenames
- Group Membership Searching: unlimited
- Max Group Membership Search Level: 0

1. Click **Save**.
2. On the left pane click the button **Activate Changes** to activate all the changes made.
3. Log out from the console, and restart the WebLogic server and, if installed in a cluster, all managed servers to successfully have all the changes activated.

# Configuring WebLogic with Sun Java System Directory Server

**To configure WebLogic with a Sun Java System Directory Server:**

1. Open the Administration console page by typing the following URL:

   http://localhost:<port_number>/console

2. In the left pane, choose **Lock & Edit**.
3. In the left pane, choose **agileDomain > Security Realms > AgileRealm > Providers > Authentication**.
4. Click **New** and enter a name, such as “Iplanet” for the iplanet authenticator and select **iPlanet Authenticator** and click **OK**.
5. Click on the newly created iplanet authenticator and change the value of the Control Flag to **SUFFICIENT** and then click **Save**.

**Important** The following is a sample of the values and settings needed for your own configuration. Be sure to use the applicable settings for your specific company's needs.

6. Select provider specific from the current page and fill in the following entries with the following values:

- `User Name Attribute: uid`
- `Host: <machine name where the iplanet server is running>`
- `Principal: < uid=admin,ou=administrators,ou=topologymanagement,o=netscaperoot >`
- `User object class: Person`
- `User Search Scope: subtree`
- `Static Member DN Attribute: uniquemember`
- `Group From Name Filter: (|(&(cn=%g)(objectclass=groupofUniqueNames))(&(cn=%g)(objectclass=groupOfURLs)))`
- `Static Group DNs from Member DN Filter: (&(uniquemember=%M)(objectclass=groupofuniquenames))`
- `Results Time Limit: 0`
- `Cache TTL: 60`
- `Dynamic Group Name Attribute: cn`
- `Credential: <password given for the iplanet to authenticate>`
- `Confirm Credential: <confirm the password>`
- `Group Search Scope: subtree`
- `Group Base DN: ou=agile, dc=agile,dc=agilesoft,dc=com`

- Dynamic Group Object Class: groupofURLs
- User From Name Filter: (&(uid=%u)(objectclass=person))
- Dynamic Member URL Attribute: memberURL
- Cache size: 32
- Cache Enabled: select the check box
- Connection Retry Limit: 1
- Connect Timeout: 0
- Parallel Connect Delay: 0
- User Dynamic Group DN Attribute: < Fill this if you are using dynamic group>
- Static Group Name Attribute: cn
- User Base DN: ou=agile, dc=agile,dc=agilesoft,dc=com
- Follow Referrals: select the check box
- Port: 389
- Static Group Object Class: groupofuniquenames
- Group Membership Searching: unlimited
- Max Group Membership Search Level: 0

1. Click **Save**.
2. On the left pane click the button **Activate Changes** to activate all the changes made.
3. Log out from the console, and restart the WebLogic server and, if installed in a cluster, all managed servers to successfully have all the changes activated.

# Configuring WebLogic with Microsoft Active Directory Server

**To configure WebLogic with Microsoft Active Directory Server:**

1. Open the Administration console page by typing the following URL:

   http://localhost:<port_number>/console

2. In the left pane, choose **Lock & Edit**.
3. In the left pane, choose **agileDomain > Security Realms > AgileRealm > Providers > Authentication**.
4. Click **New** and enter a name, such as “ActiveDirectoryServer” for the ADS authenticator and select **ActiveDirectoryAuthenticator** and click **OK**.
5. Click on the newly created ActiveDirectoryServer authenticator and change the value of the Control Flag to **SUFFICIENT** and then click **Save**.

**Important** The following is a sample of the values and settings needed for your own configuration. Be sure to use the applicable settings for your specific company's needs.

6. Select provider specific from the current page and fill in the following entries with the following values:

- User Name Attribute: sAMAccountName
- Principal: Administrator@enterprise.uab.edu
- Host: 10.176.138.35
- All Users Filter: (objectclass=person)
- User Search Scope: subtree
- All Groups Filter: (Keep it Empty)
- Static Member DN Attribute: member
- Group from Name Filter: (Keep it Empty)
- Bind Anonymously On Referrals (Uncheck this Box)
- Static Group DNs from Member DN Filter: (&(member=%M)(objectclass=group))
- Results Time Limit: 0
- Credential: < password given for the ADS to authenticate>
- Confirm Credential: < password given for the ADS to authenticate>
- Group Search Scope: subtree
- Cache Size: 32
- User from Name Filter: (keep it empty)
- Dynamic Member URL Attribute: (keep it empty)
- Connection Retry Limit: 1
- Connect Timeout: 0
- User Dynamic Group DN Attribute: (keep it empty)
- Static Group Name Attribute: cn
- User Base DN: OU=agile,DC=enterprise,DC=uab,DC=edu

**Note** This is operating with the assumption that all Agile users are found under this path.

- Use Token Groups for Group Membership Lookup (Uncheck this Box)
- Port: 389
- Follow Referrals (check this Box)

- Propagate Cause for Login Exception (Uncheck this box)
- User Object Class: user
- Cache TTL: 60
- Use Retrieved User Name as Principal (uncheck the box)
- Dynamic Group Name Attribute: (keep the field empty)
- Group Base DN: OU=agile,DC=enterprise,DC=uab,DC=edu

**Note** This is operating with the assumption that all Agile user groups are found under this path.

- Dynamic Group Object Class: (keep the field empty)
- SSLEnabled: (uncheck the box)
- Cache Enabled (check this box)
- Parallel Connect Delay: 0
- Ignore Duplicate Membership: (uncheck this box)
- Static Group Object Class: group
- Group Membership Searching: unlimited
- Max Group Membership Search Level: 0

1. Click **Save**.
2. On the left pane click the button **Activate Changes** to activate all the changes made.
3. Log out from the console, and restart the WebLogic server and, if installed in a cluster, all managed servers to successfully have all the changes activated.

# Creating Users in an LDAP Directory

You have the following options for creating Agile PLM users:

- **Create all users in Agile PLM** – this can be done even if your company uses LDAP for its non-Agile applications.
- **Create all user objects using the corporate LDAP system** – the basic user data (for instance, user ID, first and last name, password, email address) are imported to Agile PLM, where each user's profile is completed.
- **Combine the two approaches** – your company may use LDAP for its employees that are assigned to use Agile PLM, but create non-employees (for instance, supplier users) within the Agile PLM system. In this case, the LDAP accounts are imported and validated, and there is no risk reconciling user data within Agile PLM.

**Note** You cannot create regular Agile PLM users (that is, Power or Concurrent users) *both* in Agile PLM and in LDAP. The only way to combine the two approaches is to use Agile PLM to create Restricted users only.

**Note** More specifically, you should not create multiple users with the same User ID (username), especially in the WebSphere environment. WebSphere uses an embedded LDAP adapter to authenticate users, and that adapter does not allow multiple users with the same ID.

Users created in Agile PLM with the "**(Restricted)**" supplier user roles are authenticated through the Agile PLM database.

## Configuring Multiple LDAP User Repositories

You can configure multiple LDAP user repositories for the security domain by repeating the configuration steps for the specific directory server.

Do *not* delete the AgileAuthenticator authentication provider: it is used to authenticate users against the Agile database. The Control Flag for the AgileAuthenticator must remain **Optional**. AgileAuthenticator must be the first authentication provider in the list if there are additional authentication providers, such as Sun Java System or Microsoft Active Directory.

# Enhanced LDAP

**To modify LDAP information:**

1. Under **Server Settings,** double-click **LDAP** node. The LDAP Configuration Editor window appears.
2. On the **LDAP Configuration** tab, double-click the row that names the LDAP server you want to populate. This is also where you can create a new LDAP server configuration.

   The **Edit LDAP** dialog presents the contents of the LDAP server you selected for your modification.
3. When you have completed entering modifications, click **OK**.
4. With the row selected (for the LDAP server you modified), click the **Preview** button. The **Preview Results** tab lets you preview the results of your LDAP query.

   When you click **Preview**, the server validates the LDAP configuration. If there are errors – such as inconsistent data across LDAP servers in a cluster, or duplicated users on multiple servers, or syntax errors – the server passes an exception back to the client, and you will see a popup window displaying the errors.

   - If the LDAP configuration contains clustered LDAP servers, the preview shows data from only one of the LDAP servers, since the data is identical across the cluster.
   - If the LDAP configuration contains multiple LDAP servers, the preview shows the union of data from all LDAP servers.
5. Click **Save** to save changes to the LDAP configuration. Again, the server validates the LDAP configuration. If any errors are detected, an error message displays and the exception prevents the Save operation from completing.

# Synchronizing LDAP and Agile PLM

In Java Client, the **Refresh Users from LDAP** button on the **Users** node toolbar window integrates

LDAP-originated users into the Agile PLM system. In Web Client, if you use LDAP, the **Refresh from Directory Server** button is enabled in the **Users** node under the More button menu options (under **Tools and Settings > Address Book > Users**) for the same purpose.

Refer to your LDAP documentation to import user accounts to Agile PLM.

After clicking **Refresh** (or **Refresh from Directory Server**) to automatically update the accounts for use in Agile PLM, complete each user object by populating Agile PLM user properties as required.

Synchronization of Agile PLM users with LDAP-created user accounts depends on the Agile PLM users' Login ID being equivalent to the LDAP user-accounts' Login ID. During an upgrade of Agile PLM systems, it assumes user authentication through the Agile PLM database.

There is another way to synchronize users between LDAP and Agile PLM: a script is included in the Bin directory called migrateUserToDB.bat/sh that you can run manually or on a scheduled basis. This script serves the same purpose as the **Refresh** (or **Refresh from Directory Server**) button in Java and Web clients, respectively.

**Important** Be sure to restart the application server after executing the script.

# LDAP-controlled User Properties

If your company is using LDAP for user accounts, the following properties will always be managed by LDAP. They cannot be edited in Agile PLM.

- UserID
- First Name
- Last Name
- Password
- Email Address
- Title
- Work Phone
- Mobile Phone
- Fax Number

**Note** If an LDAP directory is used to create and manage Agile PLM users, by default users do not require a separate approval password. However, you can uncheck the **Use Login Password for Approval Password** property for a user and enter an approval password.

You can set up **Account Policy** functionality on the LDAP server. For more information, see your LDAP system documentation.

# Agile LDAP Configuration

The LDAP Configuration Editor (**Server Settings > LDAP** node) is used to configure your integration

between Agile PLM and your company's LDAP system.

Click the **New LDAP** icon and fill in the properties (defined below). You can double-click an existing configuration (row in the table) to bring up the Edit LDAP dialog. You can remove an existing configuration by selecting its row, clicking the **Delete LDAP** icon, and following the prompt.

You can define multiple sets of parameters to configure integration with multiple directory servers.

**Important** Verify all settings in LDAP Configuration Editor with your LDAP administrator.

# Directory Service Connection Parameters

Connection parameters include the hostname, port, protocol, account name and filter. The account name is used to connect to the directory server during synchronization, so it must have the appropriate privileges. The filter is used to select only a subset of the users defined in the directory server as Agile users.

# Configuration Parameters Defined

The properties in the New LDAP configuration page are described in the table below:

| Field Name | Description |
|---|---|
| ID | Unique string identifying the LDAP server. The string must be less than 30 characters and cannot be changed once in use. |
| Description | Information about the server configuration |
| Agent | The Directory Server used for authentication; valid values are SunONEDirectory or ActiveDirectory |
| URL | The URL for the authentication agent |
| Domain | The authentication string when using Active Directory Server in the format of <username>@<auth.domain.name> |
| Username | Username (does not need to be the LDAP Administrator) |
| Password | Encrypted password of the user |
| User Path | Tree under which all Agile users can be found; this property should be set to the node closest to the root of the Directory Tree structure; any user that is not found under the subtree starting at this node should not be on the Agile system. |
| Search Scope | Scope of search for Agile users under the user-path node; valid values are ONE_LEVEL or SUB_TREE; this property should be set to ONE_LEVEL only if all users in the organization are directly under the User Path node |
| Search Filter | Search filter for Agile users under the <user-path> node; this must be a valid LDAP search filter that matches all Agile users under the scope defined by <auth.ldap.user.path> and <auth.ldap.user.search.scope>; users not matching this filter are considered invalid users on the Agile system; a valid LDAP search |

| Field Name | Description |
|---|---|
| | filter must be enclosed in parentheses. |
| Mechanism | Authentication mechanism supported by the directory server; valid values are "simple" or "strong" |
| Group Path | This property should be set to the node closest to root of the Directory Tree structure; it's the path in which group search starts |
| Group Scope | Valid values: ONE_LEVEL, SUB_TREE; similar to <search-scope> |
| Group Filter | This must be a valid LDAP search filter that matches the LDAP groups where all expected users to be used in Agile are contained.<br>When using the Group Filter to synchronize all users and user groups from the LDAP server, the <user-path> and <search-scope> must be set for the Search Filter.<br>With User Group Sync function enabled. You cannot remove/ add users for LDAP Synched User group on User's Tab |
| Group Membership | Used for authentication to check if the current user belongs to the specified group (defined in group-filter); using "%M" to designate current login user DN. |
| Dynamic Group Filter | This must be a valid LDAP search filter that matches the LDAP Dynamic groups that contain all expected users to be used in Agile |
| Failover Links | Links that point to alternate LDAP servers that the system tries to access, in listed order, when the primary LDAP server fails |
| Fail Attempts | Specifies the maximum number of attempts to access other servers that the system can make when the LDAP server fails |
| Disable Agile User If Not Found In LDAP | Valid values: TRUE or FALSE<br>When set to TRUE and the administrator runs the "migrateUsersToDB -p" command, the Agile active users (migrated from LDAP and made active) who are either removed from LDAP or not in the current LDAP search path are disabled. |

### *Mapping LDAP Attributes*

**To map an Agile attribute to an LDAP-system attribute:**

1. In the User-LDAP Attributes Mapping or Usergroup-LDAP Attributes Mapping field, click Add + to create a new entry.
2. From the New Attribute Mapping dialog, choose an Agile ID (that is, attribute) from the dropdown list.
3. In the LDAP ID field, type the name of an attribute from your LDAP system that you want mapped to the selected Agile attribute.

For example, "Date09" might map to an attribute called "End Date".

## Multiple Directory Server Support

It is possible to indicate multiple directory servers. This is useful if you have users in multiple domains that need access to Agile, or if you have backup directory servers to provide fail-over support.

On the Edit LDAP page, create one or more nodes depending on how many directories need to be supported; click the **Preview Result** tab: it will display all the users on the first node or server. Now, if you stop the services of this server, then the Preview Result will display the users from the backup or secondary server. (Remember to restore services to the primary server.)

If a backup or secondary directory server is configured, the integration module tries the backup server if access to the primary server fails.

## Configuration Scripts

One directory server can be configured during the Agile installation. Additional directory servers can be configured manually after installation. Agile provides scripts to enable configuration after installation. These scripts are located in the *agile_home*\agileDomain\bin directory:

- encryptpwd — ldapconfig.xml needs an encrypted password for the directory server user; this script generates an encrypted password based on the existing user password
- checkLDAPConfig — use for checking LDAP configurations; all errors should be fixed, if encountered
- migrateUserstoDB — use to migrate a user from LDAP to the Agile database; this script allows you to update existing users and to create new users in the database.

## Synchronizing Users and User Groups

This feature applies to iPlanet server.

- LDAP user groups and group members can be synchronized into the Agile system using <group-filter> to "synch" the static users in the user group, and <dynamic-group-filter> to "synch" dynamic group users. To enable this function, make sure that **auth.ldap.group** is enabled (set to True) in **agile.properties** file.

  **Note** With User Group "synch" function enabled, you cannot remove or add users on a user group's **Users** tab that have been synchronized (that is, where users have been added to a user group via LDAP).

# Chapter 33

# Systemwide Preferences

**This chapter includes the following:**


- Modifying Preferences ..... 371
- Default Systemwide Preferences ..... 371
- Details of Selected Preferences ..... 377
- Languages tab ..... 379


## Modifying Preferences

The **Preferences** node lets you view and configure Agile PLM's systemwide preferences and Languages.

**Note** There is no connection between the **Preferences** node, which are systemwide server settings, and User Preferences, which are found on every user object (**User Settings > Users** node **> [any user] > Preferences** tab). However, a subset of the User Preferences are "system-related"; again, these do not pertain to settings in **Preferences**.

**To modify Preference settings:**

1. Under **Server Settings,** double-click **Preferences**. The Preferences window appears.
2. In any editable field, enter a new value or select from the dropdown list.
3. When you have finished, click **Save**.

## Default Systemwide Preferences

The following table lists and describes the Agile PLM Preferences. They are listed in alphabetical order.

| Preference Name | Description | Default |
|---|---|---|
| **Add User's Default Site to New Items** | For enterprises that utilize "Sites" information, this Preference can be set so the Default Site of a user is automatically added when the user creates an item. | No |
| **Agile Drive** | Controls the user created Agile Drive access to Agile objects and documents outside of Agile. See Agile Drive on page 377 for more information. | Disabled |
| **Agile Script Log Level** | Sets "level" of logging on an Agile system. A production system might be set to a lower level, such as Off, Error, or Warn. A development system would likely be set to a | Off |

| Preference Name | Description | Default |
|---|---|---|
| | higher level, such as All, Debug, or Fatal.<br>Settings are All, Debug, Error, Fatal, Info, Off, or Warn. | |
| **Allow Download of Productivity Components** | Controls whether your users will have access to the Java applets that permit advanced file uploading and other features, depending on what is allowed by the License Key your company purchased. Settings are No and Yes. See Allow Download of Productivity Components on page 390, on page 377. | No |
| **Allow Password Reset** | Lets you select whether Web Client users who forget their password can automatically receive a new one by clicking a link on login screen. See Allow Password Reset on page 377. | No |
| **Allow Unsecured Files** | Allows Agile PLM to manage pointers to unsecured URLs, that is, it allows attachments to be URLs (as opposed to files). If you select Yes, pointers to URLs are stored on the Agile Application Server computer, although Checkin and Checkout are disabled for referenced URLs. | Yes |
| **Checksum Computation** | Sets whether to enable or disable checksum computation for attachment files. Settings are Enabled or Disabled. See Handle File Checksum on page 350. | Disabled |
| **Content Service Maximum Retries** | Determines how many times, after the initial attempt, a transfer order object will attempt to make delivery to a failing destination before further delivery attempts are blocked. See Content Service Maximum Retries on page 378. | 5 |
| **Content Service Wait Time (in seconds)** | Determines how often the Content Server “wakes up” to check for transfer order objects to process. | 300 |
| **Default User Name** | Names of users appear with the first name appearing before the last name, or vice versa (with appropriate comma).<br>Changing Default User Name setting requires re-starting the server for the change to take effect. | FirstName LastName |
| **Display URLs as Hyperlinks** | Text and Multitext fields enable typed URLs to automatically convert to hyperlinks. Setting this to No will display URLs typed in Text and Multitext fields as plain text. Note that certain "URL" and "mail" fields in Agile are not controlled by this preference; they will always convert typed URLs to hyperlinks. | Yes |
| **Display User ID** | Determines whether or not User IDs (usernames) appear.<br>Changing Display User ID setting requires re-starting the server for the change to take effect. | No |

| Preference Name | Description | Default |
|---|---|---|
| **Enable BI Publisher for Reports** | Determines whether or not BI Publisher is used to generate reports in multiple output formats.<br>Setting this to Yes generates reports in multiple output formats using BI Publisher. Setting this to No means reports continue using existing reporting layout. | No |
| **Event Maximum Nested Levels Allowed** | Determines how many times any script will run. If you intend that scripts run multiple times to complete their designed operation, adjust this setting to the desired maximum times allowed. But other scripts will also run multiple times, with possible unintended effects. | 10 |
| **Event Monitor Log Level** | Sets the level of logging in **Event Handler Monitor** (**System Settings** > **Event Management**).<br>Settings are **All** (logs all invoked Subscribers), **Error** (logs only errors), or **Off** (does not display log information). | All |
| **Filename Prefix** | Agile File Manager uses a prefix that is attached to each file that is stored in the file vault. Type in the identifier that will be automatically added to every attachment file. The default is “agile”.<br>**Caution!** This effects of this systemwide preference are far-reaching and the value should not be changed without good reason. | agile |
| **Garbage Collection Wake-up Time (in seconds)** | The “garbage collection” is a thread for cleaning out idle threads in the cache. After “cleaning,” garbage collection is put to “sleep” for the number of seconds in this setting. 300 seconds (5 minutes) is a reasonable value. | 300 |
| **Load to CAD** | Pertains to Engineering Collaboration solution. When set to Enabled, a **Load to CAD** button appears and is functional in Web Client. Design objects can then be uploaded to your designers' CAD system. | Disabled |
| | | |
| **Maximum Query Results Displayed** | Sets a maximum number of search results that can be displayed (advanced or quick searches); can be any positive integer less than 5000, but 1000 is recommended.<br>If a user’s Max Rows Displayed user preference is set to fewer rows than this systemwide preference, the former value will override the latter value.<br>**Caution**: The Full Search Display privilege applies to results of report queries, not to ordinary searches. All users, whether or not they have the Full Search Display | 1000 |

| Preference Name | Description | Default |
|---|---|---|
| | privilege, will see the maximum number of search results specified in this property. | |
| **Maximum Report Results** | Determines maximum number of objects displayed in Agile PLM Custom Reports; note that it does not apply to Standard Reports.<br>This preference is overridden by Full Search Display (FSD) privilege (see Caution).<br>The upper limit to this field has been increased to 500,000 (a comma is not needed in the field).<br>Note that the "Export to Excel" feature of search results (in Java Client, from page of Search Results: **Export Search Result** menu icon > **Excel** or **CSV** ) regards this Preference: the number in this field limits the maximum number of results exported to Excel or CSV files. (This is not to be confused with this sequence accessed from Business Object: **Actions** > **Microsoft Excel** > **Export to Excel**, which does not use this setting.)<br>**Caution**: Users with the FSD privilege see all results of reports; also, all privilege checking is bypassed on users with this privilege when they view report results. Users without the FSD privilege see the maximum number of reports specified in this property. | 50,000 |
| **Maximum Running Reports** | Defines maximum number of reports that the application server will run simultaneously. | 50 |
| **Notification Wait Time (in seconds)** | Determines how often the system “wakes up” to check for notification emails to send. Default is 600. Valid setting is 30 to 3600 seconds. | 60 |
| **Notification Time Out (in seconds)** | This setting is the timeout time for the user transaction in the notification thread. Valid setting is 30 to 3600 seconds. | 1800 |
| **Object Cache Table Size** | Sets the relative size of the object cache table (Hashtable), which helps determine how objects are distributed in cache (smaller means more objects per row, larger means fewer objects per row). You can choose Small, Medium, or Large Table (default is Medium Table), but it is an internally used attribute used in server implementation; therefore, no recommendation | Medium Table |

| Preference Name | Description | Default |
|---|---|---|
| | regarding performance is offered for this setting. | |
| **Optimized Replication** | When there are more than two File Managers, allows the PLM system to keep track of the File Managers that contain each file. During replication, the requesting FM only contacts other FMs that contain the file, and downloads a given file from the nearest FM (shortest 'ping').<br>Settings are Enabled or Disabled. Disabled can be set to troubleshoot problems with the optimizations. | Enabled |
| **Recently Visited Folder Size** | Controls the maximum number of objects that each user can store. The list selection allows these values: 10, 25, 50, 100. | 10 |
| **Reference Designators Allow Range Expand Collapse** | Allows reference designators to be described as a range instead of a sequence. See Reference Designators Allow Range Expand Collapse on page 378. | Collapse |
| **Reference Designator Range Indicator** | The symbol that appears between lower and upper range values in reference designator statements. Available symbols are asterisk (*****), at (**@**), caret (**^**), plus (**+**), hyphen (**-**), colon (**:**), semicolon (**;**), tilde (~), or **None**. | Hyphen (-) |
| **Reminder/Escalation Weekend Setting** | In escalation and reminder periods of changes, determines how the system will count weekend days (Skip Saturday *or* Sunday, Skip Saturday *and* Sunday, or Don't Skip [= *count both weekend days*]). This preference is not part of Product Portfolio Management scheduling. | Don't skip |
| **Retain Local Files** | When adding an attachment to a business object or checking it in, determines whether local files (on user's computer) will be deleted or retained, or whether the user is prompted to choose to delete it (Yes) or not (No).<br>Settings are Delete, Retain, and Warning.<br>Note: This preference is applicable only in Advanced uploader, and is not applicable in Standard uploader: Standard uploader does not delete files automatically when this preference is set to Delete. | Warning |
| **Save As Attachments** | Permits one of two ways to replicate a file, or allows each user to be prompted to choose the method. Settings are Prompt, Create new copy of file(s), Do not copy file(s), and Reference existing file(s). See Save As Attachments on page 389. | Reference existing file(s) |

| Preference Name | Description | Default |
|---|---|---|
| **Save As BOM for Different Classes** | Indicates whether or not to copy **BOM** tab values when copying from one item to another.<br>See more information about the "Save As" preferences in Copy Values along with Tabs on page 378. | Values Copied |
| **Save As Manufacturer for Different Classes** | Indicates whether or not to copy **Manufacturers** tab values when copying from one item to another. | Values Copied |
| **Save As Page 2 for Different Classes** | Indicates whether or not to copy **Page Two** tab values when copying from one class to another. | Values Copied |
| **Save As Page 3 for Different Classes** | Indicates whether or not to copy **Page Three** tab values when copying from one class to another. | Values Copied |
| **Save As Page 3 for Different Subclasses** | Indicates whether or not to copy **Page Three** tab values when copying from one subclass to another. | Values Copied |
| **Signoff User Dual Identification Type** | Controls whether approve/reject signoff (including user signing off for "self" or for another user due to Escalation or Transfer Authority) requires a dual identification, or "second signoff."<br>Settings are N/A (Not Applicable, the default), User ID, and Login Password. Selecting N/A means that these situations will not require a dual ID; selecting either User ID or Login Password means a dual ID is always required to sign off.<br>For more information, see Signoff User Dual Identification on page 378. | N/A |
| **Threshold (in seconds)** | One of the settings that determines when certain system actions, such as cleanup and reset tasks, are performed on any object that is idle in the logged-in user session longer than the value of the setting. Valid setting is 30–3600 sec. | 300 |
| **Thumbnail Support** | Allows the system to present thumbnail representations for Agile objects in Web Client. | Disabled |
| **Web Client Timeout for Corporate Users** | This setting dictates the number of minutes corporate users can be idle in Web Client before Agile terminates the session. | 25 |
| **Web Client Timeout for Supplier Users** | This setting dictates the number of minutes supplier users can be idle in Web Client before Agile terminates the session. | 25 |

**Note** The proxy server for your Agile PLM system has a default connection timeout of 1500 seconds, or 25 minutes. For information on how to change the proxy timeout setting, see the chapter “Installing and Configuring Agile Web Components” in the *Agile PLM Installation Guide for Windows*.

# Details of Selected Preferences

This section contains details about selected systemwide Preferences.

## Agile Drive

Enabling the Agile Drive preference allows the administrator to access the Agile Drive system of managing the file attachments of Agile objects in the form of files and folders from your computer system, without requiring you to invoke the Agile application. It connects Agile's Distributed File Manager as a Network Place on your system, thus enabling Explorer-like operations, such as copy or delete, on file attachments of Agile objects. See Administering Agile Drive and *Agile Drive User Guide* for more information.

## Allow Download of Productivity Components

This preference controls whether your users will have access to the Java applets that permit advanced file uploading, download of multiple files and the Excel integration (in PG&C). If it is set to No, users will not see the File Productivity Preference in their own user profile. If it is set to Yes, users will see the File Productivity Preference and be able to choose their own method.

- The preference allows the following functions regarding attachment management. The systemwide preference operates in conjunction with each user’s File Productivity (user) Preference.
  - Advanced File Uploader
  - Automated checkin based on checkout location
  - Allow download of individual files instead of a Zip file in Web Client.
- The preference allows the Microsoft Excel-based Client to be integrated with Agile PG&C, which allows supplier users to provide compliance information to buyers while using Microsoft Excel.

## Allow Password Reset

The **Allow Password Reset** preference lets you choose whether Web Client users who forget their password can automatically receive a new one. The default setting is No, which forces a user to contact the administrator to reset the password. Java Client users who forget their password must contact the administrator for a new one.

When **Allow Password Reset** is set to Yes, the **Forgot your password?** link appears on Web Client’s login screen. Since **Preferences** are *systemwide* settings, every Web Client user sees the link. If a user clicks the **Forget your password?** link, he can enter his username and email address to receive a new random seven-character password. When the user logs into Web Client again, he is prompted to choose a new password.

## Copy Values along with Tabs

The several “Save As” preferences determine whether values are copied from the **BOM**, **Manufacturers**, **Page Two**, and **Page Three** tabs when you use Save As to copy an object from one class to another. By default, values are copied.

| | |
|---|---|
| **Caution** | There are exceptions to copying values during Save As operations. If an attribute in the source object does not exist or is not used in the target object, it is not copied even if the Save As preference is set to Values Copied. Also, if an attribute is a list value and the list ID of the source attribute is not the same as the list ID of the target attribute, the value is not copied even if the Save As preference is set to Values Copied. |

## Content Service Wait Time

The **Content Service Wait Time** preference controls how often the system processes transfer order objects. For example, if this setting is 300, every 300 seconds the system checks if there are any transfer order objects to process and begins to process them.

## Content Service Maximum Retries

The **Content Service Maximum Retries** preference controls how many times, after the initial attempt, a transfer order object will attempt to make delivery to a failing destination. For example, if this setting is 5, the transfer order object will make one attempt at delivery. If that attempt fails, the transfer order object will make up to 5 more attempts. After the last attempt, further delivery attempts are blocked.

## Reference Designators Allow Range Expand Collapse

This preference (RDAREC) determines how reference designators are displayed in the BOM table when *not* in edit mode. This preference affects only the end user display of the reference designator data stored in the Agile database.

- **Expand** — Displays ranges of reference designators in expanded format; each reference designator is listed and the range is not collapsed. For example, the range of reference designators beginning with **R1** and ending with **R5** is displayed as:

  **R1, R2, R3, R4, R5**

- **Collapse** — Displays ranges of reference designators in collapse or concatenated format. Ranges of reference designators (3 or more reference designators in a sequence) are indicated by the first reference designator in the sequence, the reference designator range indicator character, and the last reference designator in the sequence. For example, the range of reference designators beginning with **R1** and ending with **R5** is displayed as:

  **R1-R5**

## Signoff User Dual Identification

FDA-regulated or other companies may have a corporate policy that requires double authentication

of user identify in approving or rejecting change orders. The Signoff User Dual Identification preference controls whether approve/reject signoff requires a dual identification, or "second signoff."

This setting is applied to the Review and Released statuses of all default and custom workflows, that is, when the Signoff dialog is filled in by a user, the system verifies the electronic signatures before advancing the workflow to the Review or Released status. The preference comprises all situations when the user is signing off for himself, signing off as a member of a user group, signing off for another user due to Escalation or Transfer Authority, or a combination of signoff responsibilities. (The Signoff UI options are detailed in *Getting Started with Agile PLM*, the chapter about Workflow.)

The administrator must attend to a combination of settings to properly realize the "second signoff" functionality.

Setting this preference to N/A (Not Applicable, which is the default) means that signoffs require only a single signoff from any user in any capacity. The password already required to sign off is the Approval Password; this represents the first level of security in the signoff protocol.

The other two settings, User ID and Login Password, effectively enable the dual ID requirement for any user in any signoff capacity. They dictate that users also be required to enter their User ID (also called username), or the Login Password.

If you set this preference to Login Password, then it is imperative to ensure that every user's Use Login Password for Approval property is set to No; once you change all users' Use Login Password for Approval property to No, only then can you set the Login Password. (For bulk settings, contact your Agile representative for a script called SecondSignature.Zip.)

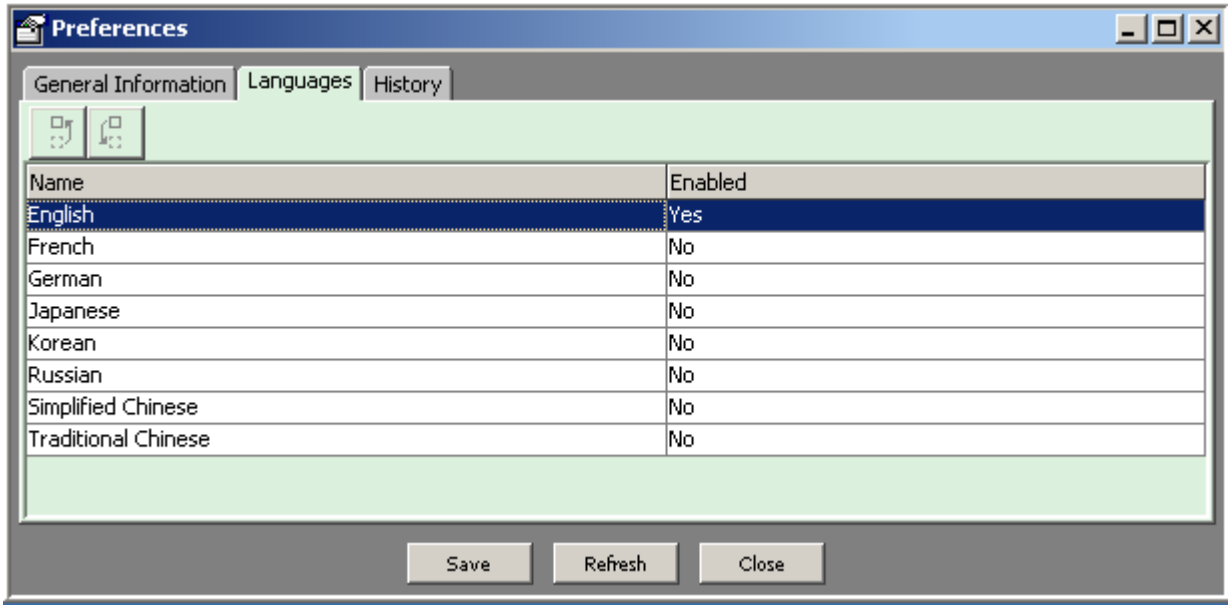
If you set this preference to User ID, you should set Display User ID preference to No

# Languages tab

The **Languages** tab lists the available languages for the Agile system. The administrator can select one or more languages at a time and enable or disable them using the Enable/Disable buttons.

| **Note** | By default, English is enabled and you cannot disable English. |
|---|---|

You will be able to see in two columns the name of the language and whether or not the language is currently enabled or disabled with a setting of Yes or No.




**Important** This list of supported languages is current as of May, 2010 and is subject to change. Please see the languages tab found in the **Preferences** node for your current installation supported languages.

Once you are finished with applying the settings you desire then click **Save**. At this point there will be a message alerting you that you must restart the server in order to apply the changes.

The Agile PLM server will successfully load translated labels and fields for languages that are enabled. The Administrator history is only generated for enabled languages.

**Note** User preferences for languages are limited to the enabled system languages.

For more information about Agile PLM Language settings, see Administrator Functions in Java Client on page 30.

# Licenses

The **Licenses** node appears under **Server Settings**. This node provides information about your company's Agile PLM license configuration.

Agile's "User Licenses" are tabulated in **General Info** tab. User licenses are assigned to each user by the Agile PLM administrator. The combination of licenses and assigned roles and privileges determines which specific tasks a user can perform in certain Agile PLM solutions.

**To view licenses information:**

1. Under **Server Settings,** double-click **Licenses** node. The Licenses window appears.

2. Click the **General Information**, **Modules**, or **History** tab.

# General Info Tab

Agile PLM has three types of user licenses: Named (formerly "Power"), Concurrent, and Restricted.

Restricted users are people outside your company (such as distributors and suppliers) who are given limited access to the Agile PLM system.

Named users and Concurrent users have access to the same Agile PLM functionality based on their roles and privileges and the company's licensing.

The key difference is that users with a Named license are not applied against concurrency counts. This means that a Named user can log in to the system at any time. All users who are not Named or Restricted are subject to concurrency counts, and may be locked out of the system due to a concurrency limit.

The **User Category** field in every user's profile displays which user license has been assigned to the user. For more information, see User Properties Defined on page 170.

The **General Info** tab of the Licenses window includes the information shown in the following table. You cannot edit this information.

| "Name" field | Description |
|---|---|
| **Count of Named Users** | Shows the number of users who are active. |
| **Count of Restricted Users** | Keeps track of how many Restricted User licenses have been assigned. |
| **Count of Concurrent Users** | Shows the number of Concurrent User licenses purchased by the company. Concurrent users can access all Agile PLM functionality purchased by the company, but are subject to concurrency counts, and may be locked out of the system due to a concurrency limit. |
| **Total Count of Users** | Keeps track of how many PLM users of all types have been created by the company. |

# Modules Tab

The biggest change to "licensing" in Agile PLM is the removal of Agile's "Server Licenses". An Oracle "license key" was obtained (at http://licensecodes.oracle.com/agile.html#PLM) and was input at install time.

Now the administrator uses the **Modules** tab of the **Licenses** node to simply select the PLM solutions that his company has purchased. This will enable those PLM Classes and other solution-based functions that pertain to your company's business, while disabling those classes and functions that do not pertain.

For Product Collaboration, select Yes from the dropdown list. Then open each of the remaining dropdown lists and select Yes or No, as appropriate. When you are finished and click the Save

button, you will be prompted "Please verify your company is contractually licensed to use the modules [PLM solutions] selected." You will need to restart the system for a module selection to take effect.

Product Portfolio Management (PPM) is the only PLM solution that can operate without Product Collaboration (PC); however, it is likely that even a company using only PPM will also use PC.

**Important** While it is possible to enable a PLM solution (by selecting Yes in **Licenses > Modules**) that your company has not purchased, it is not a Best Practice to attempt to do business using an unpurchased solution. You are likely working closely with an Oracle Consulting – Agile Practice representative, and you will likely need Support services at some point. However, if a problem you have encountered has been caused in the course of deploying and creating business objects from a PLM solution that your company has not purchased, you will find it more difficult to obtain the support for PLM that you and your company require.

If you encounter difficulties in enabling the PLM solutions that your company has purchased, please contact your Agile PLM representative.

# Chapter 34

# Task Monitor and Task Configuration

**This chapter includes the following:**


- Task Monitor ..... 383
- Task Configuration ..... 384


## Task Monitor

The **Task Monitor** node is used to track the progress of certain scheduled tasks. These include:

- **Report tasks** – standard and custom reports that are scheduled to be executed

  **Note** Administrator reports cannot be scheduled.

- **User tasks** – refreshes user list from Directory Server (for example, LDAP server)
- **Full Text Search task** – indexing of attachments for full text search
- **Attachment Purge task** – periodic purging of attachments in the DFM server
- **PPM-specific tasks** – there are several Activity tasks
- **PCM-specific tasks** – there are several RFQ Notification tasks
- **PG&C-specific task** – the Compliance Rollup task has been added to manage scheduled compliance rollups.

You can use the **Thread Status** dropdown list to display only the tasks matching the following criteria:

- **All** – shows all tasks
- **Scheduled** – shows tasks scheduled when it's generating schedule results.
- **Running** – shows tasks currently running; this is the default filter
- **Finished** – shows tasks that have been successfully executed
- **Failed** – shows tasks that failed during execution
- **Timed Out** – shows tasks that timed out before full execution. Compliance Rollup task cannot time-out.

**Note** Thread status criteria (except for All) do not apply to all types of tasks. For example, report tasks are listed in the Task Monitor window only when the selected **Thread Status** is All or Finished.

The following table lists the properties of the Task Monitor window. The properties are read-only.

| Property | Notes |
|---|---|
| **Name** | Name of the scheduled report (custom or standard). Null for all other tasks. |
| **Task Type** | Report Task, FTS Task, or Attachment Purge Task. |
| **Owner** | Name of the owner of the report schedule. Null for all other tasks. |
| **Time Executed** | When the task began. |
| **Status** | Current status of the task: Scheduled, Running, Finished, Failed, Timed Out |

The **Purge** button allows you to delete tasks listed in the Task Monitor window. The Purge action applies only to selected tasks that fall within the specified date range.

**To purge tasks from the Task Monitor:**

1. Select a Thread Status to list certain tasks, such as all tasks or finished tasks.
2. Select one or more tasks.
3. Click the **Purge** button .
4. Specify values for **From Date** and **To Date**. Tasks that fall within the date range will be purged. Click the button to select a date and time for each field.
5. Click **OK**.

# Task Configuration

From the **Task Configuration** node, you can manage various scheduled server-side activities such as reports and full text search indexing. Tasks never "time-out" before completion.

| Name | Description | Default interval (minutes) |
|---|---|---|
| **Refresh User List From Directory Server** | Updates the Agile PLM user list with information from the Directory Server. | 1440 (1 day) |
| **Report Task** | Checks if there are any scheduled report tasks to run. If so, it generates the report in the background, saves it on the file server, and sends an email notification to the report's owner and shared users. | 30 |
| **Activity Summary Task** | In PPM solution, identifies the list of activities scheduled for the coming week and notifies the owners in a summary format. | User-defined |
| **Full Text Search Scheduler** | Indexes the content in files for full text search capability. | 600 (10 hours) |

| Name | Description | Default interval (minutes) |
|---|---|---|
| **Attachment Purging Task** | Moves outdated files into a special folder and cleans up metadata in the database. The system is hard-coded to begin this process at 6:00 am local time; the default interval value of 1440 re-starts the process 24 hours later. Another value entered for this task alters the start time, then re-starts it at the new interval.See Use Case for Attachment Purging Task on page 356 for a particular use case of Attachment Purging Task. | 1440 (1 day) |
| **Activity Reminder Task** | In PPM solution, identifies the list of activities that should start today and reminds the owners of these activities. | User-defined |
| **Activity Health Task** | In PPM solution, updates the "health" for all activities. It also updates the actual duration, variance, and estimated variance for these activities. | 2 |
| **Compliance Rollup Task** | In PG&C solution, runs compliance rollup calculations.<br>**Note**: Even if the Product Governance & Compliance server license is disabled and not available, "Compliance Rollup Task" still appears in this list of tasks. | 30 |
| **Due Tomorrow Notifications Task** | In PCM solution, this sets how often the server checks for new RFQ-related notifications. | 2 |
| **Admin Cache Synchronization Task** | Synchronizes servers via **Web Client > Tools > Administration > Cache Health Monitor**. | 5 |
| **Schedule Event Task** | The settings for this task run in support of triggered events that use the Scheduled Event event type. | 30 |
| **Compliance Migration Task** | This task sorts all legacy compositions into three categories: Fully Disclosed, Partially Disclosed, or Undisclosed.<br>**Note** In PG&C solution, for upgrades in the 9.2.x releases, this task must be enabled and run *once*; then this task can be disabled and is not needed. | 1 (this task is run only once by upgrade customers) |

When you double-click a task in the **Task Configuration** page (anywhere in the row), the Task Configuration **General Info** window appears. It has the following properties; you can modify the properties and save the new settings.

| Property | Description |
|---|---|
| **Name** | Name of the task being configured. |
| **Task Interval** | Periodic time interval in minutes after which the server updates the status of the task.<br>In general, the value for Task Lookback Window should always bigger than the value for Task Interval to make sure that no event is skipped. |

| Property | Description |
|---|---|
| **Task Delay Time** | How many minutes the event is delayed before it starts to run. |
| **Task Lookback Window** | How far back in minutes the task manager is to look to retrieve the events that can be run now.<br>See Use Case for Attachment Purging Task on page 356 for a particular use case that involves temporarily re-setting Task Lookback Window.<br>In general, the value for Task Lookback Window should always bigger than the value for Task Interval to make sure that no event is skipped. |
| **Task Max Event Number** | Number of events that can be running at the same time on one server. |
| **Task Restart Upon Failure** | Indicates whether the event should be restarted after it failed. Yes or No. |
| **Task Load Across Server** | Indicates whether the event can be executed at the same time in different servers of a cluster. Yes or No. |
| **Task Disabled** | Indicates whether this task is disabled (Yes) or enabled (No).<br>**Notes:** In order for a given task to be executed, this property must be set to No. Also, when this property is changed for any task, it is no longer required to re-start the system. |
| **Task Start Time** | Allows you to specify the exact time a task should start. For example, you can specify that an Activity Reminder Task or an Activity Summary Task should be triggered at 8 a.m every morning.<br>**Note**: Once you reset the time, you must restart the server for the changes to apply. |

**To disable a task in the Task Configuration window:**

1. Double-click the task to open it.
2. In the **Task Disabled** list, select Yes.
3. Click **Save**.

# Chapter 35

# Administering Attachments

**This chapter includes the following:**


- Evaluating Your Company's Needs with Attachments or Designs ..... 387
- Administrator Settings Related to Attachments ..... 387
- Assign Roles and Privileges for Attachments ..... 392
- Document Management Enhancements ..... 398
- Generating Attachment Quick Access URLs ..... 399


## Evaluating Your Company's Needs with Attachments or Designs

As with other areas of Agile PLM administration, the most important step comes before setting properties in Administrator. You have to determine an overall "policy" that supports your company's uses of attachment files, file folders, and design models.

As you decide about company objectives, these are a few factors you might evaluate:

- Does your company produce CAD design and graphic files? If so, you will build from the Designs class. Please refer to the next chapter,
- Does your company need to organize and track attachments and bundles of attachments? This may indicate building from the File Folder subclass.
- Are there other special purposes or uses of Agile's attachment capabilities to take into account?
- Invisible or Visible file folders?
- How should attachment files be named? Are such files to be named in a dictated or organized system? Or is it alright to allow users to name attachment files and count on users to use file folders to search and find specific attachments?

## Administrator Settings Related to Attachments

This section collects various Administrator settings – SmartRules, systemwide Preferences, and a user preference – for convenience. Roles and privileges that pertain to attachments are discussed in a subsequent section. Refer to node-specific chapters for more information about the capabilities in Administrator.

### File Management in Agile PLM

Administrator settings on the **Server Settings > Locations > File Manager** tab are important in relation to attachment files and where originals and copies are found. This is documented in File Management

in Agile PLM on page 348.

The Checksum internal utility is a security measure against improper handling of files. Reset privilege and Checksum Computation preference are described in Handle File Checksum on page 350.

# How SmartRules affect Attachments

## Copy Files To Rev

The Copy Files To Rev smartrule controls automatic copying of attachment files to a new revision; that is, the pending item revision that is created when an item is added to the Affected Item table of a change. The system checks this SmartRule only when a pending change is created and not when attachments are added. Possible settings are Copy, Reference (the default), Disallow, Copy with Warning, and Reference with Warning.

The following table illustrates how each setting works in the case of the following example:

- Part 55, Revision B, **Attachments** tab table lists attached file Test.txt, version 3; that is:

  **File Name** = Test.txt, **File Folde**r = Folder004, **Folder Version** = 3.

- Part 55, Revision B is added to the **Affected Items** tab of a change object, thus creating pending revision C.

| Setting | How it works |
|---|---|
| **Copy** | For the item's new pending revision, Agile creates a new file folder and creates a new copy of the attachment file, which is placed in the new file folder.<br>So, when pending revision C is created, the **Attachments** tab table lists the same attachments as revision B, that is, revision C inherits the same attachments as revision B:<br>**File Name** = Test.txt, **File Folder** = Folder004, **Folder Version** = 3 |
| **Reference** | Agile uses the existing file folder and creates a new reference to it on the item's pending revision **Attachments** tab. |
| **Disallow** | When pending revision C is created, the **Attachments** tab table is empty. No attachment references are copied. |
| **Copy with Warning** | The same as the **Copy** setting, with the option of choosing no attachments for the new pending revision.<br>When you add items to the pending ECO, you are presented with a warning dialog that includes a row for each item you are adding. To add the item and to copy the attachments, check both the **Add** checkbox and the **Attachments** checkbox. To add the item with no attachments, check only the **Add** checkbox. |
| **Reference with Warning** | The same as the **Reference** setting, with the option of choosing no attachments for the new pending revision.<br>When you add items to the pending ECO, you are presented with a warning dialog that includes a row for each item you are adding. To add the item and to reference the existing attachments, check both the **Add** checkbox and the **Attachments** checkbox. To add the item with no attachments, check only the **Add** checkbox. |

If you set this smartrule to Disallow, files that were added to the Introductory revision after the object was associated with a change (that will result in released Rev A) will not be copied to subsequent revisions. Also, when a new Change is created, all the files that were added to the Introductory revision won't be added to the newly created revision.

### Check In Attachment With Different File Extension

This smartrule determines whether an attachment file that is being checked in can have a different file extension than the file that was checked out. This smartrule applies to all business objects.

This rule is overridden when a business object's attachment is Latest-x and the file extension in the attached file folder is changed and is therefore different than the extension of the business object's checked-out file.

## How Preferences affect Attachments

Systemwide preferences are documented in Preferences on page 371.

### Filename Prefix

Agile File Manager uses a prefix that is attached to each file that is stored in the file vault. You can type in the identifier that will be automatically added to every attachment file. The default is "agile".

### Optimized Replication

When there are more than two File Managers, this preference allows the PLM system to keep track of the File Managers that contain each file. During replication, the requesting FM only contacts other FMs that contain the file, and downloads a given file from the nearest FM (which is determined by shortest 'ping' to all the FM servers).

Settings are Enabled or Disabled. Disabled can be set to troubleshoot problems with the optimizations.

### Save As Attachments

Here are further details on the available choices for the Save As Attachments preference. When Save As is used to create a new business object will include the same attached files; your choice is whether to create new copies of the files (which includes new file folders), to reuse the same file folders attached to the original business object, not to copy any files, or to allow the user to choose one of these methods at the time he performs the Save As operation.

Select the setting that best suites your company's business practices.

| | |
|---|---|
| **Note** | This preference does not apply to File Folder objects themselves, except as outlined in the table below. |

For Product Portfolio Management objects, this preference applies only to attachments in the current object; it does not propagate down to all children tasks. The preference does apply when creating a new project from a PPM template.

The following table illustrates how each setting works in the case of the following example:

- Object ABC includes the following attachment:

  **File Name** = Test.txt, **File Folde**r = Folder004, **Folder Version** = 3.

- A user opens Object ABC and uses Save As to create Object XYZ.

| Setting | Example of how it works |
|---|---|
| **Create new copy of file(s)** | When Object XYZ is created, a new copy of the Test.txt is attached in a new file folder. Object XYZ's **Attachments** tab lists:<br>▫ **File Name** = Test.txt<br>▫ **File Folde**r = Folder013<br>▫ **Version** = 1<br>A new copy of Test.txt is created in the file vault, and a new file folder (Folder013, Version 1) is created.<br>The end result (new copy of the file, new file folder, at Version 1) is the same as if the user had added the attachment Test.txt by using the **Add \| Files** button on the **Attachments** tab of Object XYZ. |
| **Reference existing file(s)** | When Object XYZ is created, it reuses the same attachments as the original Object ABC. Object XYZ's **Attachments** tab is the same as Object ABC's **Attachments** tab, and lists:<br>▫ **File Name** = Test.txt<br>▫ **File Folde**r = Folder004<br>▫ **Version** = 3<br>The end result (the file folder attachments are reused) is the same as if the user had added the attachment Test.txt by using the **Add \| By Search** button on the **Attachments** tab of Object XYZ, searching for Object ABC and selecting its attachment: Test.txt, Folder004, Version 3. |
| **Do not copy file(s)** | When Object XYZ is created, no attachments are copied from the original Object ABC.<br>The end result is that the newly created Object XYZ has no attachments; its **Attachments** tab is empty. |
| **Prompt** | The user is prompted to select one of the Save As Attachment methods:<br>▫ **Reference existing file(s)**<br>▫ **Create new copy of file(s)**<br>▫ **Do not copy file(s)**<br>Agile PLM always prompts each user whenever a Save As operation is performed. However, if the original Object ABC does not have any attachments (the Attachments table is empty), the prompt will not appear. |

## Allow Download of Productivity Components

This preference controls whether your users — specifically Web Client users — will have access to Java applets that permit advanced features. Besides specific applications to the PPM and PG&C

solutions, this preference enhances system behavior with regard to attachments in these three ways:

- The Advanced File Uploader is available to users in Web Client.
- Automated checkin is based on the checkout location in Web Client.
- Allow download of individual files instead of a single Zip file in Web Client.

**Note** Java Client users always have access to these advanced attachment features, regardless of the setting of this preference.

If this preference is set to Yes, the end user must also have the appropriate user profile setting for his **File Productivity Preference** attribute.

If you have set this preference to Yes, and a Web Client user does not have access to the advanced attachment features, check his user profile. See also File Productivity User Preference in the User Profile on page 391.

| Setting | Results |
|---|---|
| Yes | File Productivity Preference attribute appears in the user profile, **Preferences** tab. User may set this to Advanced, Standard, or Prompt. If set to Advanced or Prompt, the Web Client user has access to advanced attachment features. |
| No | File Productivity Preference attribute does **not** appear in the user profile, **Preferences** tab. The Web Client user does not have access to advanced attachment features. |

## File Productivity User Preference in the User Profile

**Note** **File Productivity Preference** is a user preference – strictly, simply an attribute in every User object – not a systemwide preference. It is discussed here because it works in conjunction with Allow Download of Productivity Components systemwide preference: the **File Productivity Preference** user preference will not appear in any user's Profile if the Allow Download of Productivity Components systemwide preference is set to No.

User properties are documented in User Properties Defined on page 170.

This user setting controls whether the user has access to advanced productivity components, or standard components, or wants to be prompted in each situation. The available settings are Prompt, Advanced, or Standard. The default is Prompt; you as administrator can set any of the values for any user; each user can change the setting in their User Profile (again, assuming the systemwide preference **Allow Download of Productivity Components** is set to Yes).

**Note** The "Working with Attachments" chapter of *Agile PLM Getting Started Guide* includes instructions for the end user for setting his user profile **File Productivity Preference** and explains the behavior of each setting.

If the user preference is set to Standard mode, non-supported files will not be opened automatically. Instead, the user is prompted to choose Save or Open.

**Note** Make sure you choose appropriate character sets where required. There is a potential problem when adding files or URLs from a business object to a file folder object. If the file folder's File Description attribute has a Character Set = Numeric, but the business object's File Description attribute has a Character Set = Alpha or AlphaNumeric, the latter will override the former, meaning whatever control you wanted to impose in the file folder's File Description (being numeric) will be lost, the system will save the original character set.

# Assign Roles and Privileges for Attachments

Agile PLM roles are fully documented in Roles on page 187, and privileges are fully documented in Privileges and Privilege Masks on page 207. You should evaluate your objectives carefully before creating new roles and privileges. While customized roles and privileges may need to be created, it is always preferable if your objectives can be handled by out-of-box roles and privilege masks.

The following section provides information about privileges that are constructed to realize attachment capabilities.

## Attachment Privileges

Some privileges allow users to perform actions on attachments; we will call them "attachment privileges" in this discussion. Users who are assigned Folder Administrator or Folder Manager roles may need to review the information below.

The following privileges allow users to perform actions on attachments:

- Checkin
- Checkout
- Cancel Checkout
- GetFile
- PrintFile
- ViewFile
- Modify

**Important** GetFile privilege works in tandem with Checkout privilege to actually deliver the attachment file to the user's machine. A user with Checkout but not GetFile will see in History that the file is checked out, but it has not been downloaded to the user. The GetFile privilege (without Checkout) allows the administrator to permit a user to get a file – open it, read it – without being able to change it in the product record.

You can create privilege masks and roles that allow you to provide users all attachment actions from the business object **Attachments** tab; direct access to the file folder object is not required. Thus, file folder objects are "invisible" to the user, even though the user has the capability to perform all attachment actions from the business object **Attachments** tab.

**Note** Attachment actions performed from the business object's **Attachments** tab are carried out on the referenced file folder object. For example, checking out an attached file on the **Attachments** tab actually checks out the referenced file folder. However, if the user has the correct attachment privileges, he does not also require file folder privileges.

However, if required, attachment privileges can apply directly to file folder objects, and privileges for file folder objects should be reserved for those users who will actively maintain those objects. These users can search for file folder objects, open them, check files in and out on the **Files** tab, get files, print files, and view files. But you don't want every user to have this range of permissions on the file folders, and those roles don't give users the correct functionality anyway.

What will be more useful for standard users is having attachment privileges for specific classes or subclasses of business objects (or, as needed, specific objects). A user who has attachment privileges for manufacturer parts can perform attachment actions on any manufacturer part's **Attachments** tab, provided that the Folder Version selected in the attachments table is the latest version. If a user's attachment privileges pertain to a subclass of manufacturer parts, say, Resistors, the user can perform attachment actions on any resistor's **Attachments** tab, again provided that the Folder Version selected in the attachments table is the latest version.

As long as the latest version is selected, a user—from the **Attachments** tab of the current object—can check out and check in and perform the other attachment actions.

Therefore, a user with attachment privileges for *only* manufacturer parts can perform attachment actions from a manufacturer part's **Attachments** tab but *not* from an item's **Attachments** tab.

To summarize, users do not need privileges for file folder objects to work with attachments; however, they do need privileges for the business objects they will be working with (with the privilege specifying a class, subclass, or specific object). Users can create, check in, and check out file folder objects, but only in association with the Agile PLM class, subclass, or object for which they have business object attachment capability privileges.

## Attachment Capability Privileges in a Role

Some out-of-box roles have all of the above-named attachment privileges for the object types that role would usually work with, that is, the object types that the role is designed to create and modify. They may also have file capabilities (get, view, and print) for other related object types.

For example, the Item Content Manager role includes all seven attachment privileges for items and changes. It also includes GetFile, PrintFile, and ViewFile privileges (not Checkout or Checkin) for manufacturers, manufacturer parts, and price objects (that is, objects that are listed on the **Manufacturers** and **Prices** tabs of an item).

Following the Item Content Manager role as an example, you can create roles that allow users to get, print, and view attachments for all objects related to that user's Agile PLM system activities, and limit the checkin/checkout privileges to only those objects that the user can create and modify. A role that includes privileges for file folder objects does not allow you to control the user's attachment capabilities to the same level of granularity.

## Modify Privilege and Attachments

The Modify privilege allows a user to modify or edit fields on the tab of an object. Modify privileges are assigned using the AppliedTo attribute settings.

Agile PLM uses Modify privilege mask Applied To properties to determine and control certain specific user actions and attachment capabilities. For example, if a user has a role that includes a Modify privilege mask for parts and the Applied To property includes the **Parts.Attachment.FileDescription** attribute, then that user is able to modify the File Description field on the **Attachments** tab of a part object. This same modify action will also modify the File Description field on the **Files** tab of the referenced file folder object, however, a modify privilege mask for the file folder object is *not* required.

The following table lists the Modify privilege mask Applied To properties and which attachment actions they control.

**Note** This table uses File Folder or Business Object as a placeholder for the class or subclass name in the Applied To attribute column. The actual class or subclass name in any privilege mask is determined by the reusable criteria specified in the privilege mask. (See Privilege Mask Components on page 221.) If you view the example privilege masks mentioned in the Notes column, the Applied To properties will be appropriate for that specific privilege mask (for example, Manufacturer parts.Attachments.FileName).

The Notes column lists at least one example privilege mask that demonstrates a privilege mask that provides the user action; it is not a comprehensive list of example privilege masks. In cases where a Modify privilege mask is not required, only the needed privilege mask is listed (such as Example - Checkin for File Folders). If multiple privilege masks (in addition to a modify privilege mask) are needed to enable the user action, they are all listed.

| User action | Privilege | Class | Applied to | Example Privilege Mask or Role / Notes |
|---|---|---|---|---|
| **File Folder attachment actions** | | | | |
| Add file to File Folder | Modify | File Folder | Files.Filename | Example - Modify File Folders |
| Checkin File from File Folder | Checkin | File Folder | | Example - Checkin for all File Folders (criteria = Example - All File Folders)<br><br>Example - Checkin for File Folders (criteria = Example - All File Folders Checked Out By Me) |
| Delete File from File Folder | Modify | File Folder | Files.Filename | Example - Modify File Folders |
| Example file folder roles:<br>Example - File Folder Manager allows user to work with his own file folder objects.<br>Example - File Folder Administrator allows user to work with any file folder objects. | | | | |
| **Business Object attachment actions** | | | | |

| User action | Privilege | Class | Applied to | Example Privilege Mask or Role / Notes |
|---|---|---|---|---|
| Add new attachment to Business Object | Modify | Business Object | Attachment.Filename | Example - Modify Mfr Parts |
| Add attachment by search to Business Object | Discover | Business Object (the business object being searched against) | | As an example, if the user is searching for item objects from which to add attachments to a Mfr Part, then he requires:<br>Example - Discover Items |
| | Modify | Business Object | Attachment.Filename | Example - Modify Mfr Parts |
| | | | | Example - Component Engineer role includes both these example privilege masks and would allow a user to add an attachment to a Manufacturer Part by using **Add \| Search** to search for Items. |
| Add multiple attachments in one file folder to business object and specify Folder Description | Modify | Business Object | Attachment.Filename | Example - Modify Mfr Parts |
| | Modify | Business Object | Attachment.Folder Description | Example - Modify Mfr Parts |
| Checkin attachment on Business Object | Checkin | Business Object | | Example - Checkin for Mfr Parts |
| Remove attachment from Business Object | Modify | Business Object | Attachment.Filename | Example - Modify Mfr Parts |
| Edit File Description on Business Object **Attachments** tab | Modify | Business Object | Attachment.File Description | Example - Modify Mfr Parts |
| Edit Folder Description on Business Object **Attachments** tab | Modify | Business Object | Attachment.Folder Description | Example - Modify Mfr Parts |
| Show Version on Business Object **Attachments** tab | Read | Business Object | Attachment.Filename | Example - Read Mfr Parts |

# Markup Subclass

| | |
|---|---|
| **Note** | Attachment redline markups and file folder Ad Hoc redline markups are created, modified, and viewed in the AutoVue for Agile. |

The Markup subclass of the File Folders class is used to save attachment-file redline markups. By assigning the appropriate privilege masks to different kinds of users, specific users (for example, vendors) can see and modify only their own markups, but cannot see markups created by other users.

For detailed information about Markup subclass privilege masks, see Redline Markup Default Roles and Privileges on page 396.

Markup subclass objects are visible in Agile PLM for any user that has Read privileges for Markup objects. For example, if a user searches for File Folders class objects, Markup objects are included in the search results. However, the redline markup file they contain can be viewed only when viewing the item attachment-file or the file-folder file with which the redlines are associated. Attempting to view the redline markup from the **Files** tab of the Markup subclass object displays an error message when the AutoVue for Agile opens.

See also Discovery and Read Privileges on page 236.

## Markup Promotion

Markup Promotion is a generic capability supported within AutoVue to “promote” (copy) markup files from one base file to another. The user can promote markups from previous revisions (for change-based markup) or previous versions (for ad-hoc markup), permitting the user to see if the changes made in the following revision or version address the comments originally made in the markup.

This works with both File Folders class (therefore, objects from File Folder and Markup subclasses) and the new Designs class (objects from Design subclass). It works with both types of markup:

- Change-based – from Changes **Redline Attachments** tab
- Ad-hoc – from File Folders or Designs **Files** tab

Markups are selectable for promotion if they are on a previous revision or version and the base file extension is the same, for example, if Version 1 is **file_v1.pdf** and Version 2 is **file_v2.pdf**.

# Redline Markup Default Roles and Privileges

Agile PLM uses the file folder subclass Markup to store attachment redline markups created in the AutoVue for Agile. Agile PLM provides a default set of Markup subclass privilege masks and two default roles, **Markup for Self** and **Markup for All**. If you have upgraded PLM, you must create these roles by referring to the example roles **Example - Markup for Self** and **Example - Markup for Others**. See also Markup Subclass on page 396.

**Note** The **Markup for Self** and **Markup for All** default roles replace the Attachment Redlines for Self (ARS) and Attachment Redlines for Others (ARO) privileges. The privileges ARS and ARO are disabled; ARS and ARO privilege masks have no effect. However, any previously assigned custom ARS or ARO privilege masks are not deleted or removed; so, you can refer to the ARS and ARO privilege masks when creating and assigning redline capabilities with Markup subclass privilege masks or roles.

A user with Markup capabilities can create and modify the following types of redline markups:

- Create and modify attachment redline markups for the affected items of any change order object for which he has Read privileges.
- Create and modify Ad Hoc file folder redline markups for any file folder object for which he has Read privileges.

The **Markup for Self** role allows the user to redline attachments from the **Affected Items** tab of a change order and to Ad Hoc redline files on the **Files** tab of a file folder, and he can modify or delete only those redlines he created.The privilege masks in the **Markup for Self** role are based on the Markup for Self criteria: Markup Page Two.Create User Equal to $USER.

The **Markup for Others** role allows the user to redline attachments from the **Affected Items** tab of a change order and to Ad Hoc redline files on the **Files** tab of a file folder, and he can modify or delete only those redlines created by other users. The privilege masks for the **Markup for Others** role are based on the All Markups criteria.

The following table lists the default privilege masks in each of the default roles for Markup subclass:

| Markup for Self role | Markup for Others role |
|---|---|
| Read My Markups | Read All Markups |
| Modify My Markups | Modify All Markups |
| Checkout My Markups | Checkout All Markups |
| Checkin My Markups | Checkin All Markups |
| Create Markups | Create Markups |

## Redline Markup Privilege Guidelines

The following rules and guidelines apply to redline markup capabilities:

- If a user does not have Read privilege for a specific Markup object, the associated redline markup is not listed in the markup panel in AutoVue for Agile.
- If a user does not have Modify privilege for a specific Markup object listed in the AutoVue for Agile markup panel, the AutoVue for Agile markup panel is disabled and the user cannot edit any redline markup for that viewed file.
- If a user does not have a Create Markup object privilege mask, he cannot create new markups from the AutoVue for Agile.
- In **Admin > Settings > Data Settings > Classes**, for **File folders** class, the Page Two attribute **Create**

**User** must be enabled. The default value is $USER. This attribute is used to determine "ownership" of each redline markup.

- In order to differentiate between privilege masks for Markup subclasses and all other file folder subclasses, Create, Read, Modify, Checkout, and Checkin privilege masks for file folders must use criteria that specify specific file folder subclasses (for example, Markup subclass and File Folder subclass). This allows you to use a set of privilege masks specific to Markup subclass. The Markup-specific privilege masks are assigned only to those users to whom you wish to grant redline capabilities.

  In contrast, privilege masks based on criteria that specify the File Folders base class or the File folders class will apply to all file folder subclasses, thus depriving you of the ability to restrict redline capabilities (by *not* assigning to users Markup subclass privilege masks). That is, privilege masks based on criteria for the File Folders base class or the File folders class will always apply to Markup subclass objects.

- The default Markup subclass privilege masks are based on broadly-defined reusable criteria (**All Markups** and **Markups for Self**). However, you are not limited to these criteria when creating Markup subclass privilege masks. For example, you can enable a list attribute or a multilist attribute, then create criteria that will determine Markup capabilities based on the settings of that list attribute or multilist attribute.

# Document Management Enhancements

A few recent enhancements concerning attachments are discussed below.

## View All Versions

On the standard **Actions** menu, the most significant added Action is **View All Versions**. The View All Versions action is available in File Folder and Design objects.

The **View All Versions** utility is extremely useful as file-folder structures or design structures grow, as it tracks the versions of each design or file-folder. Each version displays:

- Approved (users who have approved file folder or design in workflow), including the Awaiting Approval icon (users who have not yet approved);
- Checkin Date;
- Checkin User (which captures the user who checked in the version, and also can read through to the design **Structure** tab and to **Attachments** tabs in designs or file folders);
- Change Info and Label (as filled in by users); and,
- Revision and Revision Date.

Many of these fields are pulled from the object's **Title Block** tab.

Using the **View** dropdown, you can have a quick **View** of a selected version of the design, or select two versions and **Compare** them visually.

## Limit Attachment Size

Administrators can set the maximum file size that can be uploaded at **System Settings > Viewer and Files > Upload File Size Limit (in MB)**. This setting defines the maximum size (in MB) of uploaded files

for Java Client, Web Client, Portal Client, and SDK.

The default value, 0, indicates "No Limit"; any non-negative integer sets the size limit in MB. Users will see an error message in Java and Web clients if a file is too large. Files that successfully upload (before a failure occurs) remain uploaded, other files (after a failure occurs) will not be uploaded.

## Identify Checkin User

The Checkin User field is available on File Folders and Designs **Title Block** tab and on any business object's **Attachments** tab. It identifies the user who checked in the specified version.

This field is not available on Project Content Tab.

## URLs can be up to 255 characters

Attached URLs can be up to 255 characters in length.

# Generating Attachment Quick Access URLs

Attachment URLs allow you to do the following:

- Generate a URL for an attachment on item objects.
- Generate a URL for a file on file folder objects.
- Generate a URL for a file on the **Content** tab of Project objects.
- Paste URLs from Agile to another application
- Externally open files outside of the Agile application.

You can generate attachment URLs from Items, Activities, and File Folders by two methods:

- Using Agile PLM commands to copy a URL to your clipboard:

  Agile-Generated Attachment Quick Access URLs on page 399
- Manually typing an Attachment Smart URL in an easy-to-understand format:

  User-Generated Attachment Quick Access Smart URL Formats on page 400

The URL format enables you to identify which object and files are being referenced. Because of the easy-to-understand format, you can generate your own URLs to point to specific file attachments without logging into the Agile system, if you know where the files are located.

## Agile-Generated Attachment Quick Access URLs

From Web Client, you can only generate Attachment Quick Access URLs from the **Attachments** tab of any object, the **Content** tab of Activities, and **Files** tab File Folders by clicking **Get Shortcut** after performing one of the following scenarios:

- Selecting **only** one row from the **Attachments**, **Content**, or **Files** tab to generate a URL that points to the folder corresponding to the file on that row.
- Selecting all the rows on the **Attachments**, **Content**, or **Files** tab to generate a URL that points to

the collection of files on the tab.

**Note** You must have Read privileges for an object in order to generate a URL for the object.

**Caution** If you select more than one row on the **Attachments**, Content, or **Files** tab, the Get Shortcut link is disabled.

The URL is copied to your clip board. You can then paste it into another application or file, such as a word processing file, a spreadsheet file, or an email.

# User-Generated Attachment Quick Access Smart URL Formats

With the easy-to-understand Smart URL format, you can generate your own URLs to point to specific file attachments without logging into the Agile system, if you know where the files are located.

The formats of user-generated attachment Smart URLs depend on the object type, and fall into three main categories:

- Item Attachment Quick Access URL Formats (on page 400)
- File Folders Quick Access URL Formats (on page 402)
- Program Content Quick Access URL Formats (on page 403)

## Item Attachment Quick Access Smart URL Formats

You can generate Smart URLs for item attachments in the following format:

http://<server:port>/<VirtualPath>/link/<ObjectType>/<ObjectNumber>[/Revision/ChangeNumber]/files/Folder/<FolderNumber>[/<FileAction>]

or

http://<server:port>/<VirtualPath>/link/<ObjectType>/<ObjectNumber>[/Revision/ChangeNumber]/files/<FileName>[/<FileAction>]

where

| <Server:Port>/<Virtual Path> element | Identifies the Agile instance |
|---|---|
| link | The text "link" identifies the URL as a special attachment URL to the Agile server. |
| ObjectType | Refers to the subclass of the Item, File Folder, or Project/Activity. |
| ObjectNumber | Refers to the actual object number for the object type. |
| Revision | Identifies an item revision by specifying the change number corresponding to that revision. |
| ChangeNumber | Identifies a unique revision (Released or Pending) of the item. If the number is not specified, the latest released revision is attached. If the number is not specified and the item does not have a released |

| <Server:Port>/<Virtual Path> element | Identifies the Agile instance |
|---|---|
| | change, the Introductory revision is attached. |
| Folder | Indicates that the object is a file folder, not the actual file. |
| FolderNumber | Identifies the file folder related to the selected row. The version of the file retrieved is the most current version, based on the specific revision of the item. |
| FileName | Refers to the name of the specific file attached to the specified revision of the item. |
| FileAction | Determines the action performed with the retrieved files. Values are GET or VIEW. If FileAction is not specified, a GET is performed.<br>If the parameters specified in the URL point to a single file, the file is retrieved in its native format. If the parameters specified in the URL point to a collection of files, then all the files are returned as a zipped file called "Download.zip". |

If the specific revision of an item has multiple files contained in the same folder on the Attachments tab, but is referenced from different versions of the folder, then a Folder-based URL may not reflect the actual set of attachments from the item revision's Attachments tab.

Examples of user-generated URLs for item attachments:

- http://myappserver/mycompany/Link/Document/DOC0001/files/Folder/Folder001

  Returns a zipped file containing all files from Folder001 that are attached to the latest released revisions of DOC0001.

- http://myappserver/mycompany/Link/Document/DOC0001/files/Folder/Folder001/VIEW

  Opens all files from FOLDER001 that are attached to the latest released revision of DOC0001 in the Viewer.

### *Limitations for Item Object Attachment Quick Access Smart URLs*

The following limitations apply to retrieving the attached files of item objects:

- In the case where a specific revision of an item has multiple files on its **Attachments** tab that belong to the same folder, but are referenced from different versions of the folder, a folder-based URL may not reflect the true set of attachments from that item+revision's **Attachments** tab. For example, consider an item P0001, whose revision A has two attachments: P0001_3dmodel.dwg and P0001_specs.pdf, both in Folder1. Assume that the DWG file is from version 2 of the file folder, and the PDF file is from version 5 of the file folder. In this case, the following URL:

  http://server:port/<VirtualPath>/link/Part/P0001/Revision/C0001/files/Folder/Folder1

  returns all the files contained in either version 5 or version 2 (results are not predictable) of Folder1.

  However, the following URL:

  http://server:port/<VirtualPath>/link/Part/P0001/Revision/C0001/files/ALL

  returns the complete set of files belonging to revision A of the item, including

P0001_3dmodel.dwg from version 2 of Folder1 and P0001_specs.pdf from version 5 of Folder1.

- You cannot generate an item-based URL that will return a file from a historical version of a file folder referenced on an item's attachments tab. The user can, however, generate a folder-based URL to achieve this.
- In case of the following format example,

  http://server:port/<VirtualPath>/link/ObjectType/ObjectNumber[/Rev/ChangeNumber]/files/FileName[/FileAction]

  If two files of the same name (filename + file extension) exist on the attachments table, and a user generate a URL using filename, only the first encountered file is retrieved. If the "ALL" option is used to access files from the **Attachments** tab of an item where there are multiple files by the same name on the attachments table, or if a URL is generated to return all the files of a file folder where there are more than one file with the same name, an error is displayed indicating that there are multiple files by the same name and therefore a valid zip archive of files cannot be generated.
- If the ALL option is used to access files from a Project's **Content** tab where multiple files with the same name exist, an error is displayed indicating that there are multiple files by the same name and therefore a valid zip archive of files cannot be generated.

## File Folders Quick Access Smart URL Formats

You can generate Smart URLs for file folders in the following format:

http://<server:port>/<VirtualPath>/link/<ObjectType>/<ObjectNumber>/files/<FileName>[/</version][/<FileAction>]

where

| <Server:Port>/<Virtual Path> element | Identifies the Agile instance |
|---|---|
| link | The text "link" identifies the URL as a special attachment URL to the Agile server. |
| version | Identifies the specific version of the file folder. The value can be a number, or the string, LATEST (default). |
| FileName | Refers to the name of the specific file from the folder or string ALL which attaches all the files in the specified version of the folder.<br>If more than one file of the same name exists in a specific version of a folder and a URL is generated based on filename, only the first version of the file is attached. You cannot use the ALL value to access files from a folder where more than one file of the same name exists. |
| FileAction | Determines the action performed with the retrieved files. Values are GET or VIEW. If fileaction is not specified, a GET is performed.<br>If the parameters specified in the URL point to a single file, the file is retrieved in its native format. If the parameters specified in the URL point to a collection of files, then all the files are returned as a zipped file called "Download.zip". |

Examples of user-generated URLs for file folder attachments:

- http://myappserver/mycompany/Link/FileFolder/Folder0001/files/abc.txt/1

  Gets the abc.txt file from version 1 of Folder0001.

- http://myappserver/mycompany/Link/FileFolder/Folder0001/files/ALL/VIEW

  Opens all files from the latest version of FOLDER001 in the Viewer.

### Program Content Quick Access Smart URL Formats

You can generate Smart URLs for files on the **Content** tab of Projects and Activities. The **Content** tab allows specification of any object as valid content and allows the addition of external files, just like the **Attachments** tab. However, the **Content** tab stores the folder object related to an attached file as valid content. So, when retrieving, you must attach the entire folder.

The Smart URL for files on the **Content** tab is generated in the following format:

```
http://<server:port>/<VirtualPath>/link/<ObjectType>/<ObjectNumber>/content/Folder/<FolderNumber>[/<FileAction>]
```

where

| | |
|---|---|
| ObjectType | Represents the subclass of the Project object. |
| ObjectNumber | Represents the identifier of the Project object. |
| Content | Indicates the **Content** tab. |
| Folder | Indicates that a specific folder is being retrieved. |
| FolderNumber | Refers to the number of the folder being retrieved from the **Content** tab. The most current version of the folder is always retrieved. |
| FileAction | Determines the action performed with the retrieved files. Values are GET or VIEW. If fileaction is not specified, a GET is performed.<br>If the parameters specified in the URL point to a single file, the file is retrieved in its native format. If the parameters specified in the URL point to a collection of files, then all the files are returned as a zipped file called "Download.zip". |

## Accessing Attachment Files using Quick Access URLs

After clicking or opening an Attachment Quick Access URL, a login popup appears and you must log into the Agile system to view the attachment. Agile PLM then validates that you have the appropriate privileges to get or view the file or set of files specified in the URL. The attachments are viewable in all supported browsers, the AutoVue for Agile, or the native application.

**For example:**

1. In an email, the user clicks the Attachment Quick Access URL and an Agile login popup is displayed.
2. The user enters her login information.

   Agile PLM verifies that the user has the appropriate privileges to read and view the file.

**Note** If the URL returns a collection of files and the user does not have the appropriate Get or View privilege for one or more files, those files are excluded from the returned collection. However, a collection of files is returned to the user containing the rest of the files for which he does have the appropriate Get or View privileges.

# Chapter 36

# Administering Designs

**This chapter includes the following:**


- Evaluating Your Company's Needs with Designs ..... 405
- Purpose of Design Data Model ..... 405
- Designs Class Overview ..... 406
- Designs Class Tabs ..... 407
- Administrator Settings Related to Designs ..... 409


## Evaluating Your Company's Needs with Designs

As with other areas of Agile PLM administration, the most important step comes before setting properties in Administrator. This chapter assumes that you answered the first question in the previous chapter affirmatively.

- Does your company produce CAD design and graphic files? If so, you will build from the Designs class.

The Designs class is a business class in the File Folders base class.

Objects created in the Designs class have many of the same properties and behaviors of File Folders. The Designs class is enhanced for use with the Engineering Collaboration (EC) solution; therefore, this chapter highlights those differences. The chapter on Administering Attachments on page 387 contains foundation details about the File Folders base class.

**Note** Deployment of Design structures in Agile PLM requires an Agile EC integration. That is, full use of CAD design graphics files needs an integration application that can automatically check out and check in Designs objects, track and update parent-to-child versioning of design structures, and so forth. Contact your Oracle Consulting–Agile Practice representative for additional information about Agile EC.

## Purpose of Design Data Model

The Designs class supports structures between Design objects with explicit versions – let's call these "design structures" – and it supports integration with CAD tools. The Design object is very similar to File Folders, but has a few "extras" and specific features to realize its intended purpose.

The Designs class overcomes limitations that have existed in PLM using Documents class and the "DocuBOM" data model, which were not expressly designed for CAD use-cases. Three primary issues are resolved with design structures:

- **Structure Resolution is easily captured**
  - "Structure" information between designs can be captured, as required by CAD environments, while the "DocuBOM" structure does not directly support fixed structure

resolution.

- **Data Navigation is greatly enhanced**
  - Structures of Designs objects are evident and easy to navigate, while the combination of Part-and-Document BOMs are difficult to navigate in the Agile user interface.
- **Change Control Process carries much less "overhead"**
  - The full change control process (in the Product Collaboration solution) has too much overhead for many CAD use cases.

Although the Designs class is enabled in new installations, it is disabled upon PLM upgrade: in upgrading installations, the Agile administrator must enable the Designs class.

Benefits of Designing Structures using the Designs Class

Besides the benefits to CAD environments mentioned above, additional benefits are:

- Easy recognition of CAD data in PLM; CAD data will "look" more like it does in CAD
- Users can "double-use" part names since Designs and Parts are separate classes of objects
- Searching for design objects will be easier for CAD engineers
- Using a single object has several advantages:
  - Change history is on the Design object instead of on a separate Change object
  - Design files are not in a separate "file folder"
  - Version-specific Where Used
  - Version-specific Routing Slip
- Designs class enables the next version of Agile Engineering Collaboration, which will capture and exploit the capabilities built into the PLM model structures.

# Designs Class Overview

The Design object can be opened from Java Client, but it opens Web Client and the user works with Designs exclusively in Web Client.

There is a **Checkout** button, and modifications to a design structure cannot be done by a user without checking out the design.

Since a "parent" version of a design explicitly points at a "child" version, a user has to check out a version, update the children, and check it back in for the parent to reflect the changes. In CAD there can be multiple models for 1 part – there could be, for example, 20 or 30 versions of a single object – but in Product Collaboration the part is assigned to a single structure; moreover, the system lets you check in a "design structure" *without* checking in all child parts.

Now, those engineers who have “child" parts checked out must check that part back in, that is, he cannot "Cancel Checkout" for the part to “rejoin” the BOM. Attempting to do so will throw the error message: “This version is in use and cannot be canceled.”

A feature of the Version dropdown list is that the checked-out version is incremented and displayed in brackets, for example **[4]**, so the user can always see what the new version number will be when

the object is checked back in. This applies to both File Folders and Designs classes.

## View All Versions

On the standard **Actions** menu, the most significant added Action is **View All Versions**. The View All Versions action is available in File Folder and Design objects.

The **View All Versions** utility is extremely useful as file-folder structures or design structures grow, as it tracks the versions of each design or file-folder. Each version displays:

- Approved (users who have approved file folder or design in workflow), including the Awaiting Approval icon (users who have not yet approved);
- Checkin Date;
- Checkin User (which captures the user who checked in the version, and also can read through to the design **Structure** tab and to **Attachments** tabs in designs or file folders);
- Change Info and Label (as filled in by users); and,
- Revision and Revision Date.

Many of these fields are pulled from the object's **Title Block** tab.

Using the **View** dropdown, you can have a quick **View** of a selected version of the design, or select two versions and **Compare** them visually.

# Designs Class Tabs

The tabs on Designs objects are: **Title Block**, **Files**, **Structure**, **Routing Slip**, **Relationships**, **Where Used**, and **History**. Most of these are given a look in the sections below; **Relationships** and **History** tabs are documented in *Getting Started with Agile PLM*.

## Title Block Tab

By default, the Design object **Title Block** tab contains the fields listed in the following table. The administrator determines which fields are enabled and visible on the **Title Block** tab.

In Web Client, the **Title Block** can contain two additional sections, called **PageTwo** and **PageThree** by default. Agile administrators can add custom class fields to **PageTwo** section and custom subclass fields to the **PageThree** section. The Agile administrator determines whether these sections are enabled, and what they are called.

| Field | Description |
|---|---|
| Number | The Design object's number, completed when the file object is created. |
| Type | The subclass of this Design object. Depending on Agile system settings, this field is automatically completed when the Design object is created. |
| Lifecycle Phase | Current lifecycle; selected from a list. |
| Description | Text describing the Design object |
| Version | The currently selected version of the Design object |

| Field | Description |
|---|---|
| Last Modified Date | Date the Design object was last modified |
| Checkout Status | Indicates whether the Design object is currently checked out or checked in. |
| Checkout User | When the Design object is checked out, displays the user who checked out the Design object |
| Checkout Date | When the Design object is checked out, displays the checkout date |
| Checkout Location | When the checkout user's File Productivity Preference is set to Advanced, the checkout location is automatically filled in |
| Checkin Date | When the Design object is checked in, displays the checkin date |
| Create Date | The date the Design object was created |
| Label | Version-specific text field that holds a specified label |
| Component Type | Displays the type of component represented by the Design |
| Thumbnail | Displays information about thumbnail graphics in Design objects; whether it is enabled or not is determined by the systemwide Preference Thumbnail Support and the User Preference settings. |
| Revision | Version-specific text field that holds the targeted revision of the Design version |
| Revision Date | Version-specific date field that holds the revision date associated with the Design version |
| Checkin User | Version-specific field indicating the user who checked in the currently selected version; completed automatically |

# Files Tab

The actions available on Designs **Files** tab are generally the same as File Folders **Files** tab. Similarly, privileges around the **Files** tab of both kinds of objects will mostly be the same; however, for best control of user access, privileges should not be based on the File Folders *base class*, which will apply to the **Files** tab of both File Folders and Designs *class* objects.

In order for the end-user to execute **Files** tab actions for Design model objects, he must have the appropriate Modify privilege masks. Refer to Attachment Privileges on page 392, especially Modify Privilege and Attachments on page 393: see the section of the table for File Folder Attachment Actions.

**Important** If you are using privilege masks based on the File Folders *base class*, assigned users will have the *same* Modify, Checkin, Checkout (and so forth) capabilities for both File Folders class objects and Designs class objects. If you need to give users separate types of access to these two classes, you must create and define privilege masks based on the File Folders class (or subclasses) or the Designs class (or subclasses).

## Structure Tab

The Designs **Structure** tab supports "read-through" fields from Designs **Title Block** tab or **PageTwo**.

Like the **Files** tab, the **Structure** tab can only be edited when the object is checked out.

In Structure, the parent explicitly specifies the version of the child.

The Multi-level dropdown allows you to **Expand** or **Collapse** the structure.

AutoVue for Agile supports Designs **Structure** tab as well as File Folder **Attachments** tab.

| Note | File Folders (objects from File Folder and Markup subclasses) do not have a Structure tab, and they cannot be in a model structure. On the other hand, where you can use an “attachment” (that is, a file folder object), you can use a Design. |
|---|---|

Icons indicate whether a Design is:

- Checked out
- Attached
- Latest Version
- Has Attachments

## Routing Slip Tab

The Routing Slip tab is functionally "version-specific". Signoff on the Design object only signs off that specific version of the object.

Adding and removing Approvers and Observers can be done only when Design objects are checked in. In order to add or remove approvers and perform signoff actions, the user needs the appropriate privilege masks for Design objects.

## Where Used Tab

Files that are associated with this design are listed on two tables of the **Where Used** tab, the **Attachments** table and **Design** table.

The **Where Used** tables inherit **Structure** tab columns and read-through fields.

# Administrator Settings Related to Designs

This section collects various Administrator settings for convenience. Refer to node-specific chapters for more information about the capabilities in Administrator.

## Display Structure Tables SmartRule

The SmartRule “Display Structure Tables” offers the choice between No Display (appropriate for non-CAD business environments) and Display.

# Design-specific Role & Privilege

Design Engineer is a Role tailored to create, access, and manage Designs class objects.

Privileges and privilege masks around the Designs business object use the Agile PLM Discovery and Read model, including on the **Structure** tab.

In order to add or remove approvers and perform signoff actions, the user needs the appropriate privilege masks for Design objects.

**Note** When working in conjunction with a CAD-to-Agile PLM integration application, each CAD user must also be an Agile PLM user. Each CAD user must have appropriate PLM roles and privileges to allow him to perform Checkout, Modify, Checkin (and so forth) actions on Design model objects. Contact your Oracle Consulting–Agile Practice representative for additional information about CAD-to-Agile PLM integration applications.

**Important** Remember that privilege masks written at the File Folder *base class* level will affect both the File Folders class (File Folder and Markup subclass) and the Designs class (Design subclass). Privilege masks tailored for CAD environments should be entered at Designs class level.

# Chapter 37

# Administering Agile Drive

**This chapter includes the following:**


- Overview ..... 411


# Overview

Agile Drive is a system extension enabling users to perform file management operations of Agile objects without requiring the user to invoke the Agile PLM application. It connects Agile's Distributed File Manager as a Network Place on your system, thus enabling Explorer-like operations, such as copy or delete, on file attachments of Agile objects. Agile Drive displays Agile objects as folders, and its attachments as files in these folders.

## File Management

With Agile Drive, you can work on files such as documents, drawings, etc., associated with Agile objects as file attachments in Agile. The actions a user can perform on files are the following:

- View / Read a file
- Save a file
- Update / Edit a file
- Delete a file
- View file properties
- Add a file
- Move a file
- Copy files

The action a user can perform on folders is the following:

- View Folder Properties

## Enabling Agile Drive

**To enable Agile Drive:**

1. Go to **System Settings** > **Preferences**.
2. From the **Agile Drive** dropdown list, select Enabled.
3. Restart all Distributed File Managers.

**Note** By default, Agile Drive is set to Disabled.

Once you enable the Agile Drive, the access will be Read-only.

For more detailed information about how to configure the required settings, set up Agile Drive for users, and perform actions using Agile Drive, see *Agile Drive User Guide*.

# Appendix A

# Configuring Single Sign-On

**This Appendix includes the following:**


- Overview of Single Sign-On in PLM ..... 413
- Configuring and Deploying SSO with Windows NTLM ..... 414


Agile PLM has the possibility of integrating aspects of your PLM system with Single Sign-On (SSO) capability.

## Overview of Single Sign-On in PLM

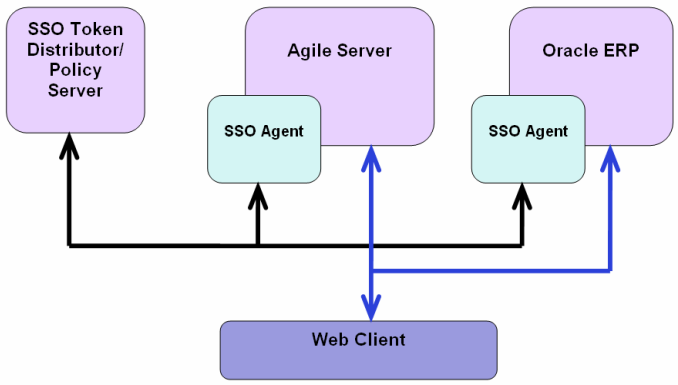
With SSO configured and enabled for your PLM system, a user that has signed in to the system once (for instance, through the corporate portal) is not prompted again by a "login" dialog in such cases as:

- Launching Web Client
- Clicking on a URL for an email notification
- When a customer's supplier launches the Microsoft Excel-based Solution from a Declaration
- When Web Client times out.

**Note** SSO is a Web-based solution that can be enabled only for Agile Web Client. SSO cannot be configured or applied from Java Client, Administrator module in Java Client, or the Agile SDK.

Single Sign-on integrates with the centralized security management, other business and training applications, and improves user productivity in Agile Web Client environment.

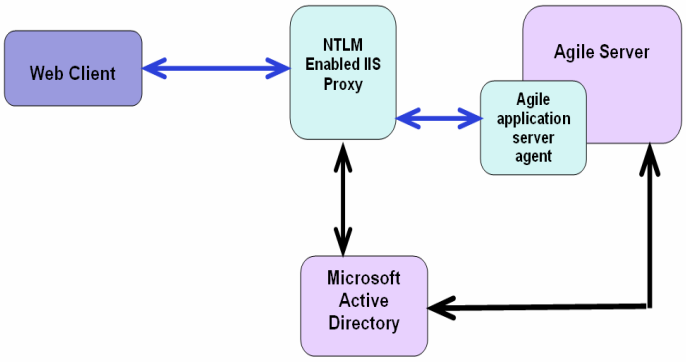
The sections below give a general overview to Single Sign-On in Agile PLM, followed by steps to configure and deploy SSO with Windows NTLM.



# Configuring and Deploying SSO with Windows NTLM

The following sections outline the steps to configure and deploy NTLM for Single sign-on capability.

## SSO NTLM Authentication




## Configure SSO for Windows NTLM

Microsoft Windows NTLM has been certified for Agile PLM. Please discuss your company's needs with your Oracle Consulting – Agile Practice representative.

### Configure IIS Proxy

These are the steps to configure Windows NTLM for Single sign-on with Internet Information Services (IIS) as proxy server.

| **Note** | The prerequisite for this configuration is that the Windows server on which IIS is running has been joined to a Windows domain. "Microsoft Active Directory" is the name of the Windows Domain controller since Windows 2000. |
|---|---|

1. In Windows 2003, choose **Start > Settings > Administrative Tools > IIS Manager**.
2. Select local computer, right-click and go to **Properties**.
3. Select **Directory Security** tab.
4. Click **Edit**.
5. Uncheck **Enable anonymous access**, check **Integrated Windows authentication**, and click **Apply**.
6. Restart IIS service.

## Configure PLM for NTLM

To configure your PLM system for SSO with NTLM, please perform these operations.

**A. Edit this file:**

**oracle_home/j2ee/home/config/jazn.xml.**

- Comment out the following elements:

```
<!--property name="idm.authentication.name" value="JavaSSO" />
<property name="idm.token.asserter.class"
value="oracle.security.jazn.sso.SSOCookieTokenAsserter" />
<property name="idm.token.collector.class"
value="oracle.security.jazn.sso.SSOCookieTokenCollector" />
<property name="idm.token.type" value="HTTP_COOKIE" />
<property name="idm.token.collector.cookie.1" value="ORA_OC4J_SSO"
/-->
```

- Add the following elements:

```
<property name="idm.authentication.name" value="AgilePLM-SSO" />
<property name="idm.token.asserter.class"
value="com.agile.admin.security.oracle.TokenAsserterImpl" />
<property name="idm.token.collector.class"
value="com.agile.admin.security.oracle.TokenCollectorImpl" />
<property name="idm.token.type" value="HTTP_REQUEST" />
<!-- SSO Type, can be one of following values: SECURE-PROXY,
SITEMINDER, NTLM /-->
<property name="agile.security.sso.interceptor.type" value="SECURE-
PROXY" />
```

**B. Edit this file:**

**oracle_home/j2ee/home/application-deployments/Agile/orion-application.xml.**

- Add this element <jazn-web-app auth-method="CUSTOM_AUTH" /> as seen below:

```
<jazn provider="XML">
<jazn-web-app auth-method="CUSTOM_AUTH" />
<property name="role.mapping.dynamic" value="true" />
<property name="custom.loginmodule.provider" value="true" />
</jazn>
```

**C. Edit this file:**

**<OraHome>/j2ee/home/applications/Agile/APP-INF/classes/agile.properties**

- Change the following element:

```
agile.sso.cookie.name = CMSession
```

to

```
agile.sso.cookie.name = oproxy_username
```

| **Note** | Setting this cookie property name enables users to have the correct behavior following an idle session timeout period in OAS. |
|---|---|

**D. Restart Agile.**

# Deploy NTLM

The following are possible SSO deployment scenarios with NTLM, one for secure proxy and one for transparent or no proxy.

## With Secure Proxy

With this deployment, authentication takes place on the proxy server, so it is recommended for those companies that use a proxy server.

**Request flow with this deployment:**

1. User launches browser to access Agile PLM (for example, http://agileplm.xyz.com/Agile/PLMServlet).
2. The NTLM-enabled IIS server challenges the browser for credentials.
3. After a successful NTLM handshake, the request reaches Agile Application Server (AAS) agent with user information.

   NTLM is a connection-based authentication protocol. For each new socket connection between client and server (or proxy), it has to exchange credentials by sending and responding to HTTP requests and responses.
4. The AAS agent passes the user information to the application server security framework.
5. The user will be allowed to access Agile applications.

   This authentication happens whenever the client sends an HTTP POST request; therefore, authentication can re-occur even during an established user session.

## With Transparent Proxy or No Proxy

**Request flow with this deployment:**

1. User launches browser to access Agile (for example, http://agileplm.xyz.com/Agile/PLMServlet).
2. The Agile Application Server (AAS) agent installed on server challenges the browser for credentials.
3. After a successful NTLM handshake, the AAS agent passes the user information to the application server security framework.
4. The user will be allowed to access Agile applications.