PowerCenter
10.4.1
Workflow Basics Guide
June 2020

Contents

Workflow Basics guide-copyright. 6
Preface. 7
  Informatica Resources. 7
Workflow Manager. 8
  Workflow Manager Overview. 8
  Workflow Manager Options. 11
  Navigating the Workspace. 15
  Working with Repository Objects. 17
  Checking In and Out Versioned Repository Objects. 18
  Searching for Versioned Objects. 20
  Copying Repository Objects. 20
  Comparing Repository Objects. 21
  Metadata Extensions. 22
  Expression Editor. 24
  Keyboard Shortcuts. 25
Workflows and Worklets. 26
  Workflows Overview. 26
  Creating a Workflow. 27
  Using the Workflow Wizard. 29
  Assigning an Integration Service. 30
  Workflow Reports (Deprecated). 31
  Working with Worklets. 31
  Workflow Links. 34
Sessions. 36
  Sessions Overview. 36
  Session Task. 36
  Editing a Session. 37
  Performance Details. 39
  Pre- and Post-Session Commands. 40
Session Configuration Object. 43

- Session Configuration Object Overview. . . . 43
- Advanced Settings. . . . 43
- Log Options Settings. . . . 45
- Error Handling Settings. . . . 47
- Partitioning Options Settings. . . . 48
- Session on Grid Settings. . . . 49
- Creating a Session Configuration Object. . . . 49
- Configuring a Session to Use a Session Configuration Object. . . . 50

Tasks. . . . 50
- Tasks Overview. . . . 50
- Creating a Task. . . . 51
- Configuring Tasks. . . . 52
- Working with the Assignment Task. . . . 54
- Command Task. . . . 54
- Control Task. . . . 56
- Working with the Event Task. . . . 59
- Timer Task. . . . 63

Sources. . . . 64
- Sources Overview. . . . 64
- Configuring Sources in a Session. . . . 65
- Working with Relational Sources. . . . 66
- Working with File Sources. . . . 68
- Integration Service Handling for File Sources. . . . 73
- Working with XML Sources. . . . 76
- Using a File List. . . . 77

Targets. . . . 79
- Targets Overview. . . . 79
- Configuring Targets in a Session. . . . 80
- Performing a Test Load. . . . 82
- Working with Relational Targets. . . . 82
- Working with Target Connection Groups. . . . 94
- Working with Active Sources. . . . 95
- Working with File Targets. . . . 96
- Integration Service Handling for File Targets. . . . 100
- Working with XML Targets in a Session. . . . 106
- Integration Service Handling for XML Targets. . . . 107

Working with Heterogeneous Targets. 113
Reject Files. 113
Connection Objects. 116
Connection Objects Overview 116
Connection Object Code Pages. 121
SSL Authentication Certificate Files. 121
Connection Object Permissions. 123
Environment SQL. 124
Connection Resilience. 126
Relational Database Connections. 127
FTP Connections. 130
External Loader Connections. 131
HTTP Connections. 132
PowerExchange for Amazon Redshift Connections. 134
PowerExchange for Amazon S3 Connections. 135
PowerChannel Relational Database Connections. 137
PowerExchange for Db2 Warehouse Connections. 138
PowerExchange for Essbase Connections. 139
PowerExchange for Greenplum Connections. 140
PowerExchange for Google Analytics Connections. 141
PowerExchange for Google BigQuery Connections. 141
PowerExchange for Google Cloud Spanner Connections. 142
PowerExchange for Google Cloud Storage Connections. 143
PowerExchange for Hadoop Connections. 143
HANA ODBC Connection Properties. 144
PowerExchange for JD Edwards EnterpriseOne Connections. 145
PowerExchange for JMS Connections. 146
PowerExchange for Kafka Connections. 147
PowerExchange for LDAP Connections. 149
Microsoft Azure Blob Storage Connection Properties. 149
PowerExchange for Microsoft Azure SQL Data Warehouse V3 Connections. 150
Microsoft Dynamics 365 for Sales Connection Properties. 151
PowerExchange for MongoDB JDBC Connections. 152
PowerExchange for MSMQ Connections. 153
PowerExchange for Netezza Connections. 153
PowerExchange for Oracle E-Business Suite Connection Properties. 154

PowerExchange for PeopleSoft Connections. 154
PowerExchange for PostgreSQL Connection Properties. 156
PowerExchange for Salesforce Analytics Connections. 157
PowerExchange for Salesforce Connections. 158
PowerExchange for SAP NetWeaver Connections. 159
PowerExchange for SAP NetWeaver BI Connections. 163
PowerExchange for Siebel Connections. 165
PowerExchange for Tableau Connections. 166
PowerExchange for Tableau V3 Connections. 167
PowerExchange for Teradata Parallel Transporter Connections. 168
PowerExchange for TIBCO Connections. 169
PowerExchange for Web Services Connections. 172
PowerExchange for webMethods Connections. 173
PowerExchange for WebSphere MQ Connections. 175
Connection Object Management. 177
Validation. 178
Workflow Validation. 178
Worklet Validation. 179
Task Validation. 180
Session Validation. 181
Expression Validation. 182
Scheduling and Running Workflows. 182
Workflow Schedulers 182
Workflow Scheduler Properties. 183
Scheduled States. 185
Scheduling a Workflow. 187
Creating a Reusable Scheduler. 188
Unscheduling a Workflow. 188
Disabling a Workflow. 188
Manual Workflow Runs. 189
Sending Email. 190
Sending Email Overview. 190
Configuring Email on UNIX. 191
Configuring MAPI on Windows. 192
Configuring SMTP on Windows. 194
Working with Email Tasks. 194

Working with Post-Session Email. 196
Suspension Email. 199
Using Service Variables to Address Email. 199
Tips for Sending Email. 200
Workflow Monitor. 201
Workflow Monitor Overview. 201
Using the Workflow Monitor. 202
Customizing Workflow Monitor Options. 205
Using Workflow Monitor Toolbars. 207
Working with Tasks and Workflows. 208
Workflow and Task Status. 211
Using the Gantt Chart View. 212
Using the Task View. 214
Tips for Monitoring Workflows. 215
Workflow Monitor Details. 216
Workflow Monitor Details Overview. 216
Repository Service Details. 216
Integration Service Properties. 217
Repository Folder Details. 218
Workflow Run Properties. 219
Worklet Run Properties. 220
Command Task Run Properties. 221
Session Task Run Properties. 222
Performance Details. 225
Session and Workflow Logs. 229
Session and Workflow Logs Overview. 229
Log Events. 230
Log Events Window. 232
Working with Log Files. 233
Workflow Logs. 238
Session Logs. 239
Viewing Log Events. 241
Session Properties Reference. 242
General Tab. 242
Properties Tab. 242
Mapping Tab (Transformations View). 246

Mapping Tab (Partitions View). 262
Components Tab. 262
Metadata Extensions Tab. 264
Workflow Properties Reference. 264
General Tab. 264
Properties Tab. 265
Scheduler Tab. 266
Variables Tab. 268
Events Tab. 269

Workflow Basics guide-copyright

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Preface

Use the *PowerCenter® Workflow Basics Guide* to learn how to create, run, schedule, and to monitor workflows and sessions.

You can create and configure a workflow in the Workflow Manager. Workflows are a set of instructions to run the mappings build in Designer. Workflows can contain a session and other tasks, such as an email notification. You can also schedule a workflow. You can also create worklets, which are objects that contain a set of tasks to reuse a set of workflow logic.

You can monitor the workflows and sessions in the Workflow Monitor. You can view the status of the workflows and sessions in the log and you can review detailed log events from each service in the domain.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

Workflow Manager

Workflow Manager Overview

In the Workflow Manager, you define a set of instructions called a workflow to execute mappings you build in the Designer. Generally, a workflow contains a session and any other task you may want to perform when you run a session. Tasks can include a session, email notification, or scheduling information. You connect each task with links in the workflow.

You can also create a worklet in the Workflow Manager. A worklet is an object that groups a set of tasks. A worklet is similar to a workflow, but without scheduling information. You can run a batch of worklets inside a workflow.

After you create a workflow, you run the workflow in the Workflow Manager and monitor it in the Workflow Monitor.

Workflow Manager Options

You can customize the Workflow Manager default options to control the behavior and look of the Workflow Manager tools. You can also configure options, such as grouping sessions or docking and undocking windows.

Workflow Manager Tools

To create a workflow, you first create tasks such as a session, which contains the mapping you build in the Designer. You then connect tasks with conditional links to specify the order of execution for the tasks you created. The Workflow Manager consists of three tools to help you develop a workflow:

- **Task Developer.** Use the Task Developer to create tasks you want to run in the workflow.
- **Workflow Designer.** Use the Workflow Designer to create a workflow by connecting tasks with links. You can also create tasks in the Workflow Designer as you develop the workflow.
- **Worklet Designer.** Use the Worklet Designer to create a worklet.

Workflow Tasks

You can create the following types of tasks in the Workflow Manager:

- **Assignment.** Assigns a value to a workflow variable.
- **Command.** Specifies a shell command to run during the workflow.
- **Control.** Stops or aborts the workflow.
- **Decision.** Specifies a condition to evaluate.
- **Email.** Sends email during the workflow.
- **Event-Raise.** Notifies the Event-Wait task that an event has occurred.
- **Event-Wait.** Waits for an event to occur before executing the next task.
- **Session.** Runs a mapping you create in the Designer.
- **Timer.** Waits for a timed event to trigger.

Workflow Manager Windows

The Workflow Manager displays the following windows to help you create and organize workflows:

- **Navigator.** You can connect to and work in multiple repositories and folders. In the Navigator, the Workflow Manager displays a red icon over invalid objects.
- **Workspace.** You can create, edit, and view tasks, workflows, and worklets.
- **Output.** Contains tabs to display different types of output messages. The Output window contains the following tabs:
  - **Save.** Displays messages when you save a workflow, worklet, or task. The Save tab displays a validation summary when you save a workflow or a worklet.
  - **Fetch Log.** Displays messages when the Workflow Manager fetches objects from the repository.
  - **Validate.** Displays messages when you validate a workflow, worklet, or task.
  - **Copy.** Displays messages when you copy repository objects.

- **Server.** Displays messages from the Integration Service.

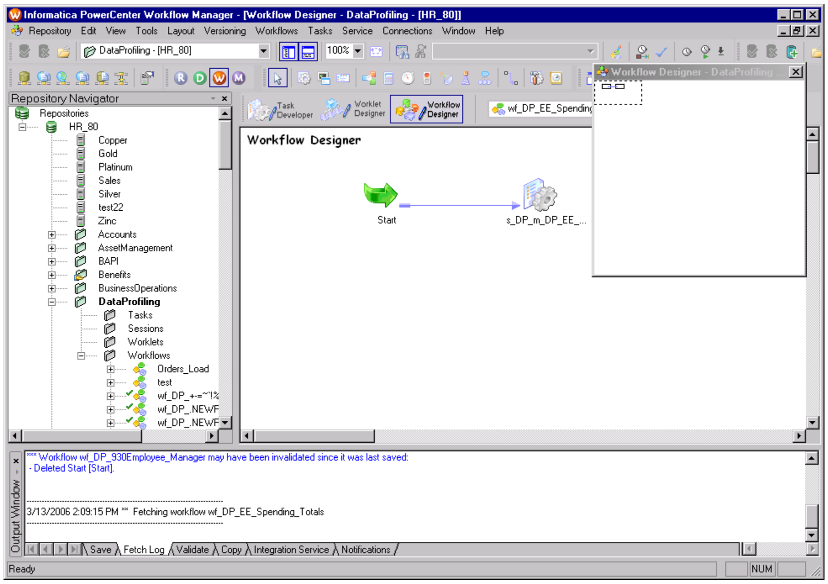
- **Notifications.** Displays messages from the Repository Service.

- **Overview.** An optional window that lets you easily view large workflows in the workspace. Outlines the visible area in the workspace and highlights selected objects in color. Click View > Overview Window to display this window.

You can view a list of open windows and switch from one window to another in the Workflow Manager. To view the list of open windows, click Window > Windows.

The Workflow Manager also displays a status bar that shows the status of the operation you perform.

The following figure shows the Workflow Manager windows:



```
*** Workflow wf_DP_930Employee_Manager may have been invalidated since it was last saved:
 - Deleted Start [Start].

----------------------------------------------------------------
3/13/2006 2:09:15 PM ** Fetching workflow wf_DP_EE_Spending_Totals
----------------------------------------------------------------
```

Setting the Date/Time Display Format

The Workflow Manager displays the date and time formats configured in the Windows Control Panel of the PowerCenter Client machine. To modify the date and time formats, display the Control Panel and open Regional Settings. Set the date and time formats on the Date and Time tabs.

Note: For the Timer task and schedule settings, the Workflow Manager displays date in short date format and the time in 24-hour format (HH:mm).

Removing an Integration Service from the Workflow Manager

You can remove an Integration Service from the Navigator. Remove an Integration Service if the Integration Service no longer exists or if you no longer use that Integration Service. When you remove an Integration Service with associated workflows, assign another Integration Service to the workflows.

1. In the Navigator, right-click on the Integration Service you want to remove.
2. Click Delete.

Workflow Manager Options

You can customize the Workflow Manager default options to control the behavior and look of the Workflow Manager tools. You can also configure the page setup for the Workflow Manager.

To configure Workflow Manager options, click Tools > Options. You can configure the following options:

- **General.** You can configure workspace options, display options, and other general options on the General tab.
- **Format.** You can configure font, color, and other format options on the Format tab.
- **Miscellaneous.** You can configure Copy Wizard and Versioning options on the Miscellaneous tab.
- **Advanced.** You can configure enhanced security for connection objects in the Advanced tab.

You can also configure the workspace layout for printing.

General Options

General options control tool behavior, such as whether or not a tool retains its view when you close it, how the Overview window behaves, and where the Workflow Manager stores workspace files.

The following table describes general options you can configure in the Workflow Manager:

Option	Description
Reload Tasks/Workflows When Opening a Folder	Reloads the last view of a tool when you open it. For example, if you have a workflow open when you disconnect from a repository, select this option so that the same workflow appears the next time you open the folder and Workflow Designer. Default is enabled.
Ask Whether to Reload the Tasks/Workflows	Appears when you select Reload tasks/workflows when opening a folder. Select this option if you want the Workflow Manager to prompt you to reload tasks, workflows, and worklets each time you open a folder. Default is disabled.
Delay Overview Window Pans	By default, when you drag the focus of the Overview window, the focus of the workbook moves concurrently. When you select this option, the focus of the workspace does not change until you release the mouse button. Default is disabled.
Arrange Workflows/ Worklets Vertically By Default	Arranges tasks in workflows vertically by default. Default is disabled.
Allow Invoking In-Place Editing Using the Mouse	By default, you can press F2 to edit objects directly in the workspace instead of opening the Edit Task dialog box. Select this option so you can also click the object name in the workspace to edit the object. Default is disabled.
Open Editor When a Task Is Created	Opens the Edit Task dialog box when you create a task. By default, the Workflow Manager creates the task in the workspace. If you do not enable this option, double-click the task to open the Edit Task dialog box. Default is disabled.
Workspace File Directory	Directory for workspace files created by the Workflow Manager. Workspace files maintain the last task or workflow you saved. This directory should be local to the PowerCenter Client to prevent file corruption or overwrites by multiple users. By default, the Workflow Manager creates files in the PowerCenter Client installation directory.

Option	Description
Display Tool Names on Views	Displays the name of the tool in the upper left corner of the workspace or workbook. Default is enabled.
Always Show the Full Name of Tasks	Shows the full name of a task when you select it. By default, the Workflow Manager abbreviates the task name in the workspace. Default is disabled.
Show the Expression on a Link	Shows the link condition in the workspace. If you do not enable this option, the Workflow Manager abbreviates the link condition in the workspace. Default is enabled.
Show Background in Partition Editor and Pushdown Optimization	Displays background color for objects in iconic view. Disable this option to remove background color from objects in iconic view. Default is disabled.
Launch Workflow Monitor when Workflow Is Started	Launches Workflow Monitor when you start a workflow or a task. Default is enabled.
Receive Notifications from Repository Service	You can receive notification messages in the Workflow Manager and view them in the Output window. Notification messages include information about objects that another user creates, modifies, or deletes. You receive notifications about sessions, tasks, workflows, and worklets. The Repository Service notifies you of the changes so you know objects you are working with may be out of date. For the Workflow Manager to receive a notification, the folder containing the object must be open in the Navigator, and the object must be open in the workspace. You also receive user-created notifications posted by the user who manages the Repository Service. Default is enabled.
Reset All	Reset all format options to the default values.

Format Options

Format options control workspace colors and fonts. You can configure format options for each Workflow Manager tool.

The following table describes the format options for the Workflow Manager:

Option	Description
Current Theme	Currently selected color theme for the Workflow Manager tools. This field is display-only.
Select Theme	Apply a color theme to the Workflow Manager tools.
Tools	Workflow Manager tool that you want to configure. When you select a tool, the configurable workspace elements appear in the list below Tools menu.
Color	Color of the selected workspace element.
Orthogonal Links	Link lines run horizontally and vertically but not diagonally in the workspace.
Solid Lines for Links	Links appear as solid lines. By default, the Workflow Manager displays orthogonal links as dotted lines.

Option	Description
Categories	Component of the Workflow Manager that you want to customize.
Change	Change the display font and language script for the selected category.
Current Font	Font of the Workflow Manager component that is currently selected in the Categories menu. This field is display-only.
Reset All	Reset all format options to the default values.

Selecting a Color Theme

Use color themes to quickly select the colors of the workspace elements in all the Workflow Manager tools. When you select a color theme, you can choose from Informatica Classic, High Contrast Black, and Color Backgrounds.

After you select a color theme for the Workflow Manager tools, you can modify the color of individual workspace elements.

To select a color theme for a Workflow Manager tool:

1. In the Workflow Manager, click Tools > Options.
2. Click the Format tab.
3. In the Color Themes section of the Format tab, click Select Theme.

   The Theme Selector dialog box appears.
4. Select a theme from the Theme menu.
5. Click the tabs in the Preview section to see how the workspace elements appear in each of the Workflow Manager tools.
6. Click OK to apply the color theme.

Miscellaneous Options

Miscellaneous options control the display settings and available functions of the Copy Wizard, versioning, and target load options. Target options control how the Integration Service loads targets. To configure the Copy Wizard, Versioning, and Target Load Type options, click Tools > Options and select the Miscellaneous tab.

The following table describes the miscellaneous options:

Option	Description
Validate Copied Objects	Validates the copied object. Enabled by default.
Generate Unique Name When Resolved to “Rename”	Generates unique names for copied objects if you select the Rename option. For example, if the workflow wf_Sales has the same name as a workflow in the destination folder, the Rename option generates the unique name wf_Sales1. Default is enabled.

Option	Description
Get Default Object When Resolved to “Choose”	Uses the object with the same name in the destination folder if you select the Choose option. Default is disabled.
Show Check Out Image in Navigator	Displays the Check Out icon when an object has been checked out. Default is enabled.
Allow Delete Without Checkout	You can delete versioned repository objects without first checking them out. You cannot, however, delete an object that another user has checked out. When you select this option, the Repository Service checks out an object to you when you delete it. Default is disabled.
Check In Deleted Objects Automatically After They Are Saved	Checks in deleted objects after you save the changes to the repository. When you clear this option, the deleted object remains checked out and you must check it in from the results view. Default is disabled.
Target Load Type	Sets default load type for sessions. You can choose normal or bulk loading. Any change you make takes effect after you restart the Workflow Manager. You can override this setting in the session properties. Default is Bulk.
Reset All	Resets all Miscellaneous options to the default values.

Enhanced Security

The Workflow Manager has an enhanced security option to specify a default set of permissions for connection objects. When you enable enhanced security, the Workflow Manager assigns default permissions on connection objects for users, groups, and others.

When you disable enable enhanced security, the Workflow Manager assigns read, write, and execute permissions to all users that would otherwise receive permissions of the default group. If you delete the owner from the repository, the Workflow Manager assigns ownership of the object to the administrator.

To enable enhanced security for connection objects:

1. Click Tools > Options.
2. Click the Advanced Tab.
3. Select Enable Enhanced Security.
4. Click OK.

Page Setup Options

Page Setup options allow you to control the layout of the workspace you are printing. You can configure headers, footers, and frame of the Workflow Manager in the Page Setup dialog box.

The following table describes the page setup options:

Option	Description
Header and Footer	Displays the window title, page number, number of pages, current date and current time in the printout of the workspace. You can also indicate the alignment of the header and footer.
Options	Adds a frame or corner to the page, shows full name of the tasks and options. You can also choose to print in color or black and white.

Navigating the Workspace

Perform the following operations to navigate the Workflow Manager workspace:

- Customize windows.
- Customize toolbars.
- Search for tasks, links, events and variables.
- Arrange objects in the workspace.
- Zoom and pan the workspace.

Customizing Workflow Manager Windows

You can customize the following options for the Workflow Manager windows:

- **Display a window.** From the menu, select View. Then select the window you want to open.
- **Close a window.** Click the small x in the upper right corner of the window.
- **Dock or undock a window.** Double-click the title bar or drag the title bar toward or away from the workspace.

Using Toolbars

The Workflow Manager can display the following toolbars to help you select tools and perform operations quickly:

- **Standard.** Contains buttons to connect to and disconnect from repositories and folders, toggle windows, zoom in and out, pan the workspace, and find objects.
- **Connections.** Contains buttons to create and edit connections, and assign Integration Services.
- **Repository.** Contains buttons to connect to and disconnect from repositories and folders, export and import objects, save changes, and print the workspace.
- **View.** Contains buttons to customize toolbars, toggle the status bar and windows, toggle full-screen view, create a new workbook, and view the properties of objects.
- **Layout.** Contains buttons to arrange and restore objects in the workspace, find objects, zoom in and out, and pan the workspace.
- **Tasks.** Contains buttons to create tasks.
- **Workflow.** Contains buttons to edit workflow properties.
- **Run.** Contains buttons to schedule the workflow, start the workflow, or start a task.

- **Versioning.** Contains buttons to check in objects, undo checkouts, compare versions, list checked-out objects, and list repository queries.
- **Tools.** Contains buttons to connect to the other PowerCenter Client applications. When you use a Tools button to open another PowerCenter Client application, PowerCenter uses the same repository connection to connect to the repository and opens the same folders.

You can perform the following operations with toolbars:

- Display or hide a toolbar.
- Create a new toolbar.
- Add or remove buttons.

Searching for Items

The Workflow Manager includes search features to help you find tasks, links, variables, events in the workspace, and text in the Output window. You can search for items in any Workflow Manager tool or Output window.

There are two ways to search for items in the workspace:

- **Find in Workspace.**
- **Find Next.**

Searching Objects Simultaneously

You can search multiple items at once and return a list of all task names, link conditions, event names, or variable names that contain the search string.

1. In any Workflow Manager tool, click the Find in Workspace toolbar button or click Edit > Find in Workspace.

   The Find in Workspace dialog box appears.
2. Choose search for tasks, links, variables, or events.
3. Enter a search string, or select a string from the list.

   The Workflow Manager saves the last 10 search strings in the list.
4. Specify whether or not to match whole words and whether or not to perform a case-sensitive search.
5. Click Find Now.

   The Workflow Manager lists task names, link conditions, event names, or variable names that match the search string at the bottom of the dialog box.
6. Click Close.

Searching Objects Individually

When you search through items one at a time, the Workflow Manager highlights the first task, link, event, variable, or text string that contains the search string. If you repeat the search, the Workflow Manager highlights the next item that contains the search string.

To find a single object:

1. To search for a task, link, event, or variable, open the appropriate Workflow Manager tool and click a task, link, or event. To search for text in the Output window, click the appropriate tab in the Output window.
2. Enter a search string in the Find field on the standard toolbar.

   The search is not case sensitive.
3. Click Edit > Find Next, click the Find Next button on the toolbar, or press Enter or F3 to search for the string.

   The Workflow Manager highlights the first task name, link condition, event name, or variable name that contains the search string, or the first string in the Output window that matches the search string.
4. To search for the next item, press Enter or F3 again.

   The Workflow Manager alerts you when you have searched through all items in the workspace or Output window before it highlights the same objects a second time.

Arranging Objects in the Workspace

The Workflow Manager can arrange objects in the workspace horizontally or vertically. In the Task Manager, you can also arrange tasks evenly in the workspace by choosing Tile. To arrange objects in the workspace, click Layout > Arrange and choose Horizontal, Vertical, or Tile. To display the links as horizontal and vertical lines, click Layout > Orthogonal Links.

Zooming the Workspace

You can zoom and pan the workspace to adjust the view. Use the toolbar or Layout menu options to set zoom levels. To maximize the size of the workspace window, click View > Full Screen. To go back to normal view, click the Close Full Screen button or press Esc.

To pan the workspace, click Layout > Pan or click the Pan button on the toolbar. Drag the focus of the workspace window and release the mouse button when it is in the appropriate position. Double-click the workspace to stop panning.

Working with Repository Objects

Use the Workflow Manager to perform the following general operations with repository objects:

- View properties for each object.
- Enter descriptions for each object.
- Rename an object.
- Refresh an object.

To edit any repository object, you must first add a repository in the Navigator so you can access the repository object. To add a repository in the Navigator, click Repository > Add. Use the Add Repositories dialog box to add the repository.

Viewing Object Properties

To view properties of a repository object, first select the repository object in the Navigator. Click View > Properties to view object properties. Or, right-click the repository object and choose Properties.

You can view properties of a folder, task, worklet, or workflow. For folders, the Workflow Manager displays folder name and whether the folder is shared. Object properties are read-only.

You can also view dependencies for repository objects.

Entering Descriptions for Repository Objects

When you edit an object in the Workflow Manager, you can enter descriptions and comments for that object. The maximum number of characters you can enter is 2,000 bytes/K, where K is the maximum number of bytes a character contains in the selected repository code page. For example, if the repository code page is a Japanese code page where each character can contain up to two bytes (K=2), each description and comment field can contain up to 1,000 characters.

Renaming Repository Objects

You can rename repository objects by clicking the Rename button in the Edit Tasks dialog box or the Edit Workflow dialog box. You can also rename repository objects by clicking the object name in the workspace and typing in the new name.

Refreshing Repository Objects

You can refresh the repository folder list or a folder to reflect its latest changes. When you refresh a folder, its contents are refreshed.

To refresh a folder, right-click the open folder, and then select Refresh.

To refresh the repository folder list, right-click the repository, and then select Refresh Folder List.

Checking In and Out Versioned Repository Objects

When you work with versioned objects, you must check out an object if you want to change it, and save it when you want to commit the changes to the repository. You must check in the object to allow other users to make changes to it. Checking in an object adds a new numbered version to the object history.

Checking In Objects

You commit changes to the repository by checking in objects. When you check in an object, the repository creates a new version of the object and assigns it a version number. The repository increments the version number by one each time it creates a new version.

To check in an object from the Workflow Manager workspace, select the object or objects and click Versioning > Check in. If you are checking in multiple objects, you can choose to apply comment to all objects.

If you want to check out or check in scheduler objects in the Workflow Manager, you can run an object query to search for them. You can also check out a scheduler object in the Scheduler Browser window when you edit the object. However, you must run an object query to check in the object.

If you want to check out or check in session configuration objects in the Workflow Manager, you can run an object query to search for them. You can also check out objects from the Session Config Browser window when you edit them.

You also can check out and check in session configuration and scheduler objects from the Repository Manager.

Viewing and Comparing Versioned Repository Objects

You can view and compare versions of objects in the Workflow Manager. If an object has multiple versions, you can find the versions of the object in the View History window. In addition to comparing versions of an object in a window, you can view the various versions of an object in the workspace to graphically compare them.

Use the following rules and guidelines when you view older versions of objects in the workspace:

- You cannot simultaneously view multiple versions of composite objects, such as workflows and worklets.
- Older versions of a composite object might not include the child objects that were used when the composite object was checked in. If you open a composite object that includes a child object version that is purged from the repository, the preceding version of the child object appears in the workspace as part of the composite object. For example, you might want to view version 5 of a workflow that originally included version 3 of a session, but version 3 of the session is purged from the repository. When you view version 5 of the workflow, version 2 of the session appears as part of the workflow.
- You cannot view older versions of sessions if they reference deleted or invalid mappings, or if they do not have a session configuration.

Opening an Older Version of an Object

When you view an older version, the version number appears as a prefix before the object name. You can simultaneously view multiple versions of a non-composite object in the workspace.

1. In the workspace or Navigator, select the object and click Versioning > View History.
2. Select the version you want to view in the workspace and click Tools > Open in Workspace.

Comparing Two Versions of an Object

You can compare two versions of an object through the workspace, Navigator, or the View History window.

1. In the workspace or Navigator, select an object and click Versioning > View History.
2. Select the versions you want to compare and click Compare > Selected Versions.

   -or-

   Select a version and click Compare > Previous Version to compare a version of the object with the previous version.

   The Diff Tool appears.

Searching for Versioned Objects

Use an object query to search for versioned objects in the repository that meet specified conditions. When you run a query, the repository returns results based on those conditions. You may want to create an object query to perform the following tasks:

- **Track repository objects during development**. You can add Label, User, Last saved, or Comments parameters to queries to track objects during development.
- **Associate a query with a deployment group**. When you create a dynamic deployment group, you can associate an object query with it.

To create an object query, click Tools > Queries to open the Query Browser.

From the Query Browser, you can create, edit, and delete queries. You can also configure permissions for each query from the Query Browser. You can run any queries for which you have read permissions from the Query Browser.

Copying Repository Objects

You can copy repository objects, such as workflows, worklets, or tasks within the same folder, to a different folder, or to a different repository. If you want to copy the object to another folder, you must open the destination folder before you copy the object into the folder.

Use the Copy Wizard in the Workflow Manager to copy objects. When you copy a workflow or a worklet, the Copy Wizard copies all of the worklets, sessions, and tasks in the workflow. You must resolve all conflicts that occur. Conflicts occur when the Copy Wizard finds a workflow or worklet with the same name in the target folder or when the connection object does not exist in the target repository. If a connection object does not exist, you can skip the conflict and choose a connection object after you copy the workflow. You cannot copy connection objects. Conflicts may also occur when you copy Session tasks.

You can configure display settings and functions of the Copy Wizard by choosing Tools > Options.

Note: Use the Import Wizard in the Workflow Manager to import objects from an XML file. The Import Wizard provides the same options to resolve conflicts as the Copy Wizard.

Copying Sessions

When you copy a Session task, the Copy Wizard looks for the database connection and associated mapping in the destination folder. If the mapping or connection does not exist in the destination folder, you can select a new mapping or connection. If the destination folder does not contain any mapping, you must first copy a mapping to the destination folder in the Designer before you can copy the session.

When you copy a session that has mapping variable values saved in the repository, the Workflow Manager either copies or retains the saved variable values.

Copying Workflow Segments

You can copy segments of workflows and worklets when you want to reuse a portion of workflow or worklet logic. A segment consists of one or more tasks, the links between the tasks, and any condition in the links. You can copy reusable and non-reusable objects when copying and pasting segments. You can copy segments of workflows or worklets into workflows and worklets within the same folder, within another folder, or within a folder in a different repository. You can also paste segments of workflows or worklets into an empty Workflow Designer or Worklet Designer workspace.

To copy a segment from a workflow or worklet:

1. Open the workflow or worklet.
2. To select a segment, highlight each task you want to copy. You can select multiple reusable or non-reusable objects. You can also select segments by dragging the pointer in a rectangle around objects in the workspace.
3. Click Edit > Copy.
4. Open the workflow or worklet into which you want to paste the segment. You can also copy the object into the Workflow or Worklet Designer workspace.
5. Click Edit > Paste.

The Copy Wizard opens, and notifies you if it finds copy conflicts.

Comparing Repository Objects

Use the Workflow Manager to compare two repository objects of the same type to identify differences between the objects. For example, if you have two similar Email tasks in a folder, you can compare them to see which one contains the attributes you need. When you compare two objects, the Workflow Manager displays their attributes in detail.

You can compare objects across folders and repositories. You must open both folders to compare the objects. You can compare a reusable object with a non-reusable object. You can also compare two versions of the same object.

You can compare the following types of objects:

- Tasks
- Sessions
- Worklets
- Workflows

You can also compare instances of the same type. For example, if the workflows you compare contain worklet instances with the same name, you can compare the instances to see if they differ. Use the Workflow Manager to compare the following instances and attributes:

- **Instances of sessions and tasks in a workflow or worklet comparison.** For example, when you compare workflows, you can compare task instances that have the same name.
- **Instances of mappings and transformations in a session comparison.** For example, when you compare sessions, you can compare mapping instances.
- **The attributes of instances of the same type within a mapping comparison.** For example, when you compare flat file sources, you can compare attributes, such as file type (delimited or fixed), delimiters, escape characters, and optional quotes.

You can compare schedulers and session configuration objects in the Repository Manager. You cannot compare objects of different types. For example, you cannot compare an Email task with a Session task.

When you compare objects, the Workflow Manager displays the results in the Diff Tool window. The Diff Tool output contains different nodes for different types of objects.

When you import Workflow Manager objects, you can compare object conflicts.

Comparing Objects

Use the following procedure to compare objects:

1. Open the folders that contain the objects you want to compare.
2. Open the appropriate Workflow Manager tool.
3. Click Tasks > Compare.

   -or-

   Click Worklets > Compare.

   -or-

   Click Workflow > Compare.
4. In the dialog box that appears, select the objects that you want to compare.
5. Click Compare.

   Tip: You can also compare objects from the Navigator or workspace. In the Navigator, select the objects, right-click and select Compare Objects. In the workspace, select the objects, right-click and select Compare Objects.
6. To view more differences between object properties, click the Compare Further icon or right-click the differences.
7. If you want to save the comparison as a text or HTML file, click File > Save to File.

Metadata Extensions

You can extend the metadata stored in the repository by associating information with individual repository objects. For example, you may want to store your name with the worklets you create. If you create a session, you can store your telephone extension with that session. You associate information with repository objects using metadata extensions. You can create and promote metadata extensions on the Metadata Extensions tab.

The following table describes the configuration options for the Metadata Extensions tab:

Metadata Extensions Tab Options	Description
Extension Name	Name of the metadata extension. Metadata extension names must be unique for each type of object in a domain. Metadata extension names cannot contain any special characters except underscores and cannot begin with numbers.
Datatype	Datatype: numeric (integer), string, boolean, or XML.

Metadata Extensions Tab Options	Description
Value	For a numeric metadata extension, the value must be an integer. For a boolean metadata extension, choose true or false. For a string or XML metadata extension, click the Edit button on the right side of the Value field to enter a value of more than one line. The Workflow Manager does not validate XML syntax.
Precision	Maximum length for string or XML metadata extensions.
Reusable	Makes the metadata extension reusable or non-reusable. Check to apply the metadata extension to all objects of this type (reusable). Clear to make the metadata extension apply to this object only (non-reusable). **Note:** If you make a metadata extension reusable, you cannot change it back to non-reusable. The Workflow Manager makes the extension reusable as soon as you confirm the action.
UnOverride	This column appears only if the value of one of the metadata extensions was changed. To restore the default value, click Revert.
Description	Description of the metadata extension.

Creating a Metadata Extension

You can create user-defined, reusable, and non-reusable metadata extensions for repository objects using the Workflow Manager. To create a metadata extension, you edit the object for which you want to create the metadata extension and then add the metadata extension to the Metadata Extensions tab.

Tip: To create multiple reusable metadata extensions, use the Repository Manager.

To create a metadata extension:

1. Open the appropriate Workflow Manager tool.
2. Drag the appropriate object into the workspace.
3. Double-click the title bar of the object to edit it.
4. Click the Metadata Extensions tab.

   This tab lists the existing user-defined and vendor-defined metadata extensions. User-defined metadata extensions appear in the User Defined Metadata Domain. If they exist, vendor-defined metadata extensions appear in their own domains.
5. Click the Add button.

   A new row appears in the User Defined Metadata Extension Domain.
6. Configure the metadata extension.
7. Click OK.

Editing a Metadata Extension

You can edit user-defined, reusable, and non-reusable metadata extensions for repository objects using the Workflow Manager. To edit a metadata extension, you edit the repository object, and then make changes to the Metadata Extensions tab.

What you can edit depends on whether the metadata extension is reusable or non-reusable. You can promote a non-reusable metadata extension to reusable, but you cannot change a reusable metadata extension to non-reusable.

Editing Reusable Metadata Extensions

If the metadata extension you want to edit is reusable and editable, you can change the value of the metadata extension, but not any of its properties. However, if the vendor or user who created the metadata extension did not make it editable, you cannot edit the metadata extension or its value.

To edit the value of a reusable metadata extension, click the Metadata Extensions tab and modify the Value field. To restore the default value for a metadata extension, click Revert in the UnOverride column.

Editing Non-Reusable Metadata Extensions

If the metadata extension you want to edit is non-reusable, you can change the value of the metadata extension and its properties. You can also promote the metadata extension to a reusable metadata extension.

To edit a non-reusable metadata extension, click the Metadata Extensions tab. You can update the Datatype, Value, Precision, and Description fields.

To make the metadata extension reusable, select Reusable. If you make a metadata extension reusable, you cannot change it back to non-reusable. The Workflow Manager makes the extension reusable as soon as you confirm the action.

To restore the default value for a metadata extension, click Revert in the UnOverride column.

Deleting a Metadata Extension

You can delete metadata extensions for repository objects. You delete reusable metadata extensions using the Repository Manager. Use the Workflow Manager to delete non-reusable metadata extensions. Edit the repository object and then delete the metadata extension from the Metadata Extensions tab.

Expression Editor

The Workflow Manager provides an Expression Editor for any expression in the workflow. You can enter expressions using the Expression Editor for Link conditions, Decision tasks, and Assignment tasks.

The Expression Editor displays built-in variables, user-defined workflow variables, and predefined workflow variables such as $Session.status.

The Expression Editor also displays the following functions:

- **Transformation language functions.** SQL-like functions designed to handle common expressions.
- **User-defined functions.** Functions you create in PowerCenter based on transformation language functions.
- **Custom functions.** Functions you create with the Custom Function API.

Adding Comments

You can add comments using -- or // comment indicators with the Expression Editor. Use comments to give descriptive information about the expression, or you can specify a valid URL to access business documentation about the expression.

Validating Expressions

Use the Validate button to validate an expression. If you do not validate an expression, the Workflow Manager validates it when you close the Expression Editor. You cannot run a workflow with invalid expressions.

Expressions in link conditions and Decision task conditions must evaluate to a numeric value. Workflow variables used in expressions must exist in the workflow.

Expression Editor Display

The Expression Editor can display syntax expressions in different colors for better readability. If you have the latest Rich Edit control, riched20.dll, installed on the system, the Expression Editor displays expression functions in blue, comments in grey, and quoted strings in green.

You can resize the Expression Editor. Expand the dialog box by dragging from the borders. The Workflow Manager saves the new size for the dialog box as a client setting.

Keyboard Shortcuts

When editing a repository object or maneuvering around the Workflow Manager, use the following Keyboard shortcuts to help you complete different operations quickly.

The following table lists the Workflow Manager keyboard shortcuts for editing a repository object:

Task	Shortcut
Cancel editing in a cell.	Esc
Select and clear a check box.	Space Bar
Copy text from a cell onto the clipboard.	Ctrl+C
Cut text from a cell onto the clipboard.	Ctrl+X
Edit the text of a cell.	F2
Find all combination and list boxes.	Type the first letter on the list.
Find tables or fields in the workspace.	Ctrl+F
Move around cells in a dialog box.	Ctrl+directional arrows
Paste copied or cut text from the clipboard into a cell.	Ctrl+V
Select the text of a cell.	F2

The following table lists the Workflow Manager keyboard shortcuts for navigating in the workspace:

Task	Shortcut
Create links.	Ctrl+F2. Press Ctrl+F2 to select first task you want to link. Press Tab to select the rest of the tasks you want to link. Press Ctrl+F2 again to link all the tasks you selected.
Edit task name in the workspace.	F2
Expand selected node and all its children.	SHIFT + * (use asterisk on numeric keypad)
Move across Select tasks in the workspace.	Tab
Select multiple tasks.	Ctrl+mouse click

Workflows and Worklets

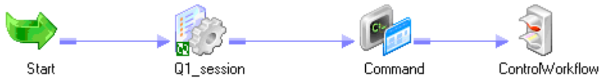
Workflows Overview

A workflow is a set of instructions that tells the Integration Service how to run tasks such as sessions, email notifications, and shell commands. After you create tasks in the Task Developer and Workflow Designer, you connect the tasks with links to create a workflow.

In the Workflow Designer, you can specify conditional links and use workflow variables to create branches in the workflow. The Workflow Manager also provides Event-Wait and Event-Raise tasks to control the sequence of task execution in the workflow. You can also create worklets and nest them inside the workflow.

Every workflow contains a Start task, which represents the beginning of the workflow.

The following figure shows a sample workflow:



You can create workflows with branches to run tasks concurrently.

When you create a workflow, select an Integration Service to run the workflow. You can start the workflow using the Workflow Manager, Workflow Monitor, or *pmcmd*.

Use the Workflow Monitor to see the progress of a workflow during its run. The Workflow Monitor can also show the history of a workflow.

Use the following guidelines when you develop a workflow:

1. **Create a workflow.** Create a workflow in the Workflow Designer or by using the Workflow Generation Wizard in the PowerCenter Designer.
2. **Add tasks to the workflow.** You might have already created tasks in the Task Developer. Or, you can add tasks to the workflow as you develop the workflow in the Workflow Designer.

3. **Connect tasks with links.** After you add tasks to the workflow, connect them with links to specify the order of execution in the workflow.
4. **Specify conditions for each link.** You can specify conditions on the links to create branches and dependencies.
5. **Validate workflow.** Validate the workflow in the Workflow Designer to identify errors.
6. **Save workflow.** When you save the workflow, the Workflow Manager validates the workflow and updates the repository.
7. **Run workflow.** In the workflow properties, select an Integration Service to run the workflow. Run the workflow from the Workflow Manager, Workflow Monitor, or *pmcmd*. You can monitor the workflow in the Workflow Monitor.

Related Topics:

- "Manual Workflow Runs" on page 189
- "Workflow Monitor" on page 201
- "Workflow Properties Reference" on page 264

Creating a Workflow

A workflow must contain a Start task. The Start task represents the beginning of a workflow. When you create a workflow, the Workflow Designer creates a Start task and adds it to the workflow. You cannot delete the Start task.

After you create a workflow, you can add tasks to the workflow. The Workflow Manager includes tasks such as the Session, Command, and Email tasks.

Finally, you connect workflow tasks with links to specify the order of execution in the workflow. You can add conditions to links.

When you edit a workflow, the Repository Service updates the workflow information when you save the workflow. If a workflow is running when you make edits, the Integration Service uses the updated information the next time you run the workflow.

You can also create a workflow through the Workflow Wizard in the Workflow Manager or the Workflow Generation Wizard in the PowerCenter Designer.

Creating a Workflow Manually

Use the following procedure to create a workflow manually:

1. Open the Workflow Designer.
2. Click Workflows > Create.
3. Enter a name for the new workflow.
4. Click OK.

   The Workflow Designer creates a Start task in the workflow.

Creating a Workflow Automatically

Use the following procedure to create a workflow automatically:

1. Open the Workflow Designer. Close any open workflow.
2. Click the session button on the Tasks toolbar.
3. Click in the Workflow Designer workspace.

   The Mappings dialog box appears.
4. Select a mapping to associate with the session and click OK.

   The Create Workflow dialog box appears. The Workflow Designer names the workflow wf_*SessionName* by default. You can rename the workflow or change other workflow properties.
5. Click OK.

   The Workflow Designer creates a workflow for the session.

Adding Tasks to Workflows

After you create a workflow, you add tasks you want to run in the workflow. You may already have created tasks in the Task Developer. Or, you may want to create tasks in the Workflow Designer as you develop the workflow.

If you have already created tasks in the Task Developer, add them to the workflow by dragging the tasks from the Navigator to the Workflow Designer workspace.

To create and add tasks as you develop the workflow, click Tasks > Create in the Workflow Designer. Or, use the Tasks toolbar to create and add tasks to the workflow. Click the button on the Tasks toolbar for the task you want to create. Click again in the Workflow Designer workspace to create and add the task.

Tasks you create in the Workflow Designer are non-reusable. Tasks you create in the Task Developer are reusable.

Deleting a Workflow

You may decide to delete a workflow that you no longer use. When you delete a workflow, you delete all non-reusable tasks and reusable task instances associated with the workflow. Reusable tasks used in the workflow remain in the folder when you delete the workflow.

If you delete a workflow that is running, the Integration Service aborts the workflow. If you delete a workflow that is scheduled to run, the Integration Service removes the workflow from the schedule.

You can delete a workflow in the Navigator window, or you can delete the workflow currently displayed in the Workflow Designer workspace:

- To delete a workflow from the Navigator window, open the folder, select the workflow and press the Delete key.
- To delete a workflow currently displayed in the Workflow Designer workspace, click Workflows > Delete.

Using the Workflow Wizard

Use the Workflow Wizard to automate the process of creating sessions, adding sessions to a workflow, and linking sessions to create a workflow. The Workflow Wizard creates sessions from mappings and adds them to the workflow. It also creates a Start task and lets you schedule the workflow. You can add tasks and edit other workflow properties after the Workflow Wizard completes. If you want to create concurrent sessions, use the Workflow Designer to manually build a workflow.

Before you create a workflow, verify that the folder contains a valid mapping for the Session task.

Complete the following steps to build a workflow using the Workflow Wizard:

1. Assign a name and Integration Service to the workflow.
2. Create a session.
3. Schedule the workflow.

You can also use the Workflow Generation Wizard in the PowerCenter Designer to generate sessions and workflows.

Step 1. Assign a Name and Integration Service to the Workflow

In the first step of the Workflow Wizard, you add the name and description of the workflow and choose the Integration Service to run the workflow.

1. In the Workflow Manager, open the folder containing the mapping you want to use in the workflow.
2. Open the Workflow Designer.
3. Click Workflows > Wizard.

   The Workflow Wizard appears.
4. Enter a name for the workflow.

   The convention for naming workflows is wf_*WorkflowName*.
5. Enter a description for the workflow.
6. Select the Integration Service to run the workflow and click Next.

Step 2. Create a Session

In the second step of the Workflow Wizard, you create a session based on a mapping. You can add tasks later in the Workflow Designer workspace.

1. In the second step of the Workflow Wizard, select a valid mapping and click the right arrow button.

   The Workflow Wizard creates a Session task in the right pane using the selected mapping and names it s_*MappingName* by default.
2. You can select additional mappings to create more Session tasks in the workflow.

   When you add multiple mappings to the list, the Workflow Wizard creates sequential sessions in the order you add them.
3. Use the arrow buttons to change the session order.
4. Specify whether the session should be reusable.

   When you create a reusable session, use the session in other workflows.

5. Specify how you want the Integration Service to run the workflow.

   You can specify that the Integration Service runs sessions only if previous sessions complete, or you can specify that the Integration Service always runs each session. When you select this option, it applies to all sessions you create using the Workflow Wizard.

Step 3. Schedule a Workflow

In the third step of the Workflow Wizard, you can schedule a workflow to run continuously, repeat at a specified time or interval, or start manually. The Integration Service runs a workflow unless the prior workflow run fails.

When you configure a task, you can configure the workflow to fail if the task fails. If you configure the workflow to fail when a task fails, the Integration Service removes the workflow from the schedule, and you must reschedule it. You can reschedule the workflow through the Workflow Manager or through *pmcmd*. If you do not configure the workflow to fail when a task fails, the Integration Service reschedules the workflow.

1. In the third step of the Workflow Wizard, configure the scheduling and run options.
2. Click **Next**.

   The Workflow Wizard displays the settings for the workflow.
3. Verify the workflow settings, then click **Finish**. To edit settings, click **Back**.

   The completed workflow opens in the Workflow Designer workspace. From the workspace, you can add tasks, create concurrent sessions, add conditions to links, or change properties.

Assigning an Integration Service

Before you can run a workflow, you must assign an Integration Service to run it. You can choose an Integration Service to run a workflow by editing the workflow properties. You can also assign an Integration Service from the menu. When you assign a service from the menu, you can assign multiple workflows without editing each workflow.

Assigning a Service from the Workflow Properties

Use the following procedure to assign a service within the workflow properties:

1. In the Workflow Designer, open the Workflow.
2. Click Workflows > Edit.

   The Edit Workflow dialog box appears.
3. On the General tab, click the Browse Integration Services button.

   A list of Integration Services appears.
4. Select the Integration Service that you want to run the workflow.
5. Click OK twice to select the Integration Service for the workflow.

Assigning a Service from the Menu

When you assign an Integration Service to a workflow you overwrite the service selected in the workflow properties.

1. Close all folders in the repository.
2. Click Service > Assign Integration Service.

   The Assign Integration Service dialog box appears.
3. From the Choose Integration Service list, select the service you want to assign.
4. From the Show Folder list, select the folder you want to view. Or, click All to view workflows in all folders in the repository.
5. Click the Selected check box for each workflow you want the Integration Service to run.
6. Click Assign.

Workflow Reports (Deprecated)

You can view PowerCenter Repository Reports for workflows in the Workflow Manager. When you view the report, the Workflow Manager launches JasperReports Server in a browser window and displays the report.

An administrator uses the Administrator tool to create a Reporting and Dashboards Service and adds a reporting source for the service. The reporting source must be the PowerCenter repository that contains the workflows that you want to report on.

The Workflow Composite Report includes information about the following components in a workflow:

- **Tasks.** Tasks contained in the workflow.
- **Events.** User-defined and built-in events in the workflow.
- **Variables.** User-defined and built-in variables in the workflow.

Viewing a Workflow Report

View the Workflow Composite Report to get more information about the workflow tasks, events, and variables in a workflow.

To view a Workflow Composite Report:

1. In the Workflow Manager, open a workflow.
2. Right-click in the workspace and choose View Workflow Report.

The Workflow Manager launches JasperReports Server in the default browser for the client machine and runs the Workflow Composite Report.

Working with Worklets

A worklet is an object representing a set of tasks created to reuse a set of workflow logic in multiple workflows. You can create a worklet in the Worklet Designer.

To run a worklet, include the worklet in a workflow. The workflow that contains the worklet is called the parent workflow. When the Integration Service runs a worklet, it expands the worklet to run tasks and evaluate links within the worklet. It writes information about worklet execution in the workflow log.

Suspending Worklets

When you choose Suspend on Error for the parent workflow, the Integration Service also suspends the worklet if a task in the worklet fails. When a task in the worklet fails, the Integration Service stops executing the failed task and other tasks in its path. If no other task is running in the worklet, the worklet status is “Suspended.” If one or more tasks are still running in the worklet, the worklet status is “Suspending.” The Integration Service suspends the parent workflow when the status of the worklet is “Suspended” or “Suspending.”

Developing a Worklet

To develop a worklet, you must first create a worklet. After you create a worklet, configure worklet properties and add tasks to the worklet. You can create reusable worklets in the Worklet Designer. You can also create non-reusable worklets in the Workflow Designer as you develop the workflow.

Creating a Reusable Worklet

You can create reusable worklets in the Worklet Designer. You can view a list of reusable worklets in the Navigator Worklets node.

1. In the **Worklet Designer**, click **Worklet** > **Create**.

   The **Create Worklet** dialog box appears.

2. Enter a name for the worklet.
3. If you want to add the worklet to a workflow that is enabled for concurrent execution, enable the worklet for concurrent execution.
4. Click **OK**.

   The **Worklet Designer** creates a Start task in the worklet.

Creating a Non-Reusable Worklet

You can create a non-reusable worklet in the Workflow Designer as you develop the workflow. Non-reusable worklets exist only in the workflow. You cannot use a non-reusable worklet in another workflow. After you create the worklet in the Workflow Designer, open the worklet to edit it in the Worklet Designer.

1. In the **Workflow Designer**, open a workflow.
2. Click **Tasks** > **Create**.
3. For the **Task** type, select **Worklet**.
4. Enter a name for the task.
5. Click **Create**.

   The **Workflow Designer** creates the worklet and adds it to the workspace.

6. Click **Done**.

Note: You can promote non-reusable worklet to reusable worklet by selecting the Make Reusable option in the worklet properties in a non-versioned repository. In a versioned repository, the reusable option is unavailable. To rename a non-reusable worklet, open the worklet properties in the Workflow Designer.

Configuring Worklet Properties

When you use a worklet in a workflow, you can configure the same set of general task settings on the General tab as those that you configure for any other task. For example, you can make a worklet reusable, disable a worklet, configure the input link to the worklet, or fail the parent workflow based on the worklet.

In addition to general task settings, you can configure the following worklet properties:

- **Worklet variables.** Use worklet variables to reference values and record information. You use worklet variables the same way you use workflow variables. You can assign a workflow variable to a worklet variable to override its initial value.
- **Events.** To use the Event-Wait and Event-Raise tasks in the worklet, you must first declare an event in the worklet properties.
- **Metadata extension.** Extend the metadata stored in the repository by associating information with repository objects.

RELATED TOPICS:

- "Metadata Extensions" on page 22
- "Working with the Event Task" on page 59

Adding Tasks in Worklets

After you create a worklet, add tasks by opening the worklet in the Worklet Designer. A worklet must contain a Start task. The Start task represents the beginning of a worklet. When you create a worklet, the Worklet Designer creates a Start task for you.

1. Create a non-reusable worklet in the Workflow Designer workspace.
2. Right-click the worklet and choose Open Worklet.

   The Worklet Designer opens so you can add tasks in the worklet.
3. Add tasks in the worklet by using the Tasks toolbar or click Tasks > Create in the Worklet Designer.
4. Connect tasks with links.

Declaring Events in Worklets

Use Event-Wait and Event-Raise tasks in a worklet like you would use workflows. To use the Event-Raise task, you first declare a user-defined event in the worklet. Events in one instance of a worklet do not affect events in other instances of the worklet. You cannot specify worklet events in the Event tasks in the parent workflow.

Viewing Links in a Worklet

When you edit a workflow or worklet, you can view the forward or backward link paths to other tasks. You can highlight paths to see links in the workflow branch from the Start task to the last task in the branch.

Nesting Worklets

You can nest a worklet within another worklet. When you run a workflow containing nested worklets, the Integration Service runs the nested worklet from within the parent worklet. You can group several worklets together by function or simplify the design of a complex workflow when you nest worklets.

You might choose to nest worklets to load data to fact and dimension tables. Create a nested worklet to load fact and dimension data into a staging area. Then, create a nested worklet to load the fact and dimension data from the staging area to the data warehouse.

You might choose to nest worklets to simplify the design of a complex workflow. Nest worklets that can be grouped together within one worklet. To nest an existing reusable worklet, click Tasks > Insert Worklet. To create a non-reusable nested worklet, click Tasks > Create, and select worklet.

Workflow Links

Use links to connect each task in a workflow or worklet. You can specify conditions with links to create branches. The Workflow Manager does not allow you to use links to create loops. Each link in the workflow or worklet can run only once.

After you create links between tasks, you can create conditions for each link to determine the order of operation in the workflow. If you do not specify conditions for each link, the Integration Service runs the next task in the workflow or worklet by default.

Use predefined or user-defined workflow and worklet variables in the link condition. If the link condition evaluates to True, the Integration Service runs the next task in the workflow or worklet. If the link condition evaluates to False, the Integration Service does not run the next task.

You can view results of link evaluation during workflow runs in the workflow log file.

Linking Two Tasks

Link tasks manually when you do not want to link multiple tasks.

1. In the Tasks toolbar, click the Link Tasks button.
2. In the workspace, click the first task you want to connect and drag it to the second task.
3. A link appears between the two tasks.

Linking Tasks Concurrently

Link tasks concurrently when you want to link one task to multiple tasks.

1. In the workspace, click the first task you want to connect.
2. Ctrl-click all other tasks you want to connect.

   Note: Do not use Ctrl+A or Edit > Select All to choose tasks.
3. Click Tasks > Link Concurrent.

   A link appears between the first task you selected and each task you added. The first task you selected links to each task concurrently.

Linking Tasks Sequentially

Link tasks sequentially when you want to link tasks in order between one task and each subsequent task you add.

1. In the workspace, click the first task you want to connect.
2. Ctrl-click the next task you want to connect. Continue to add tasks in the order you want them to run.
3. Click Tasks > Link Sequential.

Creating Link Conditions

Use link conditions to specify the order of execution or to create branches.

1. In the Workflow Designer or Worklet Designer workspace, double-click the link you want to specify.

   The Expression Editor appears.
2. In the Expression Editor, enter the link condition.

   The Expression Editor provides predefined workflow and worklet variables, user-defined workflow and worklet variables, variable functions, and boolean and arithmetic operators.
3. Validate the expression using the Validate button.

   The Workflow Manager displays validation results in the Output window.

   Tip: Drag the end point of a link to move it from one task to another without losing the link condition.

Example of Link Conditions

A workflow has two Session tasks, s_STORES_CA and s_STORES_AZ. You want the Integration Service to run the second Session task only if the first Session task has no target failed rows.

To accomplish this, you can set the following link condition between the sessions so that the s_STORES_AZ runs only if the number of failed target rows for S_STORES_CA is zero:

```
$s_STORES_CA.TgtFailedRows = 0
```

After you specify the link condition in the Expression Editor, the Workflow Manager validates the link condition and displays it next to the link in the workflow or worklet.

Viewing Links in a Workflow or Worklet

When you edit a workflow or worklet, you can view the forward or backward link paths to other tasks. You can highlight paths to see links in the workflow branch from the Start task to the last task in the branch.

1. In the Workflow Designer or Worklet Designer workspace, right-click a task and choose Highlight Path.
2. Select Forward Path, Backward Path, or Both.

   The Workflow Manager highlights all links in the branch you select.

Deleting Links in a Workflow or Worklet

When you edit a workflow or worklet, you can delete multiple links at once without deleting the connected tasks.

1. In the Workflow Designer or Worklet Designer workspace, select all links you want to delete.

   Tip: Use the mouse to drag the selection, or you can Ctrl-click the tasks and links.
2. Click Edit > Delete Links.

   The Workflow Manager removes all selected links.

Sessions

Sessions Overview

A session is a set of instructions that tells the Integration Service how and when to move data from sources to targets. A session is a type of task, similar to other tasks available in the Workflow Manager. In the Workflow Manager, you configure a session by creating a Session task. To run a session, you must first create a workflow to contain the Session task.

When you create a Session task, enter general information such as the session name, session schedule, and the Integration Service to run the session. You can select options to run pre-session shell commands, send On-Success or On-Failure email, and use FTP to transfer source and target files.

Configure the session to override parameters established in the mapping, such as source and target location, source and target type, error tracing levels, and transformation attributes. You can also configure the session to collect performance details for the session and store them in the PowerCenter repository. You might view performance details for a session to tune the session.

You can run as many sessions in a workflow as you need. You can run the Session tasks sequentially or concurrently, depending on the requirement.

The Integration Service creates several files and in-memory caches depending on the transformations and options used in the session.

Session Task

You create a Session task for each mapping that you want the Integration Service to run. The Integration Service uses the instructions configured in the session to move data from sources to targets.

You can create a reusable Session task in the Task Developer. You can also create non-reusable Session tasks in the Workflow Designer as you develop the workflow. After you create the session, you can edit the session properties at any time.

Note: Before you create a Session task, you must configure the Workflow Manager to communicate with databases and the Integration Service. You must assign appropriate permissions for any database, FTP, or external loader connections you configure.

Creating a Session Task

Create the Session task in the Task Developer or the Workflow Designer. Session tasks created in the Task Developer are reusable.

1. In the Task Developer or Workflow Designer, click Tasks > Create.
2. Select Session Task for the task type.
3. Enter a name for the Session task. Do not use the period character (.) in Session task names. PowerCenter does not allow a Session task name with the period character.
4. Click Create.
5. Select the mapping you want to use in the Session task and click OK.
6. Click Done.

Editing a Session

After you create a session, you can edit it. For example, you might need to adjust the buffer and cache sizes, modify the update strategy, or clear a variable value saved in the repository.

Double-click the Session task to open the session properties. The session has the following tabs, and each of those tabs has multiple settings:

- **General tab.** Enter session name, mapping name, and description for the Session task, assign resources, and configure additional task options.
- **Properties tab.** Enter session log information, test load settings, and performance configuration.
- **Config Object tab.** Enter advanced settings, log options, and error handling configuration.
- **Mapping tab.** Enter source and target information, override transformation properties, and configure the session for partitioning.
- **Components tab.** Configure pre- or post-session shell commands and emails.
- **Metadata Extension tab.** Configure metadata extension options.

You can edit session properties at any time. The repository updates the session properties immediately.

If the session is running when you edit the session, the repository updates the session when the session completes. If the mapping changes, the Workflow Manager might issue a warning that the session is invalid. The Workflow Manager then lets you continue editing the session properties. After you edit the session properties, the Integration Service validates the session and reschedules the session.

RELATED TOPICS:

- "Session Validation" on page 181
- "Session Properties Reference" on page 242

Applying Attributes to All Instances

When you edit the session properties, you can apply source, target, and transformation settings to all instances of the same type in the session. You can also apply settings to all partitions in a pipeline. You can apply reader or writer settings, connection settings, and properties settings.

For example, you might need to change a relational connection from a test to a production database for all the target instances in a session. On the Mapping tab, you can change the connection value for one target in a session and apply the connection to the other relational target objects.

The following table shows the options you can use to apply attributes to objects in a session. You can apply different options depending on whether the setting is a reader or writer, connection, or an object property.

Setting	Option	Description
Reader Writer	Apply Type to All Instances	Applies a reader or writer type to all instances of the same object type in the session. For example, you can apply a relational reader type to all the other readers in the session.
Reader Writer	Apply Type to All Partitions	Applies a reader or writer type to all the partitions in a pipeline. For example, if you have four partitions, you can change the writer type in one partition for a target instance. Use this option to apply the change to the other three partitions.
Connections	Apply Connection Type	Applies the same type of connection to all instances. Connection types are relational, FTP, queue, application, or external loader.
Connections	Apply Connection Value	Apply a connection value to all instances or partitions. The connection value defines a specific connection that you can view in the connection browser. You can apply a connection value that is valid for the existing connection type.
Connections	Apply Connection Attributes	Apply only the connection attribute values to all instances or partitions. Each type of connection has different attributes. You can apply connection attributes separately from connection values.
Connections	Apply Connection Data	Apply the connection value and its connection attributes to all the other instances that have the same connection type. This option combines the connection option and the connection attribute option.
Connections	Apply All Connection Information	Applies the connection value and its attributes to all the other instances even if they do not have the same connection type. This option is similar to Apply Connection Data, but it lets you change the connection type.
Properties	Apply Attribute to all Instances	Applies an attribute value to all instances of the same object type in the session. For example, if you have a relational target you can choose to truncate a table before you load data. You can apply the attribute value to all the relational targets in the session.
Properties	Apply Attribute to all Partitions	Applies an attribute value to all partitions in a pipeline. For example, you can change the name of the reject file name in one partition for a target instance, then apply the file name change to the other partitions.

Applying Connection Settings

When you apply connection settings you can apply the connection type, connection value, and connection attributes. You can only apply a connection value that is valid for a connection type unless you choose the Apply All Connection Information option. For example, if a target instance uses an FTP connection, you can only choose an FTP connection value to apply to it. The Apply All Connection Information option lets you apply a new connection type, connection value, and connection attributes.

Applying Attributes to Partitions or Instances

When you apply attributes to all instances or partitions in a session, you must open the session and edit one of the session objects. You apply attributes or properties to other instances by choosing an attribute in that object and selecting to apply its value to the other instances or partitions.

To apply attributes to all instances or partitions:

1. Open a session in the workspace.
2. Click the Mappings tab.
3. Choose a source, target, or transformation instance from the Navigator. Settings for properties, connections, and readers or writers might display, depending on the object you choose.
4. Right-click a reader, writer, property, or connection value.

   A list of options appears.
5. Select an option from the list and choose to apply it to all instances or all partitions.
6. Click OK to apply the attribute or property.

Performance Details

You can configure a session to collect performance details and store them in the PowerCenter repository. Collect performance data for a session to view performance details while the session runs. Write performance data for a session in the PowerCenter repository to store and view performance details for previous session runs.

If you want to write performance data to the repository you must perform the following tasks:

- Configure the session to collect performance data.
- Configure the session to write performance data to repository.
- Configure Integration Service to persist run-time statistics to the repository at the verbose level.

The Workflow Monitor displays performance details for each session that is configured to collect or write performance details.

Configuring Performance Details

You can collect performance details for a session to view while the session runs and to store in the repository for future reference.

1. In the Workflow Manager, open the session properties and select the Properties tab.
2. Select Collect performance data to view performance details while the session runs.

3. Select Write Performance Data to Repository to store and view performance details for previous session runs.

   You must also configure the Integration Service to store the run-time information at the verbose level.

4. Click OK.

Pre- and Post-Session Commands

You can create pre- and post-session commands to perform tasks before and after a session. Use SQL commands to perform database tasks. Use shell commands to perform operating system tasks.

Pre- and Post-Session SQL Commands

You can specify pre- and post-session SQL in the Source Qualifier transformation and the target instance when you create a mapping. When you create a Session task in the Workflow Manager you can override the SQL commands on the Mapping tab. You might want to use these commands to drop indexes on the target before the session runs, and then recreate them when the session completes.

The Integration Service runs pre-session SQL commands before it reads the source. It runs post-session SQL commands after it writes to the target.

You can use parameters and variables in SQL executed against the source and target. Use any parameter or variable type that you can define in the parameter file. You can enter a parameter or variable within the SQL statement, or you can use a parameter or variable as the command. For example, you can use a session parameter, $ParamMyPreSQL, as the source pre-session SQL command, and set $ParamMyPreSQL to the SQL statement in the parameter file.

Guidelines for Entering Pre- and Post-Session SQL Commands

Use the following guidelines when creating the SQL statements:

- Use any command that is valid for the database type. However, the Integration Service does not allow nested comments, even though the database might.
- Use a semicolon (;) to separate multiple statements. The Integration Service issues a commit after each statement.
- The Integration Service ignores semicolons within /* ...*/.
- If you need to use a semicolon outside of comments, you can escape it with a backslash (\).
- The Workflow Manager does not validate the SQL.

Error Handling

You can configure error handling on the Config Object tab. You can choose to stop or continue the session if the Integration Service encounters an error issuing the pre- or post- session SQL command.

Using Pre- and Post-Session Shell Commands

The Integration Service can perform shell commands at the beginning of the session or at the end of the session. Shell commands are operating system commands. Use pre- or post-session shell commands, for example, to delete a reject file or session log, or to archive target files before the session begins.

The Workflow Manager provides the following types of shell commands for each Session task:

- **Pre-session command.** The Integration Service performs pre-session shell commands at the beginning of a session. You can configure a session to stop or continue if a pre-session shell command fails.
- **Post-session success command.** The Integration Service performs post-session success commands only if the session completed successfully.
- **Post-session failure command.** The Integration Service performs post-session failure commands only if the session failed to complete.

Use the following guidelines to call a shell command:

- Use any valid UNIX command or shell script for UNIX nodes, or any valid DOS or batch file for Windows nodes.
- Configure the session to run the pre- or post-session shell commands.

The Workflow Manager provides a task called the Command task that lets you configure shell commands anywhere in the workflow. You can choose a reusable Command task for the pre- or post-session shell command. Or, you can create non-reusable shell commands for the pre- or post-session shell commands.

If you create a non-reusable pre- or post-session shell command, you can make it into a reusable Command task.

The Workflow Manager lets you choose from the following options when you configure shell commands:

- **Create non-reusable shell commands.** Create a non-reusable set of shell commands for the session. Other sessions in the folder cannot use this set of shell commands.
- **Use an existing reusable Command task.** Select an existing Command task to run as the pre- or post-session shell command.

Configure pre- and post-session shell commands in the Components tab of the session properties.

Using Parameters and Variables

You can use parameters and variables in pre- and post-session commands. Use any parameter or variable type that you can define in the parameter file. You can enter a parameter or variable within the command, or you can use a parameter or variable as the command. For example, you can include service process variable $PMTargetFileDir in the command text in pre- and post-session commands. When you use a service process variable instead of entering a specific directory, you can run the same workflow on different Integration Services without changing session properties. You can also use a session parameter, $ParamMyCommand, as the pre- or post-session shell command, and set $ParamMyCommand to the command in a parameter file.

Configuring Non-Reusable Shell Commands

When you create non-reusable pre- or post-session shell commands, the commands are only visible in the session properties. The Workflow Manager does not create Command tasks from these non-reusable commands. You can convert a non-reusable shell command to a reusable Command task.

To create non-reusable pre- or post-session shell commands:

1. In the Components tab of the session properties, select Non-reusable for pre- or post-session shell command.
2. Click the Edit button in the Value field to open the Edit Pre- or Post-Session Command dialog box.
3. Enter a name for the command in the General tab.
4. If you want the Integration Service to perform the next command only if the previous command completed successfully, select Fail Task if Any Command Fails in the Properties tab.
5. In the Commands tab, click the Add button to add shell commands.

   Enter one command for each line.
6. Click OK.

Creating a Reusable Command Task from Pre- or Post-Session Commands

If you create non-reusable pre- or post-session shell commands, you can make them into a reusable Command task. After you make the pre- or post-session shell commands into a reusable Command task, you cannot revert back.

To create a Command Task from non-reusable pre- or post-session shell commands, click the Edit button to open the Edit dialog box for the shell commands. In the General tab, select the Make Reusable check box.

After you select the Make Reusable check box and click OK, a new Command task appears in the Tasks folder in the Navigator window. Use this Command task in other workflows, just as you do with any other reusable workflow tasks.

Configuring Reusable Shell Commands

Use the following procedure to call an existing reusable Command task as the pre- or post-session shell command for the Session task:

1. In the Components tab of the session properties, click Reusable for the pre- or post-session shell command.
2. Click the Edit button in the Value field to open the Task Browser dialog box.
3. Select the Command task you want to run as the pre- or post-session shell command.
4. Click the Override button in the Task Browser dialog box if you want to change the order of the commands, or if you want to specify whether to run the next command when the previous command fails.

   Changes you make to the Command task from the session properties only apply to the session. In the session properties, you cannot edit the commands in the Command task.
5. Click OK to select the Command task for the pre- or post-session shell command.

   The name of the Command task you select appears in the Value field for the shell command.

Pre-Session Shell Command Errors

You can configure the session to stop or continue if a pre-session shell command fails. If you select stop, the Integration Service stops the session, but continues with the rest of the workflow. If you select

Continue, the Integration Service ignores the errors and continues the session. By default the Integration Service stops the session upon shell command errors.

Configure the session to stop or continue if a pre-session shell command fails in the Error Handling settings on the Config Object tab.

Session Configuration Object

Session Configuration Object Overview

Each folder in the repository has a default session configuration object that contains session properties such as commit and load settings, log options, and error handling settings. You can create multiple configuration objects if you want to apply different configuration settings to multiple sessions.

When you create a session, the Workflow Manager applies the default configuration object settings to the Config Object tab of the session. You can also choose a configuration object to use for the session.

When you edit a session configuration object, each session that uses the session configuration object inherits the changes. When you override the configuration object settings in the Session task, the session configuration object does not inherit changes.

Configuration Object and Config Object Tab Settings

You can configure the following settings in a session configuration object or on the Config Object tab in session properties:

- **Advanced.** Advanced settings allow you to configure constraint-based loading, lookup caches, and buffer sizes.
- **Log options.** Log options allow you to configure how you want to save the session log. By default, the Log Manager saves only the current session log.
- **Error handling.** Error Handling settings allow you to determine if the session fails or continues when it encounters pre-session command errors, stored procedure errors, or a specified number of session errors.
- **Partitioning options**. Partitioning options allow the Integration Service to determine the number of partitions to create at run time.
- **Session on grid**. When Session on Grid is enabled, the Integration Service distributes session threads to the nodes in a grid to increase performance and scalability.

Advanced Settings

Advanced settings allow you to configure constraint-based loading, lookup caches, and buffer sizes.

The following table describes the Advanced settings of the Config Object tab:

Advanced Settings	Description
Constraint Based Load Ordering	Integration Service loads targets based on primary key-foreign key constraints where possible.
Cache Lookup() Function	If selected, the Integration Service caches PowerMart 3.5 LOOKUP functions in the mapping, overriding mapping-level LOOKUP configurations. If not selected, the Integration Service performs lookups on a row-by-row basis, unless otherwise specified in the mapping.
Default Buffer Block Size	Size of buffer blocks used to move data from sources to targets. By default, this value is set to auto. You can specify auto or a numeric value. The default unit is bytes. Append KB, MB, or GB to the value to specify other units. For example, 1048576 or 1024KB or 1MB.
Line Sequential Buffer Length	Number of bytes that the PowerCenter Integration Service reads for each line. Increase this setting from the default of 1024 bytes if source flat file records are larger than 1024 bytes.
Maximum Partial Session Log Files	The maximum number of partial log files to save. Configure this option with Session Log File Max Size or Session Log File Max Time Period. Default is one.
Maximum Memory Allowed for Auto Memory Attributes	Maximum memory allocated for automatic cache when you configure the Integration Service to determine session cache size at run time. You enable automatic memory settings by configuring a value for this attribute. The default unit is bytes. Append KB, MB, or GB to the value to specify other units. For example, 1048576 or 1024KB or 1MB.
Maximum Percentage of Total Memory Allowed for Auto Memory Attributes	Maximum percentage of memory allocated for automatic cache when you configure the Integration Service to determine session cache size at run time.
Additional Concurrent Pipelines for Lookup Cache Creation	Restricts the number of pipelines that the Integration Service can create concurrently to pre-build lookup caches. Configure this property when the Pre-build Lookup Cache property is enabled for a session or transformation. When the Pre-build Lookup Cache property is enabled, the Integration Service creates a lookup cache before the Lookup transformation receives the data. If the session has multiple Lookup transformations, the Integration Service creates an additional pipeline for each lookup cache that it builds. To configure the number of pipelines that the Integration Service can create concurrently, select Auto or enter a numeric value: - Auto. The Integration Service determines the number of pipelines it can create at run time. - Numeric value. The Integration Service can create the specified number of pipelines to create lookup caches.
Custom Properties	Configure custom properties of the Integration Service for the session. You can override custom properties that the Integration Service uses after the DTM process has started. The Integration Service also writes the override value of the property to the session log.

Advanced Settings	Description
Pre-build Lookup Cache	Allows the Integration Service to build the lookup cache before the Lookup transformation receives the data. The Integration Service can build multiple lookup cache files at the same time to improve performance. You can configure this option in the mapping or the session. The Integration Service uses the session-level setting if you configure the Lookup transformation option as Auto. Configure one of the following options: - Auto. The Integration Service uses the value configured in the session. - Always allowed. The Integration Service can build the lookup cache before the Lookup transformation receives the first source row. The Integration Service creates an additional pipeline to build the cache. - Always disallowed. The Integration Service cannot build the lookup cache before the Lookup transformation receives the first row. You must configure the number of pipelines that the Integration Service can build concurrently. Configure the Additional Concurrent Pipelines for Lookup Cache Creation session property. The Integration Service can pre-build lookup cache if this property is greater than zero.
DateTime Format String	Date time format defined in the session configuration object. Default format specifies microseconds: MM/DD/YYYY HH24:MI:SS.US. You can specify seconds, milliseconds, or nanoseconds. MM/DD/YYYY HH24:MI:SS, specifies seconds. MM/DD/YYYY HH24:MI:SS.MS, specifies milliseconds. MM/DD/YYYY HH24:MI:SS.US, specifies microseconds. MM/DD/YYYY HH24:MI:SS.NS, specifies nanoseconds.
Pre 85 Timestamp Compatibility	Trims subseconds to maintain compatibility with versions prior to 8.5. The Integration Service converts the Oracle Timestamp datatype to the Oracle Date datatype. The Integration Service trims subsecond data for the following sources, targets, and transformations: - Relational sources and targets - XML sources and targets - SQL transformation - XML Generator transformation - XML Parser transformation Default is disabled.

Log Options Settings

Configure log options to define how to save backward compatible session log files. By default, the Log Manager saves the current session log. You can save multiple log files. You can configure a real-time session to split the session log file into multiple files. You can limit the commit statistics messages by defining how often to write commit statistics to the session log.

The following table shows the Log Options settings of the Config Object tab:

Log Options Settings	Description
Save Session Log By	Configure this option to save session log files. If you select Save Session Log by Timestamp, the Log Manager saves all session logs, appending a time stamp to each log. If you select Save Session Log by Runs, the Log Manager saves a designated number of session logs. Configure the number of sessions in the Save Session Log for These Runs option. You can also use the $PMSessionLogCount service variable to save the configured number of session logs for the Integration Service.
Save Session Log for These Runs	Number of historical session logs you want the Log Manager to save. The Log Manager saves the number of historical logs you specify, plus the most recent session log. When you configure five runs, the Log Manager saves the most recent session log, plus historical logs 0-4. You can configure up to 2,147,483,647 historical logs. If you configure zero logs, the Log Manager saves the most recent session log.
Session Log File Max Size	Maximum number of megabytes for a session log file. Configure a maximum size to enable log file rollover. When the log file reaches the maximum size, the Integration Service creates a another log file. If you set the size to zero the session log file size has no limit. Configure this option for real-time sessions that generate large session logs. The Integration Service writes the session logs to multiple files. Each file is a partial log file. Default is zero.
Session Log File Max Time Period	Maximum number of hours that the Integration Service writes to a session log file. Configure the maximum period to enable log file rollover by time. When the period is over, the Integration service creates another log file. Configure this option for real-time sessions that might generate large session logs. The Integration Service writes the session logs to multiple files. Each file is a partial log file. Default is zero.
Maximum Partial Session Log Files	Maximum number of session log files to save. The Integration Service overwrites the oldest partial log file if the number of log files has reached the limit. Configure this option in conjunction with the maximum time period or maximum file size option. You must configure one of these options to enable session log rollover. If you set the maximum number to 0, the number of session log files is unlimited. Default is 1.
Writer Commit Statistics Log Frequency	Frequency that the Integration Service writes commit statistics in the session log. The Integration Service writes commit statistics to the session log after the specified number of commits occurs. The Integration Service writes commit statistics after each commit. Default is 1.
Writer Commit Statistics Log Interval	Time interval, in minutes, to write commit statistics to the session log. The Integration Service writes commit statistics to the session log after each time interval.

Related Topics:

- "Session Logs" on page 239

Error Handling Settings

Error Handling settings allow you to determine if the session fails or continues when it encounters pre-session command errors, stored procedure errors, or a specified number of session errors.

The following table describes the Error handling settings of the Config Object tab:

Error Handling Settings	Description
Stop On Errors	Indicates how many non-fatal errors the Integration Service can encounter before it stops the session. Non-fatal errors include reader, writer, and DTM errors. Enter the number of non-fatal errors you want to allow before stopping the session. The Integration Service maintains an independent error count for each source, target, and transformation. If you specify 0, non-fatal errors do not cause the session to stop. Optionally use the $PMSessionErrorThreshold service variable to stop on the configured number of errors for the Integration Service.
Override Tracing	Overrides tracing levels set on a transformation level. Selecting this option enables a menu from which you choose a tracing level: None, Terse, Normal, Verbose Initialization, or Verbose Data.
On Stored Procedure Error	Required if the session uses pre- or post-session stored procedures. If you select Stop Session, the Integration Service stops the session on errors executing a pre-session or post-session stored procedure. If you select Continue Session, the Integration Service continues the session regardless of errors executing pre-session or post-session stored procedures. By default, the Integration Service stops the session on Stored Procedure error and marks the session failed.
On Pre-Session Command Task Error	Required if the session has pre-session shell commands. If you select Stop Session, the Integration Service stops the session on errors executing pre-session shell commands. If you select Continue Session, the Integration Service continues the session regardless of errors executing pre-session shell commands. By default, the Integration Service stops the session upon error.
On Pre-Post SQL Error	Required if the session uses pre- or post-session SQL. If you select Stop Session, the Integration Service stops the session errors executing pre-session or post-session SQL. If you select Continue, the Integration Service continues the session regardless of errors executing pre-session or post-session SQL. By default, the Integration Service stops the session upon pre- or post-session SQL error and marks the session failed.
Error Log Type	Specifies the type of error log to create. You can specify relational, file, or no log. Default is none. **Note:** You cannot log row errors from XML file sources. You can view the XML source errors in the session log.

Error Handling Settings	Description
Error Log DB Connection	Specifies the database connection for a relational error log.
Error Log Table Name Prefix	Specifies table name prefix for a relational error log. Oracle and Sybase have a 30 character limit for table names. If a table name exceeds 30 characters, the session fails.
Error Log File Directory	Specifies the directory where errors are logged. By default, the error log file directory is $PMBadFilesDir\.
Error Log File Name	Specifies error log file name. By default, the error log file name is PMError.log.
Log Row Data	Specifies whether or not to log transformation row data. When you enable error logging, the Integration Service logs transformation row data by default. If you disable this property, n/a or -1 appears in transformation row data fields.
Log Source Row Data	Specifies whether or not to log source row data. By default, the check box is clear and source row data is not logged.
Data Column Delimiter	Delimiter for string type source row data and transformation group row data. By default, the Integration Service uses a pipe (|) delimiter. Verify that you do not use the same delimiter for the row data as the error logging columns. If you use the same delimiter, you may find it difficult to read the error log file.

Partitioning Options Settings

When you configure dynamic partitioning, the Integration Service determines the number of partitions to create at run time. Configure dynamic partitioning on the Config Object tab of session properties.

The following table describes the Partitioning Options settings on the Config Object tab:

Partitioning Options Settings	Description
Dynamic Partitioning	Configure dynamic partitioning using one of the following methods: - Disabled. Do not use dynamic partitioning. Define the number of partitions on the Mapping tab. - Based on number of partitions. Sets the partitions to a number that you define in the Number of Partitions attribute. Use the $DynamicPartitionCount session parameter, or enter a number greater than 1. - Based on number of nodes in grid. Sets the partitions to the number of nodes in the grid running the session. If you configure this option for sessions that do not run on a grid, the session runs in one partition and logs a message in the session log. - Based on source partitioning. Determines the number of partitions using database partition information. The number of partitions is the maximum of the number of partitions at the source. - Based on number of CPUs. Sets the number of partitions equal to the number of CPUs on the node that prepares the session. If the session is configured to run on a grid, dynamic partitioning sets the number of partitions equal to the number of CPUs on the node that prepares the session multiplied by the number of nodes in the grid. Default is disabled.
Number of Partitions	Determines the number of partitions that the Integration Service creates when you configure dynamic partitioning based on the number of partitions. Enter a value greater than 1 or use the $DynamicPartitionCount session parameter.

Session on Grid Settings

When Session on Grid is enabled, the Integration Service distributes workflows and session threads to the nodes in a grid to increase performance and scalability.

The following table describes the Session on Grid setting on the Config Object tab:

Session on Grid Setting	Description
Is Enabled	Specifies whether the session runs on a grid.

Creating a Session Configuration Object

Create a session configuration object when you want to reuse a set of Config Object tab settings.

1. In the Workflow Manager, open a folder and click Tasks > Session Configuration.

   The Session Configuration Browser appears.
2. Click New to create a new session configuration object.
3. Enter a name for the session configuration object.
4. On the Properties tab, configure the settings.
5. Click OK.

Configuring a Session to Use a Session Configuration Object

After you create a session configuration object, you can configure sessions to use it.

1. In the Workflow Manager, open the session properties and click the Config Object tab.
2. Click the Open button in the Config Name field.

   A list of session configuration objects appears.
3. Select the configuration object you want to use and click OK.

   The settings associated with the configuration object appear on the Config Object tab.
4. Click OK.

Tasks

Tasks Overview

The Workflow Manager contains many types of tasks to help you build workflows and worklets. You can create reusable tasks in the Task Developer. Or, create and add tasks in the Workflow or Worklet Designer as you develop the workflow.

The following table summarizes workflow tasks available in Workflow Manager:

Task Name	Tool	Reusable	Description
Assignment	Workflow Designer Worklet Designer	No	Assigns a value to a workflow variable.
Command	Task Developer Workflow Designer Worklet Designer	Yes	Specifies shell commands to run during the workflow. You can choose to run the Command task if the previous task in the workflow completes.
Control	Workflow Designer Worklet Designer	No	Stops or aborts the workflow.
Decision	Workflow Designer Worklet Designer	No	Specifies a condition to evaluate in the workflow. Use the Decision task to create branches in a workflow.
Email	Task Developer Workflow Designer Worklet Designer	Yes	Sends email during the workflow.
Event-Raise	Workflow Designer Worklet Designer	No	Represents the location of a user-defined event. The Event-Raise task triggers the user-defined event when the Integration Service runs the Event-Raise task.
Event-Wait	Workflow Designer Worklet Designer	No	Waits for a user-defined or a predefined event to occur. Once the event occurs, the Integration Service completes the rest of the workflow.

Task Name	Tool	Reusable	Description
Session	Task Developer Workflow Designer Worklet Designer	Yes	Set of instructions to run a mapping.
Timer	Workflow Designer Worklet Designer	No	Waits for a specified period of time to run the next task.

The Workflow Manager validates tasks attributes and links. If a task is invalid, the workflow becomes invalid. Workflows containing invalid sessions may still be valid.

Creating a Task

You can create tasks in the Task Developer, or you can create them in the Workflow Designer or the Worklet Designer as you develop the workflow or worklet. Tasks you create in the Task Developer are reusable. Tasks you create in the Workflow Designer and Worklet Designer are non-reusable by default.

Creating a Task in the Task Developer

Use the Task Developer to create Command, Session, and Email tasks.

1. In the Task Developer, click Tasks > Create.
2. Select the task type you want to create, Command, Session, or Email.
3. Enter a name for the task. Do not use the period character (.) in task names. Workflow Manager does not allow a task name with the period character.
4. For session tasks, select the mapping you want to associate with the session.
5. Click Create.

   The Task Developer creates the workflow task.
6. Click Done to close the Create Task dialog box.

Creating a Task in the Workflow or Worklet Designer

You can create and add tasks in the Workflow Designer or Worklet Designer as you develop the workflow or worklet. You can create any type of task in the Workflow Designer or Worklet Designer. Tasks you create in the Workflow Designer or Worklet Designer are non-reusable. Edit the General tab of the task properties to promote a non-reusable task to a reusable task.

To create tasks in the Workflow Designer or Worklet Designer:

1. In the Workflow Designer or Worklet Designer, open a workflow or worklet.
2. Click Tasks > Create.
3. Select the type of task you want to create.
4. Enter a name for the task.
5. Click Create.

   The Workflow Designer or Worklet Designer creates the task and adds it to the workspace.

6. Click Done.

Configuring Tasks

After you create the task, you can configure general task options on the General tab. For each task instance in the workflow, you can configure how the Integration Service runs the task and the other objects associated with the selected task. You can also disable the task so you can run rest of the workflow without the selected task.

When you use a task in the workflow, you can edit the task in the Workflow Designer and configure the following task options in the General tab:

- **Fail parent if this task fails.** Choose to fail the workflow or worklet containing the task if the task fails.
- **Fail parent if this task does not run.** Choose to fail the workflow or worklet containing the task if the task does not run.
- **Disable this task.** Choose to disable the task so you can run the rest of the workflow without the task.
- **Treat input link as AND or OR.** Choose to have the Integration Service run the task when all or one of the input link conditions evaluates to True.

Reusable Workflow Tasks

Workflows can contain reusable task instances and non-reusable tasks. Non-reusable tasks exist within a single workflow. Reusable tasks can be used in multiple workflows in the same folder.

You can create any task as non-reusable or reusable. Tasks you create in the Task Developer are reusable. Tasks you create in the Workflow Designer are non-reusable by default. However, you can edit the general properties of a task to promote it to a reusable task.

The Workflow Manager stores each reusable task separate from the workflows that use the task. You can view a list of reusable tasks in the Tasks node in the Navigator window. You can see a list of all reusable Session tasks in the Sessions node in the Navigator window.

Promoting a Non-Reusable Workflow Task

You can promote a non-reusable workflow task to a reusable task. Reusable tasks must have unique names within the repository. When you promote a non-reusable task, the repository checks for naming conflicts. If a reusable task with the same name already exists, the repository appends a number to the reusable task name to make it unique. The repository applies the appended name to the checked-out version and to the latest checked-in version of the reusable task.

To promote a non-reusable workflow task:

1. In the Workflow Designer, double-click the task you want to make reusable.
2. In the General tab of the Edit Task dialog box, select the Make Reusable option.
3. When prompted whether you are sure you want to promote the task, click Yes.
4. Click OK.

   The newly promoted task appears in the list of reusable tasks in the Tasks node in the Navigator window.

Instances and Inherited Changes

When you add a reusable task to a workflow, you add an instance of the task. The definition of the task exists outside the workflow, while an instance of the task exists in the workflow.

You can edit the task instance in the Workflow Designer. Changes you make in the task instance exist only in the workflow. The task definition remains unchanged in the Task Developer.

When you make changes to a reusable task definition in the Task Developer, the changes reflect in the instance of the task in the workflow if you have not edited the instance.

Reverting Changes in Reusable Tasks Instances

When you edit an instance of a reusable task in the workflow, you can revert back to the settings in the task definition. When you change settings in the task instance, the Revert button appears. The Revert button appears after you override task properties. You cannot use the Revert button for settings that are read-only or locked by another user.

AND or OR Input Links

For each task, you can choose to treat the input link as an AND link or an OR link. When a task has one input link, the Integration Service processes the task when the previous object completes and the link condition evaluates to True. If you have multiple links going into one task, you can choose to have an AND input link so that the Integration Service runs the task when all the link conditions evaluates to True. Or, you can choose to have an OR input link so that the Integration Service runs the task as soon as any link condition evaluates to True.

To set the type of input links, double-click the task to open the Edit Tasks dialog box. Select AND or OR for the input link type.

Disabling Tasks

In the Workflow Designer, you can disable a workflow task so that the Integration Service runs the workflow without the disabled task. The status of a disabled task is DISABLED. Disable a task in the workflow by selecting the Disable This Task option in the Edit Tasks dialog box.

Failing Parent Workflow or Worklet

You can choose to fail the workflow or worklet if a task fails or does not run. The workflow or worklet that contains the task instance is called the parent. A task might not run when the input condition for the task evaluates to False.

To fail the parent workflow or worklet if the task fails, double-click the task and select the Fail Parent If This Task Fails option in the General tab. When you select this option and a task fails, it does not prevent the other tasks in the workflow or worklet from running. Instead, the Integration Service marks the status of the workflow or worklet as failed. If you have a session nested within multiple worklets, you must select the Fail Parent If This Task Fails option for each worklet instance to see the failure at the workflow level.

To fail the parent workflow or worklet if the task does not run, double-click the task and select the Fail Parent If This Task Does Not Run option in the General tab. When you choose this option, the Integration Service fails the parent workflow if a task did not run.

Note: The Integration Service does not fail the parent workflow if you disable a task.

Working with the Assignment Task

You can assign a value to a user-defined workflow variable with the Assignment task. To use an Assignment task in the workflow, first create and add the Assignment task to the workflow. Then configure the Assignment task to assign values or expressions to user-defined variables. After you assign a value to a variable using the Assignment task, the Integration Service uses the assigned value for the variable during the remainder of the workflow. You must create a variable before you can assign values to it. You cannot assign values to predefined workflow variables.

To create an Assignment task:

1. In the Workflow Designer, click Tasks > Create.
2. Select Assignment Task for the task type.
3. Enter a name for the Assignment task. Click Create. Then click Done.

   The Workflow Designer creates and adds the Assignment task to the workflow.
4. Double-click the Assignment task to open the Edit Task dialog box.
5. On the Expressions tab, click Add to add an assignment.
6. Click the Open button in the User Defined Variables field.
7. Select the variable for which you want to assign a value. Click OK.
8. Click the Edit button in the Expression field to open the Expression Editor.

   The Expression Editor shows predefined workflow variables, user-defined workflow variables, variable functions, and boolean and arithmetic operators.
9. Enter the value or expression you want to assign.

   For example, if you want to assign the value 500 to the user-defined variable $$custno1, enter the number 500 in the Expression Editor.
10. Click Validate.

    Validate the expression before you close the Expression Editor.
11. Repeat steps 6 to 8 to add more variable assignments.

    Use the up and down arrows in the Expressions tab to change the order of the variable assignments.
12. Click OK.

Command Task

You can specify one or more shell commands to run during the workflow with the Command task. For example, you can specify shell commands in the Command task to delete reject files, copy a file, or archive target files.

Use a Command task in the following ways:

- **Standalone Command task.** Use a Command task anywhere in the workflow or worklet to run shell commands.
- **Pre- and post-session shell command.** You can call a Command task as the pre- or post-session shell command for a Session task.

Use any valid UNIX command or shell script for UNIX servers, or any valid DOS or batch file for Windows servers. For example, you might use a shell command to copy a file from one directory to another. For a Windows server you would use the following shell command to copy the SALES_ ADJ file from the source directory, L, to the target, H:

```
copy L:\sales\sales_adj H:\marketing\
```

For a UNIX server, you would use the following command to perform a similar operation:

```
cp sales/sales_adj marketing/
```

Each shell command runs in the same environment as the Integration Service. Environment settings in one shell command script do not carry over to other scripts. To run all shell commands in the same environment, call a single shell script that invokes other scripts.

Using Parameters and Variables

You can use parameters and variables in standalone Command tasks and pre- and post-session shell commands. For example, you might use a service process variable instead of hard-coding a directory name.

You can use the following parameters and variables in commands:

- **Standalone Command tasks.** You can use service, service process, workflow, and worklet variables in standalone Command tasks. You cannot use session parameters, mapping parameters, or mapping variables in standalone Command tasks. The Integration Service does not expand these types of parameters and variables in standalone Command tasks.
- **Pre- and post-session shell commands.** You can use any parameter or variable type that you can define in the parameter file.

Assigning Resources

You can assign resources to Command task instances in the Worklet or Workflow Designer. You might want to assign resources to a Command task if you assign the workflow to an Integration Service associated with a grid. When you assign a resource to a Command task and the Integration Service is configured to check resources, the Load Balancer dispatches the task to a node that has the resource available. A task fails if the Load Balancer cannot find a node where the required resource is available.

Creating a Command Task

Complete the following steps to create a Command task:

1. In the Workflow Designer or the Task Developer, click Task > Create.
2. Select Command Task for the task type.
3. Enter a name for the Command task. Click Create. Then click Done.
4. Double-click the Command task in the workspace to open the Edit Tasks dialog box.
5. In the Commands tab, click the Add button to add a command.
6. In the Name field, enter a name for the new command.
7. In the Command field, click the Edit button to open the Command Editor.
8. Enter the command you want to run. Enter one command in the Command Editor. You can use service, service process, workflow, and worklet variables in the command.

9. Click OK to close the Command Editor.
10. Repeat steps 4 to 9 to add more commands in the task.
11. Optionally, click the General tab in the Edit Tasks dialog to assign resources to the Command task.
12. Click OK.

If you specify non-reusable shell commands for a session, you can promote the non-reusable shell commands to a reusable Command task.

Executing Commands in the Command Task

The Integration Service runs shell commands in the order you specify them. If the Load Balancer has more Command tasks to dispatch than the Integration Service can run at the time, the Load Balancer places the tasks it cannot run in a queue. When the Integration Service becomes available, the Load Balancer dispatches tasks from the queue in the order determined by the workflow service level.

You can choose to run a command only if the previous command completed successfully. Or, you can choose to run all commands in the Command task, regardless of the result of the previous command. If you configure multiple commands in a Command task to run on UNIX, each command runs in a separate shell.

If you choose to run a command only if the previous command completes successfully, the Integration Service stops running the rest of the commands and fails the task when one of the commands in the Command task fails. If you do not choose this option, the Integration Service runs all the commands in the Command task and treats the task as completed, even if a command fails. If you want the Integration Service to perform the next command only if the previous command completes successfully, select Fail Task if Any Command Fails in the Properties tab of the Command task.

You can choose a recovery strategy for the task. The recovery strategy determines how the Integration Service recovers the task when you configure workflow recovery and the task fails. You can configure the task to restart or you can configure the task to fail and continue running the workflow.

Log Files and Command Tasks

When the Integration Service processes a Command task, it creates temporary files in $PMTempDir. It writes temporary process files to $PMTempDir before it writes them to the log files. After it writes the process files to the log files, it deletes them from $PMTempDir. If the Integration Service shuts down before it deletes the process files, you must delete them manually. The process file names begin with is.process.

Control Task

Use the Control task to stop, abort, or fail the top-level workflow or the parent workflow based on an input link condition. A parent workflow or worklet is the workflow or worklet that contains the Control task.

The following table describes the options you can configure in the Control task:

Control Option	Description
Fail Me	Marks the Control task as “Failed.” The Integration Service fails the Control task if you choose this option. If you choose Fail Me in the Properties tab and choose Fail Parent If This Task Fails in the General tab, the Integration Service fails the parent workflow.
Fail Parent	Marks the status of the workflow or worklet that contains the Control task as failed after the workflow or worklet completes.
Stop Parent	Stops the workflow or worklet that contains the Control task.
Abort Parent	Aborts the workflow or worklet that contains the Control task.
Fail Top-Level Workflow	Fails the workflow that is running.
Stop Top-Level Workflow	Stops the workflow that is running.
Abort Top-Level Workflow	Aborts the workflow that is running.

Creating a Control Task

Create a Control task in the workflow to stop, abort, or fail the workflow based on an input link condition.

1. In the Workflow Designer, click Tasks > Create.
2. Select Control Task for the task type.
3. Enter a name for the Control task.
4. Click Create, and then click Done.

   The Workflow Manager creates and adds the Control task to the workflow.
5. Double-click the Control task in the workspace to open it.
6. Configure the control options on the Properties tab.

Working with the Decision Task

You can enter a condition that determines the execution of the workflow, similar to a link condition with the Decision task. The Decision task has a predefined variable called *$Decision_task_name*.condition that represents the result of the decision condition. The Integration Service evaluates the condition in the Decision task and sets the predefined condition variable to True (1) or False (0).

You can specify one decision condition per Decision task. After the Integration Service evaluates the Decision task, use the predefined condition variable in other expressions in the workflow to help you develop the workflow.

Depending on the workflow, you might use link conditions instead of a Decision task. However, the Decision task simplifies the workflow. If you do not specify a condition in the Decision task, the Integration Service evaluates the Decision task to True.

Using the Decision Task

Use the Decision task instead of multiple link conditions in a workflow. Instead of specifying multiple link conditions, use the predefined condition variable in a Decision task to simplify link conditions.

Example

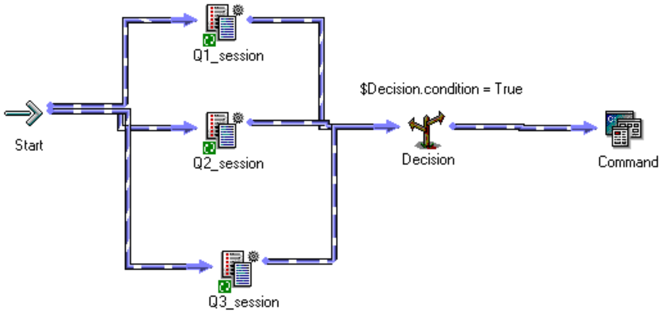
For example, you have a Command task that depends on the status of the three sessions in the workflow. You want the Integration Service to run the Command task when any of the three sessions fails. To accomplish this, use a Decision task with the following decision condition:

```
$Q1_session.status = FAILED OR $Q2_session.status = FAILED OR $Q3_session.status =
FAILED
```

You can then use the predefined condition variable in the input link condition of the Command task. Configure the input link with the following link condition:

```
$Decision.condition = True
```

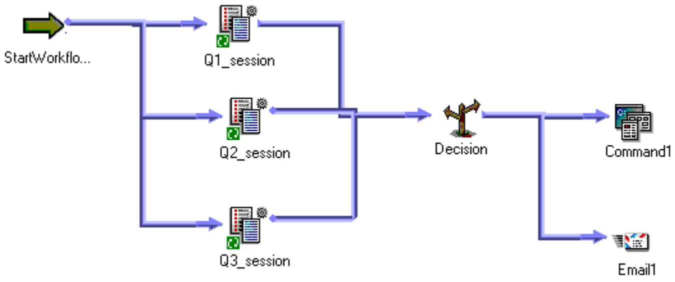
The following figure shows a sample workflow using a Decision task:



You can configure the same logic in the workflow without the Decision task. Without the Decision task, you need to use three link conditions and treat the input links to the Command task as OR links.

You can further expand the workflow. The Integration Service runs the Command task if any of the three Session tasks fails. Suppose now you want the Integration Service to also run an Email task if all three Session tasks succeed. To do this, add an Email task and use the decision condition variable in the link condition.

The following figure shows the expanded sample workflow using a Decision task:



Creating a Decision Task

Complete the following steps to create a Decision task:

1. In the Workflow Designer, click Tasks > Create.
2. Select Decision Task for the task type.
3. Enter a name for the Decision task. Click Create. Then click Done.

   The Workflow Designer creates and adds the Decision task to the workspace.
4. Double-click the Decision task to open it.
5. Click the Open button in the Value field to open the Expression Editor.
6. In the Expression Editor, enter the condition you want the Integration Service to evaluate.

   Validate the expression before you close the Expression Editor.
7. Click OK.

Working with the Event Task

You can define events in the workflow to specify the sequence of task execution. The event is triggered based on the completion of the sequence of tasks. Use the following tasks to help you use events in the workflow:

- **Event-Raise task.** Event-Raise task represents a user-defined event. When the Integration Service runs the Event-Raise task, the Event-Raise task triggers the event. Use the Event-Raise task with the Event-Wait task to define events.
- **Event-Wait task.** The Event-Wait task waits for an event to occur. Once the event triggers, the Integration Service continues executing the rest of the workflow.

To coordinate the execution of the workflow, you may specify the following types of events for the Event-Wait and Event-Raise tasks:

- **Predefined event.** A predefined event is a file-watch event. For predefined events, use an Event-Wait task to instruct the Integration Service to wait for the specified indicator file to appear before continuing with the rest of the workflow. When the Integration Service locates the indicator file, it starts the next task in the workflow.
- **User-defined event.** A user-defined event is a sequence of tasks in the workflow. Use an Event-Raise task to specify the location of the user-defined event in the workflow. A user-defined event is sequence of tasks in the branch from the Start task leading to the Event-Raise task.

  When all the tasks in the branch from the Start task to the Event-Raise task complete, the Event-Raise task triggers the event. The Event-Wait task waits for the Event-Raise task to trigger the event before continuing with the rest of the tasks in its branch.

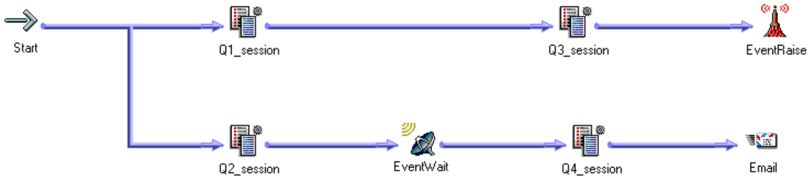
Related Topics:

- “Configuring Worklet Properties” on page 33
- “Metadata Extensions” on page 22

Example of User-Defined Events

Say you have four sessions you want to run in a workflow. You want Q1_session and Q2_session to run concurrently to save time. You also want to run Q3_session after Q1_session completes. You want to run Q4_session only when Q1_session, Q2_session, and Q3_session complete.

The following workflow shows how to accomplish this using the Event-Raise and Event-Wait tasks:



To configure the workflow, complete the following steps:

1. Link Q1_session and Q2_session concurrently.
2. Add Q3_session after Q1_session.
3. Declare an event called Q1Q3_Complete in the Events tab of the workflow properties.
4. In the workspace, add an Event-Raise task after Q3_session.
5. Specify the Q1Q3_Complete event in the Event-Raise task properties. This allows the Event-Raise task to trigger the event when Q1_session and Q3_session complete.
6. Add an Event-Wait task after Q2_session.
7. Specify the Q1Q3_Complete event for the Event-Wait task.
8. Add Q4_session after the Event-Wait task. When the Integration Service processes the Event-Wait task, it waits until the Event-Raise task triggers Q1Q3_Complete before it runs Q4_session.

The Integration Service runs the workflow in the following order:

1. The Integration Service runs Q1_session and Q2_session concurrently.
2. When Q1_session completes, the Integration Service runs Q3_session.
3. The Integration Service finishes executing Q2_session.
4. The Event-Wait task waits for the Event-Raise task to trigger the event.
5. The Integration Service completes Q3_session.
6. The Event-Raise task triggers the event, Q1Q3_complete.
7. The Integration Service runs Q4_session because the event, Q1Q3_Complete, has been triggered.
8. The Integration Service runs the Email task.

Event-Raise Tasks

The Event-Raise task represents the location of a user-defined event. A user-defined event is the sequence of tasks in the branch from the Start task to the Event-Raise task. When the Integration Service runs the Event-Raise task, the Event-Raise task triggers the user-defined event.

To use an Event-Raise task, you must first declare the user-defined event. Then, create an Event-Raise task in the workflow to represent the location of the user-defined event you just declared. In the Event-Raise task properties, specify the name of a user-defined event.

Declaring a User-Defined Event

Declare a user-defined event to use in conjunction with an Event-Raise task.

1. In the Workflow Designer, click Workflow > Edit.
2. Select the Events tab in the Edit Workflow dialog box.
3. Click the Add button to add an event name.

   Event name is not case sensitive.
4. Click OK.

Using the Event-Raise Task for a User-Defined Event

After you declare a user-defined event, use the Event-Raise task to represent the location of the event and to trigger the event.

1. In the Workflow Designer workspace, create an Event-Raise task and place it in the workflow to represent the user-defined event you want to trigger.

   A user-defined event is the sequence of tasks in the branch from the Start task to the Event-Raise task.
2. Double-click the Event-Raise task to open it.
3. On the Properties tab, click the Open button in the Value field to open the Events Browser for user-defined events.
4. Choose an event in the Events Browser.
5. Click OK twice.

Event-Wait Tasks

The Event-Wait task waits for a predefined event or a user-defined event. A predefined event is a file-watch event. When you use the Event-Wait task to wait for a predefined event, you specify an indicator file for the Integration Service to watch. The Integration Service waits for the indicator file to appear. Once the indicator file appears, the Integration Service continues running tasks after the Event-Wait task.

You can assign resources to Event-Wait tasks that wait for predefined events. You may want to assign a resource to a predefined Event-Wait task if you are running on a grid and the indicator file appears on a specific node or in a specific directory. When you assign a resource to a predefined Event-Wait task and the Integration Service is configured to check resources, the Load Balancer distributes the task to a node where the required resource is available.

Note: If you use the Event-Raise task to trigger the event when you wait for a predefined event, you may not be able to successfully recover the workflow.

You can also use the Event-Wait task to wait for a user-defined event. To use the Event-Wait task for a user-defined event, specify the name of the user-defined event in the Event-Wait task properties. The Integration Service waits for the Event-Raise task to trigger the user-defined event. Once the user-defined event is triggered, the Integration Service continues running tasks after the Event-Wait task.

Waiting for User-Defined Events

Use the Event-Wait task to wait for a user-defined event. A user-defined event is triggered by the Event-Raise task. To wait for a user-defined event, you must first use an Event-Raise task to trigger the user-defined event.

To wait for a user-defined event:

1. In the workflow, create an Event-Wait task and double-click the Event-Wait task to open it.
2. In the Events tab of the task, select User-Defined.
3. Click the Event button to open the Events Browser dialog box.
4. Select a user-defined event for the Integration Service to wait.
5. Click OK twice.

Waiting for Predefined Events

To use a predefined event, you need a shell command, script, or batch file to create an indicator file. The file must be created or sent to a directory that the Integration Service can access. The file can be any format recognized by the Integration Service operating system. You can choose to have the Integration Service delete the indicator file after it detects the file, or you can manually delete the indicator file. The Integration Service marks the status of the Event-Wait task as failed if it cannot delete the indicator file.

When you specify the indicator file in the Event-Wait task, enter the directory in which the file appears and the name of the indicator file. You must provide the absolute path for the file. If you specify the file name and not the directory, the Integration Service looks for the indicator file in the following directory:

- On Windows, the Integration Service looks for the file in the system directory. For example, on Windows 2000, the system directory is c:\winnt\system32.
- On UNIX, the Integration Service looks for the indicator file in the current working directory for the Integration Service process. On UNIX this directory is /server/bin.

You can enter the actual name of the file or use process variables to specify the location of the file. You can also use user-defined workflow and worklet variables to specify the file name and location. For example, create a workflow variable, $$MyFileWatchFile, for the indicator file name and location, and set $$MyFileWatchFile to the file name and location in the parameter file.

The Integration Service writes the time the file appears in the workflow log.

Note: Do not use a source or target file name as the indicator file name because you may accidentally delete a source or target file. Or, the Integration Service may try to delete the file before the session finishes writing to the target.

Configuring a Workflow for a Predefined Event

To use a predefined event, you need a shell command, script, or batch file to create an indicator file.

1. In the Events tab of an Event-Wait task, select Predefined.
2. Enter the path of the indicator file.
3. If you want the Integration Service to delete the indicator file after it detects the file, select the Delete Filewatch File option in the Properties tab.
4. Click OK.

Enabling Past Events

By default, the Event-Wait task waits for the Event-Raise task to trigger the event. By default, the Event-Wait task does not check if the event already occurred. You can select the Enable Past Events option so that the Integration Service verifies that the event has already occurred.

When you select Enable Past Events, the Integration Service continues executing the next tasks if the event already occurred.

Select the Enable Past Events option in the Properties tab of the Event-Wait task.

Timer Task

You can specify the period of time to wait before the Integration Service runs the next task in the workflow with the Timer task. You can choose to start the next task in the workflow at a specified time and date. You can also choose to wait a period of time after the start time of another task, workflow, or worklet before starting the next task.

The Timer task has the following types of settings:

- **Absolute time.** You specify the time that the Integration Service starts running the next task in the workflow. You may specify the date and time, or you can choose a user-defined workflow variable to specify the time.
- **Relative time.** You instruct the Integration Service to wait for a specified period of time after the Timer task, the parent workflow, or the top-level workflow starts.

For example, a workflow contains two sessions. You want the Integration Service wait 10 minutes after the first session completes before it runs the second session. Use a Timer task after the first session. In the Relative Time setting of the Timer task, specify ten minutes from the start time of the Timer task. Use a Timer task anywhere in the workflow after the Start task.

The following table describes the attributes you configure in the Timer task:

Timer Attribute	Description
Absolute Time: Specify the exact time to start	Integration Service starts the next task in the workflow at the date and time you specify.
Absolute Time: Use this workflow date-time variable to calculate the wait	Specify a user-defined date-time workflow variable. The Integration Service starts the next task in the workflow at the time you choose. The Workflow Manager verifies that the variable you specify has the Date/Time datatype. If the variable precision includes subseconds, the Integration Service ignores the subsecond portion of the time value. The Timer task fails if the date-time workflow variable evaluates to NULL.
Relative time: Start after	Specify the period of time the Integration Service waits to start executing the next task in the workflow.
Relative time: from the start time of this task	Select this option to wait a specified period of time after the start time of the Timer task to run the next task.

Timer Attribute	Description
Relative time: from the start time of the parent workflow/worklet	Select this option to wait a specified period of time after the start time of the parent workflow/worklet to run the next task.
Relative time: from the start time of the top-level workflow	Choose this option to wait a specified period of time after the start time of the top-level workflow to run the next task.

Creating a Timer Task

Create a Timer task to specify the amount of time the Integration Service waits before it starts the next task in the workflow.

1. In the Workflow Designer, click Tasks > Create.
2. Select Timer Task for the task type.
3. Double-click the Timer task to open it.
4. On the General tab, enter a name for the Timer task.
5. Click the Timer tab to specify when the Integration Service starts the next task in the workflow.
6. Specify attributes for Absolute Time or Relative Time.

Sources

Sources Overview

In the Workflow Manager, you can create sessions with the following sources:

- **Relational.** You can extract data from any relational database that the Integration Service can connect to. When extracting data from relational sources and Application sources, you must configure the database connection to the data source prior to configuring the session.
- **File.** You can create a session to extract data from a flat file, COBOL, or XML source. Use an operating system command to generate source data for a flat file or COBOL source or generate a file list.

  If you use a flat file or XML source, the Integration Service can extract data from any local directory or FTP connection for the source file. If the file source requires an FTP connection, you need to configure the FTP connection to the host machine before you create the session.
- **Heterogeneous.** You can extract data from multiple sources in the same session. You can extract from multiple relational sources, such as Oracle and Microsoft SQL Server. Or, you can extract from multiple source types, such as relational and flat file. When you configure a session with heterogeneous sources, configure each source instance separately.

Globalization Features

You can choose a code page that you want the Integration Service to use for relational sources and flat files. You specify code pages for relational sources when you configure database connections in the Workflow Manager. You can set the code page for file sources in the session properties.

Source Connections

Before you can extract data from a source, you must configure the connection properties the Integration Service uses to connect to the source file or database. You can configure source database and FTP connections in the Workflow Manager.

Allocating Buffer Memory

When the Integration Service initializes a session, it allocates blocks of memory to hold source and target data. The Integration Service allocates at least two blocks for each source and target partition. Sessions that use a large number of sources or targets might require additional memory blocks. If the Integration Service cannot allocate enough memory blocks to hold the data, it fails the session.

Partitioning Sources

You can create multiple partitions for relational, Application, and file sources. For relational or Application sources, the Integration Service creates a separate connection to the source database for each partition you set in the session properties. For file sources, you can configure the session to read the source with one thread or multiple threads.

Configuring Sources in a Session

Configure source properties for sessions in the Sources node of the Mapping tab of the session properties. When you configure source properties for a session, you define properties for each source instance in the mapping.

The Sources node lists the sources used in the session and displays their settings. To view and configure settings for a source, select the source from the list. You can configure the following settings for a source:

- Readers
- Connections
- Properties

Configuring Readers

You can click the Readers settings on the Sources node to view the reader the Integration Service uses with each source instance. The Workflow Manager specifies the necessary reader for each source instance in the Readers settings on the Sources node.

Configuring Connections

Click the Connections settings on the Sources node to define source connection information. For relational sources, choose a configured database connection in the Value column for each relational source instance. By default, the Workflow Manager displays the source type for relational sources.

For flat file and XML sources, choose one of the following source connection types in the Type column for each source instance:

- **FTP.** To read data from a flat file or XML source using FTP, you must specify an FTP connection when you configure source options. You must define the FTP connection in the Workflow Manager prior to configuring the session.
- **None.** Choose None to read from a local flat file or XML file.

Configuring Properties

Click the Properties settings in the Sources node to define source property information. The Workflow Manager displays properties, such as source file name and location for flat file, COBOL, and XML source file types. You do not need to define any properties on the Properties settings for relational sources.

Working with Relational Sources

When you configure a session to read data from a relational source, you can configure the following properties for sources:

- **Source database connection.** Select the database connection for each relational source.
- **Treat source rows as.** Define how the Integration Service treats each source row as it reads it from the source table.
- **Override SQL query**. You can override the default SQL query to extract source data.
- **Table owner name.** Define the table owner name for each relational source.
- **Source table name.** You can override the source table name for each relational source.

Selecting the Source Database Connection

Before you can run a session to read data from a source database, the Integration Service must connect to the source database. Database connections must exist in the repository to appear on the source database list. You must define them prior to configuring a session.

On the Connections settings in the Sources node, choose the database connection. You can select a connection object, use a connection variable, or use a session parameter to define the connection value in a parameter file.

Defining the Treat Source Rows As Property

When the Integration Service reads a source, it marks each row with an indicator to specify which operation to perform when the row reaches the target. You can define how the Integration Service marks each row using the Treat Source Rows As property in the General Options settings on the Properties tab.

The following table describes the options you can choose for the Treat Source Rows As property:

Treat Source Rows As Option	Description
Insert	Integration Service marks all rows to insert into the target.
Delete	Integration Service marks all rows to delete from the target.
Update	Integration Service marks all rows to update the target. You can further define the update operation in the target options.
Data Driven	Integration Service uses the Update Strategy transformations in the mapping to determine the operation on a row-by-row basis. You define the update operation in the target options. If the mapping contains an Update Strategy transformation, this option defaults to Data Driven. You can also use this option when the mapping contains Custom transformations configured to set the update strategy.

After you determine how to treat all rows in the session, you also need to set update strategy options for individual targets.

SQL Query Override

You can alter or override the default query in the mapping by entering SQL override in the Properties settings in the Sources node. You can enter any SQL statement supported by the source database.

The Workflow Manager does not validate the SQL override. The following types of errors can cause data errors and session failure:

- Fields with incompatible datatypes or unknown fields
- Typing mistakes or other errors

Overriding the SQL Query

You can override the SQL query for a relational source.

1. In the Workflow Manager, open the session properties.
2. Click the Mapping tab and open the Transformations view.
3. Click the Sources node and open the Properties settings.
4. Click the Open button in the SQL Query field to open the SQL Editor.
5. Enter the SQL override.
6. Click OK to return to the session properties.

Configuring the Table Owner Name

You can define the owner name of the source table in the session properties. For some databases such as DB2, tables can have different owners. If the database user specified in the database connection is not the owner of the source tables in a session, specify the table owner for each source instance. A session can fail if the database user is not the owner and you do not specify the table owner name.

Specify the table owner name in the Owner Name field in the Properties settings on the Mapping tab.

You can use a parameter or variable as the table owner name. Use any parameter or variable type that you can define in the parameter file. For example, you can use a session parameter, $ParamMyTableOwner, as the table owner name, and set $ParamMyTableOwner to the table owner name in the parameter file. Use a mapping parameter to include the owner name with the table name in the following types of overrides: source filter, user-defined join, query override, or pre- or post-SQL.

Overriding the Source Table Name

You can override the source table name in the session properties. Override the source table name when you use a single session to read data from different source tables. Enter a table name in the source table name, or enter a parameter or variable to define the source table name in the parameter file. You can use mapping parameters, mapping variables, session parameters, workflow variables, or worklet variables in the source table name. For example, you can use a session parameter, $ParamSrcTable, as the source table name, and set $ParamSrcTable to the source table name in the parameter file.

Note: If you override the source table name on the Properties tab of the source instance, and you override the source table name using an SQL query, the Integration Service uses the source table name defined in the SQL query.

Working with File Sources

You can create a session to extract data from flat file, COBOL, or XML sources. When you create a session to read data from a file, you can configure the following information in the session properties:

- **Source properties.** You can define source properties on the Properties settings in the Sources node, such as source file options.
- **Flat file properties.** You can edit fixed-width and delimited source file properties.
- **Line sequential buffer length.** You can change the buffer length for flat files on the Advanced settings on the Config Object tab.
- **Treat source rows as.** You can define how the Integration Service treats each source row as it reads it from the source.

Configuring Source Properties

You can define session source properties on the Properties settings on the Mapping tab.

The following table describes the properties you define for flat file source definitions:

File Source Options	Description
Input Type	Type of source input. You can choose the following types of source input: - File. For flat file, COBOL, or XML sources. - Command. For source data or a file list generated by a command. You cannot use a command to generate XML source data.
Source File Directory	Directory name of flat file source. By default, the Integration Service looks in the service process variable directory, $PMSourceFileDir, for file sources. If you specify both the directory and file name in the Source Filename field, clear this field. The Integration Service concatenates this field with the Source Filename field when it runs the session. You can also use the $InputFile*Name* session parameter to specify the file location.
Source File Name	File name, or file name and path of flat file source. Optionally, use the $InputFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Source File Directory field when it runs the session. For example, if you have "C:\data\" in the Source File Directory field, then enter "filename.dat" in the Source Filename field. When the Integration Service begins the session, it looks for "C:\data\filename.dat". By default, the Workflow Manager enters the file name configured in the source definition.
Source File Type	Indicates whether the source file contains the source data, or whether it contains a list of files with the same file properties. You can choose the following source file types: - Direct. For source files that contain the source data. - Indirect. For source files that contain a list of files. When you select Indirect, the Integration Service finds the file list and reads each listed file when it runs the session.
Command Type	Type of source data the command generates. You can choose the following command types: - Command generating data for commands that generate source data input rows. - Command generating file list for commands that generate a file list.
Command	Command used to generate the source file data.
Set File Properties link	Overrides source file properties. By default, the Workflow Manager displays file properties as configured in the source definition.
Truncate string null	Strips the first null character and all characters after the first null character from string values. Enable this option for delimited flat files that contain null characters in strings. If you do not enable this option, the PowerCenter Integration Service generates a row error for any row that contains null characters in a string. Default is disabled.

Configuring Commands for File Sources

Use a command to generate flat file source data input rows or a list of source files for a session. For UNIX, use any valid UNIX command or shell script. For Windows, use any valid DOS or batch file on Windows. You can also use service process variables, such as $PMSourceFileDir, in the command.

Generating Flat File Source Data

Use a command to generate the input rows for flat file source data. Use a command to generate or transform flat file data and send the standard output of the command to the flat file reader when the session runs. The flat file reader reads the standard output of the command as the flat file source data. Generating source data with a command eliminates the need to stage a flat file source. Use a command or script to send source data directly to the Integration Service instead of using a pre-session command to generate a flat file source.

For example, to uncompress a data file and use the uncompressed data as the source data input rows, use the following command:

```
uncompress -c $PMSourceFileDir/myCompressedFile.Z
```

The command uncompresses the file and sends the standard output of the command to the flat file reader. The flat file reader reads the standard output of the command as the flat file source data.

Generating a File List

Use a command to generate a list of source files. The flat file reader reads each file in the list when the session runs. Use a command to generate a file list when the list of source files changes often or you want to generate a file list based on specific conditions. You might want to use a command to generate a file list based on a directory listing.

For example, to use a directory listing as a file list, use the following command:

```
cd $PMSourceFileDir; ls -1 sales-records-Sep-*-2005.dat
```

The command generates a file list from the source file directory listing. When the session runs, the flat file reader reads each file as it reads the file names from the command.

To use the output of a command as a file list, select Command as the Input Type, Command generating file list as the Command Type, and enter a command for the Command property.

Configuring Fixed-Width File Properties

When you read data from a fixed-width file, you can edit file properties in the session, such as the null character or code page. You can configure fixed-width properties for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer. You cannot configure fixed-width properties for instances of reusable sessions in the Workflow Designer.

Click Set File Properties to open the Flat Files dialog box. To edit the fixed-width properties, select Fixed Width and click Advanced. The Fixed Width Properties dialog box appears. By default, the Workflow Manager displays file properties as configured in the mapping. Edit these settings to override those configured in the source definition.

The following table describes options you can define in the Fixed Width Properties dialog box for file sources:

Fixed-Width Properties Options	Description
Text/Binary	Indicates the character representing a null value in the file. This can be any valid character in the file code page, or any binary value from 0 to 255.
Repeat Null Character	If selected, the Integration Service reads repeat null characters in a single field as a single null value. If you do not select this option, the Integration Service reads a single null character at the beginning of a field as a null field. **Important:** For multibyte code pages, specify a single-byte null character if you use repeating non-binary null characters. This ensures that repeating null characters fit into the column.
Code Page	Code page of the fixed-width file. Select a code page or a variable: - Code page. Select the code page. - Use Variable. Enter a user-defined workflow or worklet variable or the session parameter $Param*Name*, and define the code page in the parameter file. Use the code page name. Default is the PowerCenter Client code page.
Number of Initial Rows to Skip	Integration Service skips the specified number of rows before reading the file. Use this to skip header rows. One row may contain multiple records. If you select the Line Sequential File Format option, the Integration Service ignores this option.
Number of Bytes to Skip Between Records	Integration Service skips the specified number of bytes between records. For example, you have an ASCII file on Windows with one record on each line, and a carriage return and line feed appear at the end of each line. If you want the Integration Service to skip these two single-byte characters, enter 2. If you have an ASCII file on UNIX with one record for each line, ending in a carriage return, skip the single character by entering 1.
Strip Trailing Blanks	If selected, the Integration Service strips trailing blanks from string values.
Line Sequential File Format	Select this option if the file uses a carriage return at the end of each record, shortening the final column.

Configuring Delimited File Properties

When you read data from a delimited file, you can edit file properties in the session, such as the delimiter or code page. You can configure delimited properties for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer. You cannot configure delimited properties for instances of reusable sessions in the Workflow Designer. Click Set File Properties to open the Flat Files dialog box.

To edit the delimited properties, select Delimited and click Advanced. The Delimited File Properties dialog box appears. By default, the Workflow Manager displays file properties as configured in the mapping. Edit these settings to override those configured in the source definition.

The following table describes options you can define in the Delimited File Properties dialog box for file sources:

Delimited File Properties Options	Description
Column Delimiters	One or more characters used to separate columns of data. Delimiters can be either printable or single-byte unprintable characters and must be different from the escape character and the quote character. You can enter a single-byte unprintable character by browsing the delimiter list in the Delimiters dialog box. You cannot select unprintable multibyte characters as delimiters. You cannot select the NULL character as the column delimiter for a flat file source. Maximum number of delimiters is 80.
Treat Consecutive Delimiters as One	By default, the Integration Service treats multiple delimiters separately. If selected, the Integration Service reads any number of consecutive delimiter characters as one. For example, a source file uses a comma as the delimiter character and contains the following record: 56, , , Jane Doe. By default, the Integration Service reads that record as four columns separated by three delimiters: 56, NULL, NULL, Jane Doe. If you select this option, the Integration Service reads the record as two columns separated by one delimiter: 56, Jane Doe.
Treat Multiple Delimiters as AND	If selected, the Integration Service treats a specified set of delimiters as one. For example, a source file contains the following record: abc~def|ghi~|~|jkl|~mno. By default, the Integration Service reads the record as nine columns separated by eight delimiters: abc, def, ghi, NULL, NULL, NULL, jkl, NULL, mno. If you select this option and specify the delimiter as (~ |), the Integration Service reads the record as three columns separated by two delimiters: abc~def|ghi, NULL, jkl|~mno.
Optional Quotes	Select No Quotes, Single Quote, or Double Quotes. If you select a quote character, the Integration Service ignores delimiter characters within the quote characters. Therefore, the Integration Service uses quote characters to escape the delimiter. For example, a source file uses a comma as a delimiter and contains the following row: `342-3849, 'Smith, Jenna', 'Rockville, MD', 6.` If you select the optional single quote character, the Integration Service ignores the commas within the quotes and reads the row as four fields. If you do not select the optional single quote, the Integration Service reads six separate fields. When the Integration Service reads two optional quote characters within a quoted string, it treats them as one quote character. For example, the Integration Service reads the following quoted string as `I'm going tomorrow`: `2353, 'I''m going tomorrow', MD` Additionally, if you select an optional quote character, the Integration Service reads a string as a quoted string if the quote character is the first character of the field. **Note:** You can improve session performance if the source file does not contain quotes or escape characters.
Code Page	Code page of the delimited file. Select a code page or a variable: - Code page. Select the code page. - Use Variable. Enter a user-defined workflow or worklet variable or the session parameter $Param*Name*, and define the code page in the parameter file. Use the code page name. Default is the PowerCenter Client code page.

Delimited File Properties Options	Description
Row Delimiter	Specify a line break character. Select from the list or enter a character. Preface an octal code with a backslash (\). To use a single character, enter the character. The Integration Service uses only the first character when the entry is not preceded by a backslash. The character must be a single-byte character, and no other character in the code page can contain that byte. Default is line-feed, \012 LF (\n).
Escape Character	Character immediately preceding a delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string. When you specify an escape character, the Integration Service reads the delimiter character as a regular character (called escaping the delimiter or quote character). **Note:** You can improve session performance for mappings containing Sequence Generator transformations if the source file does not contain quotes or escape characters.
Remove Escape Character From Data	This option is selected by default. Clear this option to include the escape character in the output string.
Number of Initial Rows to Skip	Integration Service skips the specified number of rows before reading the file. Use this to skip title or header rows in the file.

Configuring Line Sequential Buffer Length

You can configure the line buffer length for file sources. By default, the Integration Service reads a file record into a buffer that holds 1024 bytes. If the source file records are larger than 1024 bytes, increase the Line Sequential Buffer Length property in the session properties accordingly. Define the line buffer length on the Config Object tab in the session properties.

Integration Service Handling for File Sources

When you configure a session with file sources, you might take these additional features into account when creating mappings with file sources:

- Character set
- Multibyte character error handling
- Null character handling
- Row length handling for fixed-width flat files
- Numeric data handling
- Tab handling

Character Set

You can configure the Integration Service to run sessions in either ASCII or Unicode data movement mode.

The following table describes source file formats supported by each data movement path in PowerCenter:

Character Set	Unicode mode	ASCII mode
7-bit ASCII	Supported	Supported
US-EBCDIC (COBOL sources only)	Supported	Supported
8-bit ASCII	Supported	Supported
8-bit EBCDIC (COBOL sources only)	Supported	Supported
ASCII-based MBCS	Supported	Integration Service generates a warning message.
EBCDIC-based SBCS	Supported	Not supported. The Integration Service terminates the session.
EBCDIC-based MBCS	Supported	Not supported. The Integration Service terminates the session.

If you configure a session to run in ASCII data movement mode, delimiters, escape characters, and null characters must be valid in the ISO Western European Latin 1 code page. Any 8-bit characters you specified in previous versions of PowerCenter are still valid. In Unicode data movement mode, delimiters, escape characters, and null characters must be valid in the specified code page of the flat file.

Multibyte Character Error Handling

Misalignment of multibyte data in a file causes session errors. Data becomes misaligned when you place column breaks incorrectly in a file, resulting in multibyte characters that extend beyond the last byte in a column.

When you import a fixed-width flat file, you can create, move, or delete column breaks using the Flat File Wizard. Incorrect positioning of column breaks can create alignment errors when you run a session containing multibyte characters.

The Integration Service handles alignment errors in fixed-width flat files according to the following guidelines:

- **Non-line sequential file.** The Integration Service skips rows containing misaligned data and resumes reading the next row. The skipped row appears in the session log with a corresponding error message. If an alignment error occurs at the end of a row, the Integration Service skips both the current row and the next row, and writes them to the session log.
- **Line sequential file.** The Integration Service skips rows containing misaligned data and resumes reading the next row. The skipped row appears in the session log with a corresponding error message.

- **Reader error threshold.** You can configure a session to stop after a specified number of non-fatal errors. A row containing an alignment error increases the error count by 1. The session stops if the number of rows containing errors reaches the threshold set in the session properties. Errors and corresponding error messages appear in the session log file.

Fixed-width COBOL sources are always byte-oriented and can be line sequential. The Integration Service handles COBOL files according to the following guidelines:

- **Line sequential files.** The Integration Service skips rows containing misaligned data and writes the skipped rows to the session log. The session stops if the number of error rows reaches the error threshold.
- **Non-line sequential files.** The session stops at the first row containing misaligned data.

Null Character Handling

You can specify single-byte or multibyte null characters for fixed-width flat files. The Integration Service uses these characters to determine if a column is null.

The following table describes how the Integration Service uses the Null Character and Repeat Null Character properties to determine if a column is null:

Null Character	Repeat Null Character	Integration Service Behavior
Binary	Disabled	A column is null if the first byte in the column is the binary null character. The Integration Service reads the rest of the column as text data to determine the column alignment and track the shift state for shift sensitive code pages. If data in the column is misaligned, the Integration Service skips the row and writes the skipped row and a corresponding error message to the session log.
Non-binary	Disabled	A column is null if the first character in the column is the null character. The Integration Service reads the rest of the column to determine the column alignment and track the shift state for shift sensitive code pages. If data in the column is misaligned, the Integration Service skips the row and writes the skipped row and a corresponding error message to the session log.
Binary	Enabled	A column is null if it contains the specified binary null character. The next column inherits the initial shift state of the code page.
Non-binary	Enabled	A column is null if the repeating null character fits into the column with no bytes leftover. For example, a five-byte column is not null if you specify a two-byte repeating null character. In shift-sensitive code pages, shift bytes do not affect the null value of a column. A column is still null if it contains a shift byte at the beginning or end of the column. Specify a single-byte null character if you use repeating non-binary null characters. This ensures that repeating null characters fit into a column.

Row Length Handling for Fixed-Width Flat Files

For fixed-width flat files, data in a row can be shorter than the row length in the following situations:

- The file is fixed-width line-sequential with a carriage return or line feed that appears sooner than expected.
- The file is fixed-width non-line sequential, and the last line in the file is shorter than expected.

In these cases, the Integration Service reads the data but does not append any blanks to fill the remaining bytes. The Integration Service reads subsequent fields as NULL. Fields containing repeating null characters that do not fill the entire field length are not considered NULL.

Numeric Data Handling

Sometimes, file sources contain non-numeric data in numeric columns. When the Integration Service reads non-numeric data, it treats the row differently, depending on the source type. When the Integration Service reads non-numeric data from numeric columns in a flat file source or an XML source, it drops the row and writes the row to the session log. When the Integration Service reads non-numeric data for numeric columns in a COBOL source, it reads a null value for the column.

Working with XML Sources

When you create a session to read data from an XML source, you can configure source properties for that session. For example, you might want to override the source file name and location in the session properties.

The following table describes the properties you can override for XML readers in a session:

XML Source Option	Description
Treat Empty Content as Null	Treat empty XML components as Null. By default, the Integration Service does not output element tags for Null values. The Integration Service outputs tags for empty content.
Source File Directory	Location of the Source XML file. By default, the Integration Service looks in the service process variable directory, $PMSourceFileDir. You can enter the full path and file name. If you specify both the directory and file name in the Source Filename field, clear the Source File Directory. The Integration Service concatenates this field with the Source Filename field. You can also use the *$InputFileName* session parameter to specify the file directory.
Source Filename	Enter the file name or file name and path. Optionally, use the *$InputFileName* session parameter for the file name. If you specify both the directory and file name in the Source File Directory field, clear this field. The Integration Service concatenates this field with the Source File Directory field when it runs the session. For example, if you have “C:\XMLdata\” in the Source File Directory field, then enter “filename.xml” in the Source Filename field. When the Integration Service begins the session, it looks for “C:\data\filename.xml”.
Source Filetype	Use to configure multiple file sources with a file list. Choose Direct or Indirect. The option indicates whether the source file contains the source data, or whether the source file contains a list of files with the same file properties. Choose Direct if the source file contains the source data. Choose Indirect if the source file contains a list of files. When you select Indirect, the Integration Service finds the file list and reads each listed file when it runs the session.

The following table describes the properties you can override for an XML Source Qualifier in a session:

XML Source Option	Description
Validate XML Source	Provides flexibility for validating an XML source against a schema or DTD file. Select Do Not Validate to skip validation, even if the instance document has an associated DTD or schema reference. Select Validate Only if DTD is Present to validate when the XML source has a corresponding DTD or schema file. The session fails if the instance document specifies a DTD or schema and one is not present. Select Always Validate to always validate the XML file. The session fails if the DTD or schema does not exist or the data is invalid.
Partitionable	You can create multiple partitions for the source pipeline.

Server Handling for XML Sources

The Integration Service can distinguish empty values from null values in an XML source. You can choose to pass empty strings as null values by selecting the Treat Empty Content As NULL option in the Mapping tab of the session properties. By default, empty content is Not Null.

You can choose to omit fixed elements from the XML source definition. If the DTD or XML schema specifies a fixed or default value for an element, the value appears in the XML source definition.

You can define attributes as required, optional, or prohibited in an element tag. You can also specify fixed or default values for attributes. When a DTD or XML schema contains an attribute with a fixed or default value, the Integration Service passes the value into the pipeline even if the element tag in the instance document does not contain the attribute. If the attribute does not have a fixed or default value, the Integration Service passes a null value for the attribute. A parser error occurs when a required attribute is not present in an element or a prohibited attribute appears in the element tag. The Integration Service writes this error to the session log.

Using a File List

You can create a session to run multiple source files for one source instance in the mapping. You might use this feature if, for example, the organization collects data at several locations which you then want to move through the same session. When you create a mapping to use multiple source files for one source instance, the properties of all files must match the source definition.

To use multiple source files, you create a file containing the names and directories of each source file you want the Integration Service to use. This file is referred to as a file list.

When you configure the session properties, enter the file name of the file list in the Source Filename field and enter the location of the file list in the Source File Directory field. When the session starts, the Integration Service reads the file list, then locates and reads the first file source in the list. After the Integration Service reads the first file, it locates and reads the next file in the list.

The Integration Service writes the path and name of the file list to the session log. If the Integration Service encounters an error while accessing a source file, it logs the error in the session log and stops the session.

Note: When you use a file list and the session performs incremental aggregation, the Integration Service performs incremental aggregation across all listed source files.

Creating the File List

The file list contains the names of all the source files you want the Integration Service to use for the source instance in the session. Create the file list in an editor appropriate to the Integration Service platform and save it as a text file. For example, you can create a file list for an Integration Service on Windows with any text editor then save it as ASCII.

The Integration Service interprets the file list using the Integration Service code page. Map the drives on an Integration Service on Windows or mount the drives on an Integration Service on UNIX. The Integration Service skips blank lines and ignores leading blank spaces. Any characters indicating a new line, such as \n in ASCII files, must be valid in the code page of the Integration Service.

Use the following rules and guidelines when you create the file list:

- Each file in the list must use the user-defined code page configured in the source definition.
- Each file in the file list must share the same file properties as configured in the source definition or as entered for the source instance in the session property sheet.
- Enter one file name or one path and file name on a line. If you do not specify a path for a file, the Integration Service assumes the file is in the same directory as the file list.
- Each path must be local to the Integration Service node.

The following example shows a valid file list created for an Integration Service on Windows. Each of the drives listed are mapped on the Integration Service node. The western_trans.dat file is located in the same directory as the file list.

```
western_trans.dat
d:\data\eastern_trans.dat
e:\data\midwest_trans.dat
f:\data\canada_trans.dat
```

After you create the file list, place it in a directory local to the Integration Service.

Configuring a Session to Use a File List

After you create a file list for multiple source files, you can configure the session to access those files.

1. In the Workflow Manager, open the session properties.
2. Click the Mapping tab and open the Transformations view.
3. Click the Properties settings in the Sources node.
4. In the Source Filetype field, choose Indirect.
5. In the Source Filename field, replace the file name with the name of the file list.

   If necessary, also enter the path in the Source File Directory field.

   If you enter a file name in the Source Filename field, and you have specified a path in the Source File Directory field, the Integration Service looks for the named file in the listed directory.

   -or-

   If you enter a file name in the Source Filename field, and you do not specify a path in the Source File Directory field, the Integration Service looks for the named file in the directory where the Integration Service is installed on UNIX or in the system directory on Windows.
6. Click OK.

Targets

Targets Overview

In the Workflow Manager, you can create sessions with the following targets:

- **Relational.** You can load data to any relational database that the Integration Service can connect to. When loading data to relational targets, you must configure the database connection to the target before you configure the session.
- **File.** You can load data to a flat file or XML target or write data to an operating system command. For flat file or XML targets, the Integration Service can load data to any local directory or FTP connection for the target file. If the file target requires an FTP connection, you need to configure the FTP connection to the host machine before you create the session.
- **Heterogeneous.** You can output data to multiple targets in the same session. You can output to multiple relational targets, such as Oracle and Microsoft SQL Server. Or, you can output to multiple target types, such as relational and flat file.

Globalization Features

You can configure the Integration Service to run sessions in either ASCII or Unicode data movement mode.

The following table describes target character sets supported by each data movement mode in PowerCenter:

Character Set	Unicode Mode	ASCII Mode
7-bit ASCII	Supported	Supported
ASCII-based MBCS	Supported	Integration Service generates a warning message, but does not terminate the session.
UTF-8	Supported (Targets Only)	Integration Service generates a warning message, but does not terminate the session.
EBCDIC-based SBCS	Supported	Not supported. The Integration Service terminates the session.
EBCDIC-based MBCS	Supported	Not supported. The Integration Service terminates the session.

You can work with targets that use multibyte character sets with PowerCenter. You can choose a code page that you want the Integration Service to use for relational objects and flat files. You specify code pages for relational objects when you configure database connections in the Workflow Manager. The code page for a database connection used as a target must be a superset of the source code page.

When you change the database connection code page to one that is not two-way compatible with the old code page, the Workflow Manager generates a warning and invalidates all sessions that use that database connection.

Code pages you select for a file represent the code page of the *data* contained in these files. If you are working with flat files, you can also specify delimiters and null characters supported by the code page you have specified for the file.

Target code pages must be a superset of the source code page.

However, if you configure the Integration Service and Client for code page relaxation, you can select any code page supported by PowerCenter for the target database connection. When using code page relaxation, select compatible code pages for the source and target data to prevent data inconsistencies.

If the target contains multibyte character data, configure the Integration Service to run in Unicode mode. When the Integration Service runs a session in Unicode mode, it uses the database code page to translate data.

If the target contains only single-byte characters, configure the Integration Service to run in ASCII mode. When the Integration Service runs a session in ASCII mode, it does not validate code pages.

Target Connections

Before you can load data to a target, you must configure the connection properties the Integration Service uses to connect to the target file or database. You can configure target database and FTP connections in the Workflow Manager.

RELATED TOPICS:

- “Relational Database Connections” on page 127
- “FTP Connections” on page 130

Partitioning Targets

When you create multiple partitions in a session with a relational target, the Integration Service creates multiple connections to the target database to write target data concurrently. When you create multiple partitions in a session with a file target, the Integration Service creates one target file for each partition. You can configure the session properties to merge these target files.

Configuring Targets in a Session

Configure target properties for sessions in the Transformations view on Mapping tab of the session properties. Click the Targets node to view the target properties. When you configure target properties for a session, you define properties for each target instance in the mapping.

The Targets node contains the following settings where you define properties:

- Writers
- Connections
- Properties

Configuring Writers

Click the Writers settings in the Transformations view to define the writer to use with each target instance. When the mapping target is a flat file, an XML file, an SAP NetWeaver BI target, or a WebSphere MQ target, the Workflow Manager specifies the necessary writer in the session properties.

However, when the target is relational, you can change the writer type to File Writer if you plan to use an external loader.

Note: You can change the writer type for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer. You cannot change the writer type for instances of reusable sessions in the Workflow Designer.

When you override a relational target to use the file writer, the Workflow Manager changes the properties for that target instance on the Properties settings. It also changes the connection options you can define in the Connections settings.

If the target contains a column with datetime values, the Integration Service compares the date formats defined for the target column and the session. When the date formats do not match, the Integration Service uses the date format with the lesser precision. For example, a session writes to a Microsoft SQL Server target that includes a Datetime column with precision to the millisecond. The date format for the session is MM/DD/YYYY HH24:MI:SS.NS. If you override the Microsoft SQL Server target with a flat file writer, the Integration Service writes datetime values to the flat file with precision to the millisecond. If the date format for the session is MM/DD/YYYY HH24:MI:SS, the Integration Service writes datetime values to the flat file with precision to the second.

After you override a relational target to use a file writer, define the file properties for the target. Click Set File Properties and choose the target to define.

Configuring Connections

View the Connections settings on the Mapping tab to define target connection information. For relational targets, the Workflow Manager displays Relational as the target type by default. In the Value column, choose a configured database connection for each relational target instance.

For flat file and XML targets, choose one of the following target connection types in the Type column for each target instance:

- **FTP.** If you want to load data to a flat file or XML target using FTP, you must specify an FTP connection when you configure target options. FTP connections must be defined in the Workflow Manager prior to configuring sessions.
- **Loader.** Use the external loader option to improve the load speed to Oracle, DB2, Sybase IQ, or Teradata target databases.

  To use this option, you must use a mapping with a relational target definition and choose File as the writer type on the Writers settings for the relational target instance. The Integration Service uses an external loader to load target files to the Oracle, DB2, Sybase IQ, or Teradata database. You cannot choose external loader if the target is defined in the mapping as a flat file, XML, MQ, or SAP BW target.
- **Queue.** Choose Queue when you want to output to a WebSphere MQ or MSMQ message queue.
- **None.** Choose None when you want to write to a local flat file or XML file.

Configuring Properties

View the Properties settings on the Mapping tab to define target property information. The Workflow Manager displays different properties for the different target types: relational, flat file, and XML.

Performing a Test Load

You can configure the Integration Service to perform a test load. With a test load, the Integration Service reads and transforms data without writing to targets. The Integration Service reads the number you configure for the test load. The Integration Service generates all session files and performs all pre- and post-session functions, as if running the full session. To configure a session to perform a test load, enable test load and enter the number of rows to test.

The Integration Service writes data to relational targets, but rolls back the data when the session completes. For all other target types, such as flat file and SAP BW, the Integration Service does not write data to the targets.

Use the following rules and guidelines when performing a test load:

- You cannot perform a test load on sessions using XML sources.
- You can perform a test load for relational targets when you configure a session for normal mode.
- If you configure the session for bulk mode, the session fails.
- Enable a test load on the session Properties tab.

Configuring a Test Load

Configure a test load to verify that the Integration Service can process a number of rows in the mapping pipeline.

1. In the Session task, click the Properties tab.
2. In the General Options settings, click Enable Test Load.
3. Enter the number of rows to test.

Working with Relational Targets

When you configure a session to load data to a relational target, you define most properties in the Transformations view on the Mapping tab. You also define some properties on the Properties tab and the Config Object tab.

You can configure the following properties for relational targets:

- **Target database connection.** Define database connection information.
- **Target properties.** You can define target properties such as target load type, target update options, and reject options.
- **Truncate target tables.** The Integration Service can truncate target tables before loading data.
- **Deadlock retry.** You can configure the session to retry deadlocks when writing to targets or a recovery table.
- **Drop and recreate indexes.** Use pre- and post-session SQL to drop and recreate an index on a relational target table to optimize query speed.
- **Constraint-based loading.** The Integration Service can load data to targets based on primary key-foreign key constraints and active sources in the session mapping.
- **Bulk loading.** You can specify bulk mode when loading to DB2, Microsoft SQL Server, Oracle, and Sybase databases.

You can define the following properties in the session and override the properties you define in the mapping:

- **Table name prefix.** You can specify the target owner name or prefix in the session properties to override the table name prefix in the mapping.
- **Pre-session SQL.** You can create SQL commands and execute them in the target database before loading data to the target. For example, you might want to drop the index for the target table before loading data into it.
- **Post-session SQL.** You can create SQL commands and execute them in the target database after loading data to the target. For example, you might want to recreate the index for the target table after loading data into it.
- **Target table name.** You can override the target table name for each relational target.

If any target table or column name contains a database reserved word, you can create and maintain a reserved words file containing database reserved words. When the Integration Service executes SQL against the database, it places quotes around the reserved words.

When the Integration Service runs a session with at least one relational target, it performs database transactions per target connection group. For example, it commits all data to targets in a target connection group at the same time.

Target Database Connection

Before you can run a session to load data to a target database, the Integration Service must connect to the target database. Database connections must exist in the repository to appear on the target database list. You must define them prior to configuring a session.

On the Connections settings in the Targets node, choose the database connection. You can select a connection object, use a connection variable, or use a session parameter to define the connection value in a parameter file.

Target Properties

You can configure session properties for relational targets in the Transformations view on the Mapping tab, and in the General Options settings on the Properties tab. Define the properties for each target instance in the session. When you click the Transformations view on the Mapping tab, you can view and configure the settings of a specific target. Select the target under the Targets node.

The following table describes the properties available in the Properties settings on the Mapping tab of the session properties:

Target Property	Description
Target Load Type	You can choose Normal or Bulk. If you select Normal, the Integration Service loads targets normally. You can choose Bulk when you load to DB2, Sybase, Oracle, or Microsoft SQL Server. If you specify Bulk for other database types, the Integration Service reverts to a normal load. Loading in bulk mode can improve session performance, but limits the ability to recover because no database logging occurs. Choose Normal mode if the mapping contains an Update Strategy transformation. If you choose Normal and the Microsoft SQL Server target name includes spaces, configure the following connection environment SQL in the connection object: `SET QUOTED_IDENTIFIER ON`
Insert	Integration Service inserts all rows flagged for insert. Default is enabled.
Update (as Update)	Integration Service updates all rows flagged for update. Default is enabled.
Update (as Insert)	Integration Service inserts all rows flagged for update. Default is disabled.
Update (else Insert)	Integration Service updates rows flagged for update if they exist in the target, then inserts any remaining rows marked for insert. Default is disabled.
Delete	Integration Service deletes all rows flagged for delete. Default is disabled.
Truncate Table	Integration Service truncates the target before loading. Default is disabled.
Reject File Directory	Reject-file directory name. By default, the Integration Service writes all reject files to the service process variable directory, $PMBadFileDir. If you specify both the directory and file name in the Reject Filename field, clear this field. The Integration Service concatenates this field with the Reject Filename field when it runs the session. You can also use the $BadFile*Name* session parameter to specify the file directory.
Reject Filename	File name or file name and path for the reject file. By default, the Integration Service names the reject file after the target instance name: *target_name*.bad. Optionally, use the $BadFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Reject File Directory field when it runs the session. For example, if you have “C:\reject_file\” in the Reject File Directory field, and enter “filename.bad” in the Reject Filename field, the Integration Service writes rejected rows to C:\reject_file\filename.bad.

Using Session-Level Target Properties with Source Properties

You can set session-level target properties to specify how the Integration Service inserts, updates, and deletes rows. However, you can also set session-level properties for *sources*.

At the source level, you can specify whether the Integration Service inserts, updates, or deletes source rows or whether it treats rows as data driven. If you treat source rows as data driven, you must use an Update Strategy transformation to indicate how the Integration Service handles rows.

This section explains how the Integration Service writes data based on the source and target row properties. PowerCenter uses the source and target row options to provide an extra check on the session-level properties. In addition, when you use both the source and target row options, you can control inserts, updates, and deletes for the entire session or, if you use an Update Strategy transformation, based on the data.

When you set the row-handling property for a source, you can treat source rows as inserts, deletes, updates, or data driven according to the following guidelines:

- **Inserts.** If you treat source rows as inserts, select Insert for the target option. When you enable the Insert target row option, the Integration Service ignores the other target row options and treats all rows as inserts. If you disable the Insert target row option, the Integration Service rejects all rows.
- **Deletes.** If you treat source rows as deletes, select Delete for the target option. When you enable the Delete target option, the Integration Service ignores the other target-level row options and treats all rows as deletes. If you disable the Delete target option, the Integration Service rejects all rows.
- **Updates.** If you treat source rows as updates, the behavior of the Integration Service depends on the target options you select.

  The following table describes how the Integration Service loads the target when you configure the session to treat source rows as updates:

Target Option	Integration Service Behavior
Insert	If enabled, the Integration Service uses the target update option (Update as Update, Update as Insert, or Update else Insert) to update rows. If disabled, the Integration Service rejects all rows when you select Update as Insert or Update else Insert as the target-level update option.
Update as Update	Integration Service updates all rows as updates.
Update as Insert	Integration Service updates all rows as inserts. You must also select the Insert target option.
Update else Insert	Integration Service updates existing rows and inserts other rows as if marked for insert. You must also select the Insert target option.
Delete	Integration Service ignores this setting and uses the selected target update option.

  The Integration Service rejects all rows if you do not select one of the target update options.

- **Data Driven.** If you treat source rows as data driven, you use an Update Strategy transformation to specify how the Integration Service handles rows. However, the behavior of the Integration Service also depends on the target options you select.

The following table describes how the Integration Service loads the target when you configure the session to treat source rows as data driven:

Target Option	Integration Service Behavior
Insert	If enabled, the Integration Service inserts all rows flagged for insert. Enabled by default. If disabled, the Integration Service rejects the following rows: - Rows flagged for insert - Rows flagged for update if you enable Update as Insert or Update else Insert
Update as Update	Integration Service updates all rows flagged for update. Enabled by default.
Update as Insert	Integration Service inserts all rows flagged for update. Disabled by default.
Update else Insert	Integration Service updates rows flagged for update and inserts remaining rows as if marked for insert.
Delete	If enabled, the Integration Service deletes all rows flagged for delete. If disabled, the Integration Service rejects all rows flagged for delete.

The Integration Service rejects rows flagged for update if you do not select one of the target update options.

Target Table Truncation

The Integration Service can truncate target tables before running a session. You can choose to truncate tables on a target-by-target basis. If you have more than one target instance, select the truncate target table option for one target instance.

The Integration Service issues a delete or truncate command based on the target database and primary key-foreign key relationships in the session target. To optimize performance, use the truncate table command. The delete from command may impact performance.

The following table describes the commands that the Integration Service issues for each database:

Target Database	Table contains a primary key referenced by a foreign key	Table does not contain a primary key referenced by a foreign key
DB2[1]	delete from <table_name>	truncate table <table_name> immediate
Informix	delete from <table_name>	delete from <table_name>
ODBC	delete from <table_name>	delete from <table_name>
Oracle	delete from <table_name> unrecoverable	truncate table <table_name>
Microsoft SQL Server	delete from <table_name>	truncate table <table_name>[2]
Sybase 11.x	truncate table <table_name>	truncate table <table_name>

[1]. If you use a DB2 database on AS/400, the Integration Service issues a clrpfm command in both cases.
[2]. If you use the Microsoft SQL Server ODBC driver, the Integration Service issues a delete statement.

If the Integration Service issues a truncate target table command and the target table instance specifies a table name prefix, the Integration Service verifies the database user privileges for the target table by issuing a truncate command. If the database user is not specified as the target owner name or does not have the database privilege to truncate the target table, the Integration Service issues a delete command instead.

If the Integration Service issues a delete command and the database has logging enabled, the database saves all deleted records to the log for rollback. If you do not want to save deleted records for rollback, you can disable logging to improve the speed of the delete.

For all databases, if the Integration Service fails to truncate or delete any selected table because the user lacks the necessary privileges, the session fails.

If you enable truncate target tables with the following sessions, the Integration Service does not truncate target tables:

- **Incremental aggregation.** When you enable both truncate target tables and incremental aggregation in the session properties, the Workflow Manager issues a warning that you cannot enable truncate target tables and incremental aggregation in the same session.
- **Test load.** When you enable both truncate target tables and test load, the Integration Service disables the truncate table function, runs a test load session, and writes a message to the session log indicating that the truncate target tables option is turned off for the test load session.
- **Real-time.** The Integration Service does not truncate target tables when you restart a JMS or WebSphere MQ real-time session that has recovery data.

Truncating a Target Table

When you truncate target tables, you can choose to truncate tables on a target-by-target basis

1. In the Workflow Manager, open the session properties.
2. Click the Mapping tab, and then click the Transformations view.
3. Click the Targets node.
4. In the Properties settings, select Truncate Target Table Option for each target table you want the Integration Service to truncate before it runs the session.
5. Click OK.

Deadlock Retry

Select the Session Retry on Deadlock option in the session properties if you want the Integration Service to retry writes to a target database or recovery table on a deadlock. A deadlock occurs when the Integration Service attempts to take control of the same lock for a database row.

The Integration Service may encounter a deadlock under the following conditions:

- A session writes to a partitioned target.
- Two sessions write simultaneously to the same target.
- Multiple sessions simultaneously write to the recovery table, PM_RECOVERY.

Encountering deadlocks can slow session performance. To improve session performance, you can increase the number of target connection groups the Integration Service uses to write to the targets in a session. To use a different target connection group for each target in a session, use a different database connection name for each target instance. You can specify the same connection information for each connection name.

You can retry sessions on deadlock for targets configured for normal load. If you select this option and configure a target for bulk mode, the Integration Service does not retry target writes on a deadlock for that target. You can also configure the Integration Service to set the number of deadlock retries and the deadlock sleep time period.

To retry a session on deadlock, click the Properties tab in the session properties and then scroll down to the Performance settings.

Dropping and Recreating Indexes

After you insert significant amounts of data into a target, you normally need to drop and recreate indexes on that table to optimize query speed. You can drop and recreate indexes by:

- **Using pre- and post-session SQL.** The preferred method for dropping and re-creating indexes is to define an SQL statement in the Pre SQL property that drops indexes before loading data to the target. Use the Post SQL property to recreate the indexes after loading data to the target. Define the Pre SQL and Post SQL properties for relational targets in the Transformations view on the Mapping tab in the session properties.
- **Using the Designer.** The same dialog box you use to generate and execute DDL code for table creation can drop and recreate indexes. However, this process is not automatic. Every time you run a session that modifies the target table, you need to launch the Designer and use this feature.

Constraint-Based Loading

In the Workflow Manager, you can specify constraint-based loading for a session. When you select this option, the Integration Service orders the target load on a row-by-row basis. For every row generated by an active source, the Integration Service loads the corresponding transformed row first to the primary key table, then to any foreign key tables. Constraint-based loading depends on the following requirements:

- **Active source.** Related target tables must have the same active source.
- **Key relationships.** Target tables must have key relationships.
- **Target connection groups.** Targets must be in one target connection group.
- **Treat rows as insert.** Use this option when you insert into the target. You cannot use updates with constraint-based loading.

Active Source

When target tables receive rows from different active sources, the Integration Service reverts to normal loading for those tables, but loads all other targets in the session using constraint-based loading when possible. For example, a mapping contains three distinct pipelines. The first two contain a source, source qualifier, and target. Since these two targets receive data from different active sources, the Integration Service reverts to normal loading for both targets. The third pipeline contains a source, Normalizer, and two targets. Since these two targets share a single active source (the Normalizer), the Integration Service performs constraint-based loading: loading the primary key table first, then the foreign key table.

Key Relationships

When target tables have no key relationships, the Integration Service does not perform constraint-based loading. Similarly, when target tables have circular key relationships, the Integration Service reverts to a normal load. For example, you have one target containing a primary key and a foreign key related to the primary key in a second target. The second target also contains a foreign key that references the primary key in the first target. The Integration Service cannot enforce constraint-based loading for these tables. It reverts to a normal load.

Target Connection Groups

The Integration Service enforces constraint-based loading for targets in the same target connection group. If you want to specify constraint-based loading for multiple targets that receive data from the same active source, you must verify the tables are in the same target connection group. If the tables with the primary key-foreign key relationship are in different target connection groups, the Integration Service cannot enforce constraint-based loading when you run the workflow.

To verify that all targets are in the same target connection group, complete the following tasks:

- Verify all targets are in the same target load order group and receive data from the same active source.
- Use the default partition properties and do not add partitions or partition points.
- Define the same target type for all targets in the session properties.
- Define the same database connection name for all targets in the session properties.
- Choose normal mode for the target load type for all targets in the session properties.

Treat Rows as Insert

Use constraint-based loading when the session option Treat Source Rows As is set to Insert. You might get inconsistent data if you select a different Treat Source Rows As option and you configure the session for constraint-based loading.

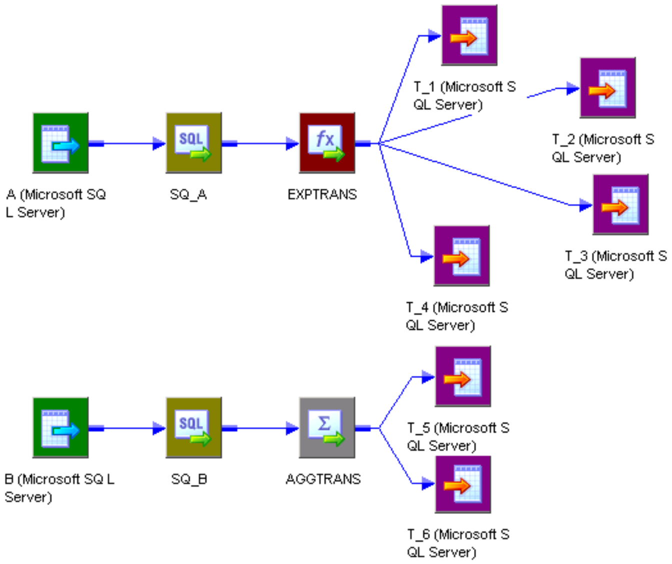
When the mapping contains Update Strategy transformations and you need to load data to a primary key table first, split the mapping using one of the following options:

- Load primary key table in one mapping and dependent tables in another mapping. Use constraint-based loading to load the primary table.
- Perform inserts in one mapping and updates in another mapping.

Constraint-based loading does not affect the target load ordering of the mapping. Target load ordering defines the order the Integration Service reads the sources in each target load order group in the mapping. A target load order group is a collection of source qualifiers, transformations, and targets linked together in a mapping. Constraint-based loading establishes the order in which the Integration Service loads individual targets within a set of targets receiving data from a single source qualifier.

Example

The following mapping is configured to perform constraint-based loading:


A (Microsoft SQ L Server)
SQ_A
EXPTRANS
T_1 (Microsoft S QL Server)
T_2 (Microsoft S QL Server)
T_3 (Microsoft S QL Server)
T_4 (Microsoft S QL Server)
B (Microsoft SQ L Server)
SQ_B
AGGTRANS
T_5 (Microsoft S QL Server)
T_6 (Microsoft S QL Server)

In the first pipeline, target T_1 has a primary key, T_2 and T_3 contain foreign keys referencing the T1 primary key. T_3 has a primary key that T_4 references as a foreign key.

Since these tables receive records from a single active source, SQ_A, the Integration Service loads rows to the target in the following order:

1. T_1
2. T_2 and T_3 (in no particular order)
3. T_4

The Integration Service loads T_1 first because it has no foreign key dependencies and contains a primary key referenced by T_2 and T_3. The Integration Service then loads T_2 and T_3, but since T_2 and T_3 have no dependencies, they are not loaded in any particular order. The Integration Service loads T_4 last, because it has a foreign key that references a primary key in T_3.

After loading the first set of targets, the Integration Service begins reading source B. If there are no key relationships between T_5 and T_6, the Integration Service reverts to a normal load for both targets.

If T_6 has a foreign key that references a primary key in T_5, since T_5 and T_6 receive data from a single active source, the Aggregator AGGTRANS, the Integration Service loads rows to the tables in the following order:

- T_5
- T_6

T_1, T_2, T_3, and T_4 are in one target connection group if you use the same database connection for each target, and you use the default partition properties. T_5 and T_6 are in another target connection group together if you use the same database connection for each target and you use the default partition properties. The Integration Service includes T_5 and T_6 in a different target connection group because they are in a different target load order group from the first four targets.

Enabling Constraint-Based Loading

When you enable constraint-based loading, the Integration Service orders the target load on a row-by-row basis.

1. In the General Options settings of the Properties tab, choose Insert for the Treat Source Rows As property.
2. Click the Config Object tab. In the Advanced settings, select Constraint Based Load Ordering.
3. Click OK.

Bulk Loading

You can enable bulk loading when you load to DB2, Sybase, Oracle, or Microsoft SQL Server.

If you enable bulk loading for other database types, the Integration Service reverts to a normal load. Bulk loading improves the performance of a session that inserts a large amount of data to the target database. Configure bulk loading on the Mapping tab.

When bulk loading, the Integration Service invokes the database bulk utility and bypasses the database log, which speeds performance. Without writing to the database log, however, the target database cannot perform rollback. As a result, you may not be able to perform recovery. Therefore, you must weigh the importance of improved session performance against the ability to recover an incomplete session.

Note: When loading to DB2, Microsoft SQL Server, and Oracle targets, you must specify a normal load for data driven sessions. When you specify bulk mode and data driven, the Integration Service reverts to normal load.

Committing Data

When bulk loading to Sybase and DB2 targets, the Integration Service ignores the commit interval you define in the session properties and commits data when the writer block is full.

When bulk loading to Microsoft SQL Server and Oracle targets, the Integration Service commits data at each commit interval. Also, Microsoft SQL Server and Oracle start a new bulk load transaction after each commit.

Tip: When bulk loading to Microsoft SQL Server or Oracle targets, define a large commit interval to reduce the number of bulk load transactions and increase performance.

Oracle Guidelines

When you enable bulk load to Oracle, the Integration Service invokes the standard Oracle client interface with the bulk routines for direct path loads.

Use the following guidelines when bulk loading to Oracle:

- Do not define CHECK constraints in the database.
- Do not define primary and foreign keys in the database. However, you can define primary and foreign keys for the target definitions in the Designer.
- To bulk load into indexed tables, choose non-parallel mode and disable the Enable Parallel Mode option.

  Note that when you disable parallel mode, you cannot load multiple target instances, partitions, or sessions into the same table.

  To bulk load in parallel mode, you must drop indexes and constraints in the target tables before running a bulk load session. After the session completes, you can rebuild them. If you use bulk loading with the session on a regular basis, use pre- and post-session SQL to drop and rebuild indexes and key constraints.

- When you use the LONG data type, verify it is the last column in the table.
- Specify the Table Name Prefix for the target when you use Oracle client 9i. If you do not specify the table name prefix, the Integration Service uses the database login as the prefix.
- For successful Oracle Real Application Cluster (RAC) failover, the Oracle target load type must be normal. If the target load type is bulk, the mapping fails.

For more information, see the Oracle documentation.

DB2 Guidelines

Use the following guidelines when bulk loading to DB2:

- You must drop indexes and constraints in the target tables before running a bulk load session. After the session completes, you can rebuild them. If you use bulk loading with the session on a regular basis, use pre- and post-session SQL to drop and rebuild indexes and key constraints.
- You cannot use source-based or user-defined commit when you run bulk load sessions on DB2.
- If you create multiple partitions for a DB2 bulk load session, you must use database partitioning for the target partition type. If you choose any other partition type, the Integration Service reverts to normal load.
- When you bulk load to DB2, the DB2 database writes non-fatal errors and warnings to a message log file in the session log directory. The message log file name is <session_log_name>.<target_instance_name>.<partition_index>.log. You can check both the message log file and the session log when you troubleshoot a DB2 bulk load session.
- If you want to bulk load flat files to DB2 for z/OS, use PowerExchange®.

For more information, see the DB2 documentation.

Table Name Prefix

The table name prefix is the owner of the target table. For some databases, such as DB2, tables can have different owners. If the database user specified in the database connection is not the owner of the target tables in a session, specify the table owner for each target instance. A session can fail if the database user is not the owner and you do not specify the table owner name.

You can specify the table owner name in the target instance or on the Mapping tab of the session properties. When you specify the table owner name in the session properties, you override table owner name in the transformation properties.

You can use a parameter or variable as the target table name prefix. Use any parameter or variable type that you can define in the parameter file. For example, you can use a session parameter, $ParamMyPrefix, as the table name prefix, and set $ParamMyPrefix to the table name prefix in the parameter file.

Note: When you specify the table owner name and you set the sqlid for a DB2 database in the connection environment SQL, the Integration Service uses table owner name in the target instance. To use the table owner name specified in the SET sqlid statement, do not enter a name in the target name prefix.

Target Table Name

You can override the target table name in the session properties. Override the target table name when you use a single session to load data to different target tables. Enter a table name in the target table name, or enter a parameter or variable to define the target table name in the parameter file. You can use mapping parameters, mapping variables, session parameters, workflow variables, or worklet variables in the target table name. For example, you can use a session parameter, $ParamTgtTable, as the target table name, and set $ParamTgtTable to the target table name in the parameter file.

Configure the target table name on the Transformation view of the Mapping tab.

Reserved Words

If any table name or column name contains a database reserved word, such as MONTH or YEAR, the session fails with database errors when the Integration Service executes SQL against the database. You can create and maintain a reserved words file, reswords.txt, in the server/bin directory. When the Integration Service initializes a session, it searches for reswords.txt. If the file exists, the Integration Service places quotes around matching reserved words when it executes SQL against the database.

Use the following rules and guidelines when working with reserved words:

- The Integration Service searches the reserved words file when it generates SQL to connect to source, target, and lookup databases.
- If you override the SQL for a source, target, or lookup, you must enclose any reserved word in quotes.
- You may need to enable some databases, such as Microsoft SQL Server and Sybase, to use SQL-92 standards regarding quoted identifiers. Use connection environment SQL to issue the command. For example, use the following command with Microsoft SQL Server:

  ```
  SET QUOTED_IDENTIFIER ON
  ```

Sample reswords.txt File

To use a reserved words file, create a file named reswords.txt and place it in the server/bin directory. Create a section for each database that you need to store reserved words for. Add reserved words used in any table or column name. You do not need to store all reserved words for a database in this file. Database names and reserved words in reswords.txt are not case sensitive.

Following is a sample reswords.txt file:

```
[Teradata]
MONTH
DATE
INTERVAL
[Oracle]
OPTION
START
[DB2]
[SQL Server]
CURRENT
[Informix]
```

```
[ODBC]
MONTH
[Sybase]
```

Teradata Array Insert

When you use ODBC to write data to a Teradata target, you can insert arrays of data into the target instead of inserting data row by row. Inserting arrays of data results in better session performance.

To insert arrays of data into a Teradata target by using ODBC, configure the OptimizeTeradataWrite custom property at the session level or at the PowerCenter Integration Service level. Set the value of the OptimizeTeradataWrite custom property to 1 to insert arrays of data into the target.

Note that the OptimizeTeradataWrite custom property is applicable only for inserting data into the target, and not for updating data in the target, deleting data from the target, or reading data from the source.

Working with Target Connection Groups

When you create a session with at least one relational target, SAP NetWeaver BI target, or dynamic MQSeries target, you need to consider target connection groups. A target connection group is a group of targets that the Integration Service uses to determine commits and loading. When the Integration Service performs a database transaction, such as a commit, it performs the transaction concurrently to all targets in a target connection group.

The Integration Service performs the following database transactions per target connection group:

- **Deadlock retry.** If the Integration Service encounters a deadlock when it writes to a target, the deadlock affects targets in the same target connection group. The Integration Service still writes to targets in other target connection groups.
- **Constraint-based loading.** The Integration Service enforces constraint-based loading for targets in a target connection group. If you want to specify constraint-based loading, you must verify the primary table and foreign table are in the same target connection group.

Targets in the same target connection group meet the following criteria:

- Belong to the same partition.
- Belong to the same target load order group and transaction control unit.
- Have the same target type in the session.
- Have the same database connection name for relational targets, and Application connection name for SAP SAP NetWeaver BI targets.
- Have the same target load type, either normal or bulk mode.

For example, suppose you create a session based on a mapping that reads data from one source and writes to two Oracle target tables. In the Workflow Manager, you do not create multiple partitions in the session. You use the same Oracle database connection for both target tables in the session properties. You specify normal mode for the target load type for both target tables in the session properties. The targets in the session belong to the same target connection group.

Suppose you create a session based on the same mapping. In the Workflow Manager, you do not create multiple partitions. However, you use one Oracle database connection name for one target, and you use a different Oracle database connection name for the other target. You specify normal mode for the target load type for both target tables. The targets in the session belong to different target connection groups.

Note: When you define the target database connections for multiple targets in a session using session parameters, the targets may or may not belong to the same target connection group. The targets belong to the same target connection group if all session parameters resolve to the same target connection name. For example, you create a session with two targets and specify the session parameter $DBConnection1 for one target, and $DBConnection2 for the other target. In the parameter file, you define $DBConnection1 as Sales1 and you define $DBConnection2 as Sales1 and run the workflow. Both targets in the session belong to the same target connection group.

Working with Active Sources

An active source is an active transformation the Integration Service uses to generate rows. An active source can be any of the following transformations:

- Aggregator
- Application Source Qualifier
- Custom, configured as an active transformation
- Joiner
- MQ Source Qualifier
- Normalizer (VSAM or pipeline)
- Rank
- Sorter
- Source Qualifier
- XML Source Qualifier
- Mapplet, if it contains any of the above transformations

Note: The Filter, Router, Transaction Control, and Update Strategy transformations are active transformations in that they can change the number of rows that pass through. However, they are not active sources in the mapping because they do not generate rows. Only transformations that can generate rows are active sources.

Active sources affect how the Integration Service processes a session when you use any of the following transformations or session properties:

- **XML targets.** The Integration Service can load data from different active sources to an XML target when each input group receives data from one active source.
- **Transaction generators.** Transaction generators, such as Transaction Control transformations, become ineffective for downstream transformations or targets if you put a transaction control point after it. Transaction control points are transaction generators and active sources that generate commits.
- **Mapplets.** An Input transformation must receive data from a single active source.
- **Source-based commit.** Some active sources generate commits. When you run a source-based commit session, the Integration Service generates a commit from these active sources at every commit interval.

- **Constraint-based loading.** To use constraint-based loading, you must connect all related targets to the same active source. The Integration Service orders the target load on a row-by-row basis based on rows generated by an active source.
- **Row error logging.** If an error occurs downstream from an active source that is not a source qualifier, the Integration Service cannot identify the source row information for the logged error row.

Working with File Targets

You can output data to a flat file in either of the following ways:

- **Use a flat file target definition.** Create a mapping with a flat file target definition. Create a session using the flat file target definition. When the Integration Service runs the session, it creates the target flat file or generates the target data based on the connected ports in the mapping and on the flat file target definition. The Integration Service does not write data in unconnected ports to a fixed-width flat file target.
- **Use a relational target definition.** Use a relational definition to write to a flat file when you want to use an external loader to load the target. Create a mapping with a relational target definition. Create a session using the relational target definition. Configure the session to output to a flat file by specifying the File Writer in the Writers settings on the Mapping tab.

You can configure the following properties for flat file targets:

- **Target properties.** You can define target properties such as partitioning options, merge options, output file options, reject options, and command options.
- **Flat file properties.** You can choose to create delimited or fixed-width files, and define their properties.

Configuring Target Properties

You can configure session properties for flat file targets in the Properties settings on the Mapping tab, and in the General Options settings on the Properties tab.

Define the properties for each target instance in the session.

The following table describes the properties you define on the Mapping tab for flat file target definitions:

Target Properties	Description
Merge Type	Type of merge the Integration Service performs on the data for partitioned targets.
Merge File Directory	Name of the merge file directory. By default, the Integration Service writes the merge file in the service process variable directory, $PMTargetFileDir. If you enter a full directory and file name in the Merge File Name field, clear this field.
Merge File Name	Name of the merge file. Default is *target_name*.out. This property is required if you select a merge type.

Target Properties	Description
Append if Exists	Appends the output data to the target files and reject files for each partition. Appends output data to the merge file if you merge the target files. You cannot use this option for target files that are non-disk files, such as FTP target files. If you do not select this option, the Integration Service truncates each target file before writing the output data to the target file. If the file does not exist, the Integration Service creates it.
Create Directory if Not Exists	Creates the target directory if it does not exist.
Header Options	Create a header row in the file target. You can choose the following options: - No Header. Do not create a header row in the flat file target. - Output Field Names. Create a header row in the file target with the output port names. - Use header command output. Use the command in the Header Command field to generate a header row. For example, you can use a command to add the date to a header row for the file target. Default is No Header.
Header Command	Command used to generate the header row in the file target.
Footer Command	Command used to generate a footer row in the file target.
Output Type	Type of target for the session. Select File to write the target data to a file target. Select Command to output data to a command. You cannot select Command for FTP or Queue target connections.
Merge Command	Command used to process the output data from all partitioned targets.
Output File Directory	Name of output directory for a flat file target. By default, the Integration Service writes output files in the service process variable directory, $PMTargetFileDir. If you specify both the directory and file name in the Output Filename field, clear this field. The Integration Service concatenates this field with the Output Filename field when it runs the session. You can also use the $OutputFile*Name* session parameter to specify the file directory.
Output File Name	File name, or file name and path of the flat file target. Optionally, use the $OutputFile*Name* session parameter for the file name. By default, the Workflow Manager names the target file based on the target definition used in the mapping: *target_name*.out. The Integration Service concatenates this field with the Output File Directory field when it runs the session. If the target definition contains a slash character, the Workflow Manager replaces the slash character with an underscore. When you use an external loader to load to an Oracle database, you must specify a file extension. If you do not specify a file extension, the Oracle loader cannot find the flat file and the Integration Service fails the session. **Note:** If you specify an absolute path file name when using FTP, the Integration Service ignores the Default Remote Directory specified in the FTP connection. When you specify an absolute path file name, do not use single or double quotes.
Reject File Directory	Name of the directory for the reject file. By default, the Integration Service writes all reject files to the service process variable directory, $PMBadFileDir. If you specify both the directory and file name in the Reject File Name field, clear this field. The Integration Service concatenates this field with the Reject File Name field when it runs the session. You can also use the $BadFile*Name* session parameter to specify the file directory.

Target Properties	Description
Reject File Name	File name, or file name and path of the reject file. By default, the Integration Service names the reject file after the target instance name: *target_name*.bad. Optionally use the $BadFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Reject File Directory field when it runs the session. For example, if you have "`C:\reject_file\`" in the Reject File Directory field, and enter "filename.bad" in the Reject Filename field, the Integration Service writes rejected rows to `C:\reject_file\filename.bad`.
Command	Command used to process the target data.
Set File Properties link	Defines flat file properties. Set the file properties when you output to a flat file using a relational target definition in the mapping.

Configuring Commands for File Targets

Use a command to process target data for a flat file target. Use any valid UNIX command or shell script on UNIX. Use any valid DOS or batch file on Windows. The flat file writer sends the data to the command instead of a flat file target.

Use a command to perform additional processing of flat file target data. For example, use a command to sort target data or compress target data. You can increase session performance by pushing transformation tasks to the command instead of the Integration Service.

To send the target data to a command, select Command for the output type and enter a command for the Command property.

For example, to generate a compressed file from the target data, use the following command:

```
compress -c - > $PMTargetFileDir/myCompressedFile.Z
```

The Integration Service sends the output data to the command, and the command generates a compressed file that contains the target data.

Note: You can also use service process variables, such as $PMTargetFileDir, in the command.

Configuring Fixed-Width Properties

When you write data to a fixed-width file, you can edit file properties in the session properties, such as the null character or code page. You can configure fixed-width properties for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer. You cannot configure fixed-width properties for instances of reusable sessions in the Workflow Designer.

In the **Transformations** view on the **Mapping** tab, click the Targets node and then click **Set File Properties** to open the Flat Files dialog box.

To edit the fixed-width properties, select **Fixed Width** and click **Advanced**.

The following table describes the options you define in the **Fixed Width Properties** dialog box:

Fixed-Width Properties Options	Description
Null Character	Optional. Character that the PowerCenter Integration Service substitutes for null values when it reads null values from a database or a flat file. You can enter any valid character in the file code page.
Repeat Null Character	Optional. Fills null value fields with the character specified in the Null Character option. If you do not select this option, then the PowerCenter Integration Service substitutes each null value with one null character.
Code Page	Optional. Code page of the fixed-width file. Select a code page or a variable: - Code page. Select the code page. - Use Variable. Enter a user-defined workflow or worklet variable or the session parameter $Param*Name*, and define the code page in the parameter file. Use the code page name. Default is the PowerCenter Client code page.

Configuring Delimited Properties

When you write data to a delimited file, you can edit file properties in the session properties, such as the delimiter or code page. You can configure delimited properties for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer. You cannot configure delimited properties for instances of reusable sessions in the Workflow Designer.

In the Transformations view on the Mapping tab, click the Targets node and then click Set File Properties to open the Flat Files dialog box. To edit the delimited properties, select Delimited and click Advanced.

The following table describes the options you can define in the Delimited File Properties dialog box:

Edit Delimiter Options	Description
Delimiters	Character used to separate columns of data. Delimiters can be either printable or single-byte unprintable characters, and must be different from the escape character and the quote character (if selected). To enter a single-byte unprintable character, click the Browse button to the right of this field. In the Delimiters dialog box, select an unprintable character from the Insert Delimiter list and click Add. You cannot select unprintable multibyte characters as delimiters.
Optional Quotes	Select None, Single, or Double. If you select a quote character, the Integration Service does not treat delimiter characters within the quote characters as a delimiter. For example, suppose an output file uses a comma as a delimiter and the Integration Service receives the following row: 342-3849, 'Smith, Jenna', 'Rockville, MD', 6. If you select the optional single quote character, the Integration Service ignores the commas within the quotes and writes the row as four fields. If you do not select the optional single quote, the Integration Service writes six separate fields.
Code Page	Code page of the delimited file. Select a code page or a variable: - Code page. Select the code page. - Use Variable. Enter a user-defined workflow or worklet variable or the session parameter $Param*Name*, and define the code page in the parameter file. Use the code page name. Default is the PowerCenter Client code page.

Integration Service Handling for File Targets

When you configure a session to write to file targets, you must correctly configure the flat file target definitions and the relational target definitions. The Integration Service loads data to flat files based on the following criteria:

- **Write to fixed-width flat files from relational target definitions.** The Integration Service adds spaces to target columns based on transformation datatype.
- **Write to fixed-width flat files from flat file target definitions.** You must configure the precision and field width for flat file target definitions to accommodate the total length of the target field.
- **Generate flat file targets by transaction.** You can configure the file target to generate a separate output file for each transaction.
- **Write empty fields for unconnected ports in fixed-width file definitions.** You can configure the mapping so that the Integration Service writes empty fields for unconnected ports in a fixed-width flat file target definition.
- **Write multibyte data to fixed-width files.** You must configure the precision of string columns to accommodate character data. When writing shift-sensitive data to a fixed-width flat file target, the Integration Service adds shift characters and spaces to meet file requirements.
- **Null characters in fixed-width files.** The Integration Service writes repeating or non-repeating null characters to fixed-width target file columns differently depending on whether the characters are single-byte or multibyte.
- **Character set.** You can write ASCII or Unicode data to a flat file target.

- **Write metadata to flat file targets.** You can configure the Integration Service to write the column header information when you write to flat file targets.

Writing to Fixed-Width Flat Files with Relational Target Definitions

When you want to output to a fixed-width file based on a relational target definition in the mapping, consider how the Integration Service handles spacing in the target file.

When the Integration Service writes to a fixed-width flat file based on a relational target definition in the mapping, it adds spaces to columns based on the transformation datatype connected to the target. This allows the Integration Service to write optional symbols necessary for the datatype, such as a negative sign or decimal point, without sending the row to the reject file.

For example, you connect a transformation Integer(10) port to a Number(10) column in a relational target definition. In the session properties, you override the relational target definition to use the File Writer and you specify to output a fixed-width flat file. In the target flat file, the Integration Service appends an additional byte to the Number(10) column to allow for negative signs that might be associated with Integer data.

The following table describes the number of bytes the Integration Service adds to the target column and optional characters it uses for each datatype:

Datatype Connected to Fixed-Width Flat File Target Column	Bytes Added by Integration Service	Optional Characters for the Datatype
Decimal	2	- Negative sign (-) for the mantissa. - Decimal point (.).
Double	7	- Negative sign for the mantissa. - Decimal point. - Negative sign, e, and three digits for the exponent, for example, -4.2-e123.
Float	7	- Negative sign for the mantissa. - Decimal point. - Negative sign, e, and three digits for the exponent.
Integer	1	- Negative sign for the mantissa.
Money	2	- Negative sign for the mantissa. - Decimal point.
Numeric	2	- Negative sign for the mantissa. - Decimal point.
Real	7	- Negative sign for the mantissa. - Decimal point. - Negative sign, e, and three digits for the exponent.

Writing to Fixed-Width Files with Flat File Target Definitions

When you want to output to a fixed-width flat file based on a flat file target definition, you must configure precision and field width for the target field to accommodate the total length of the target field. If the data for a target field is too long for the total length of the field, the Integration Service performs one of the following actions:

- Truncates the row for string columns
- Writes the row to the reject file for numeric and datetime columns

Note: When the Integration Service writes a row to the reject file, it writes a message in the session log.

When a session writes to a fixed-width flat file based on a fixed-width flat file target definition in the mapping, the Integration Service defines the total length of a field by the precision or field width defined in the target.

Fixed-width files are byte-oriented, which means the total length of a field is measured in bytes.

The following table describes how the Integration Service measures the total field length for fields in a fixed-width flat file target definition:

Datatype	Target Field Property That Determines Total Field Length
Number	Field width
String	Precision
Datetime	Field width

The following table lists the characters you must accommodate when you configure the precision or field width for flat file target definitions to accommodate the total length of the target field:

Datatype	Characters to Accommodate
Number	- Decimal separator. - Thousands separators. - Negative sign (-) for the mantissa.
String	- Multibyte data. - Shift-in and shift-out characters.
Datetime	- Date and time separators, such as slashes (/), dashes (-), and colons (:). - For example, the format `MM/DD/YYYY HH24:MI:SS.US` has a total length of 26 bytes.

When you edit the flat file target definition in the mapping, define the precision or field width great enough to accommodate both the target data and the characters in the preceding table.

For example, suppose you have a mapping with a fixed-width flat file target definition. The target definition contains a number column with a precision of 10 and a scale of 2. You use a comma as the decimal separator and a period as the thousands separator. You know some rows of data might have a negative value. Based on this information, you know the longest possible number is formatted with the following format:

```
-NN.NNN.NNN,NN
```

Open the flat file target definition in the mapping and define the field width for this number column as a minimum of 14 bytes.

Generating Flat File Targets By Transaction

You can generate a separate output file each time the Integration Service starts a new transaction. You can dynamically name each target flat file. To generate a separate output file for each transaction, add a FileName port to the flat file target definition. When you connect the FileName port in the mapping, the Integration Service creates a separate target file at each commit point. The Integration Service uses the FileName port value from the first row in each transaction to name the output file.

Writing Empty Fields for Unconnected Ports in Fixed-Width File Definitions

The Integration Service does not write data in unconnected ports to fixed-width files. For example, a fixed-width flat file target definition contains the following ports:

- EmployeeID
- EmployeeName
- Street
- City
- State

In the mapping, you connect only the EmployeeID and EmployeeName ports in the flat file target definition. You configure the flat file target definition to create a header row with the output port names. The Integration Service generates an output file with the following rows:

EmployeeID	EmployeeName
2367	John Baer
2875	Bobbi Apperley

If you want the Integration Service to write empty fields for the unconnected ports, create output ports in an upstream transformation that do not contain data. Then connect these ports containing null values to the fixed-width flat file target definition. For example, you connect the ports containing null values to the Street, City, and State ports in the flat file target definition. The Integration Service generates an output file with the following rows:

EmployeeID	EmployeeName	Street	City	State
2367	John Baer	-	-	-
2875	Bobbi Apperley	-	-	-

Writing Multibyte Data to Fixed-Width Flat Files

If you plan to load multibyte data into a fixed-width flat file, configure the precision to accommodate the multibyte data. Fixed-width files are byte-oriented, not character-oriented. So, when you configure the precision for a fixed-width target, you need to consider the number of bytes you load into the target, rather than the number of characters.

For string columns, the Integration Service truncates the data if the precision is not large enough to accommodate the multibyte data.

You might work with the following types of multibyte data:

- **Non shift-sensitive multibyte data.** The file contains all multibyte data. Configure the precision in the target definition to allow for the additional bytes.

  For example, you know that the target data contains four double-byte characters, so you define the target definition with a precision of 8 bytes.

  If you configure the target definition with a precision of 4, the Integration Service truncates the data before writing to the target.
- **Shift-sensitive multibyte data.** The file contains single-byte and multibyte data. When writing to a shift-sensitive flat file target, the Integration Service adds shift characters and spaces to meet file requirements. You must configure the precision in the target definition to allow for the additional bytes *and* the shift characters.

Note: Delimited files are character-oriented, and you do not need to allow for additional precision for multibyte data.

Writing Shift-Sensitive Multibyte Data

When writing to a shift-sensitive flat file target, the Integration Service adds shift characters and spaces if the data going into the target does not meet file requirements. You need to allow at least two extra bytes in each data column containing multibyte data so the output data precision matches the byte width of the target column.

The Integration Service writes shift characters and spaces in the following ways:

- If a column begins or ends with a double-byte character, the Integration Service adds shift characters so the column begins and ends with a single-byte shift character.
- If the data is shorter than the column width, the Integration Service pads the rest of the column with spaces.
- If the data is longer than the column width, the Integration Service truncates the data so the column ends with a single-byte shift character.

To illustrate how the Integration Service handles a fixed-width file containing shift-sensitive data, say you want to output the following data to the target:

SourceCol1	**SourceCol2**
AAAA	aaaa

A is a double-byte character, a is a single-byte character.

The first target column contains eight bytes and the second target column contains four bytes.

The Integration Service must add shift characters to handle shift-sensitive data. Since the first target column can handle eight bytes, the Integration Service truncates the data before it can add the shift characters.

TargetCol1	**TargetCol2**
-oAAA-i	aaaa

The following table describes the notation used in this example:

Notation	Description
A	Double-byte character
-o	Shift-out character
-i	Shift-in character

For the first target column, the Integration Service writes three of the double-byte characters to the target. It cannot write any additional double-byte characters to the output column because the column must end in a single-byte character. If you add two more bytes to the first target column definition, then the Integration Service can add shift characters and write all the data without truncation.

For the second target column, the Integration Service writes all four single-byte characters to the target. It does not add write shift characters to the column because the column begins and ends with single-byte characters.

Null Characters in Fixed-Width Files

You can specify any valid single-byte or multibyte character as a null character for a fixed-width target. You can also use a space as a null character.

The null character can be repeating or non-repeating. If the null character is repeating, the Integration Service writes as many null characters as possible into a target column. If you specify a multibyte null character and there are extra bytes left after writing null characters, the Integration Service pads the column with single-byte spaces. If a column is smaller than the multibyte character specified as the null character, the session fails at initialization.

Character Set

You can configure the Integration Service to run sessions with flat file targets in either ASCII or Unicode data movement mode.

If you configure a session with a flat file target to run in Unicode data movement mode, the target file code page must be a superset of the source code page. Delimiters, escape, and null characters must be valid in the specified code page of the flat file.

If you configure a session to run in ASCII data movement mode, delimiters, escape, and null characters must be valid in the ISO Western European Latin1 code page. Any 8-bit character you specified in previous versions of PowerCenter is still valid.

Writing Metadata to Flat File Targets

When you write to flat file targets, you can configure the Integration Service to write the column header information. When you enable the Output Metadata For Flat File Target option, the Integration Service writes column headers to flat file targets. It writes the target definition port names to the flat file target in the first line, starting with the # symbol. By default, this option is disabled.

When writing to fixed-width files, the Integration Service truncates the target definition port name if it is longer than the column width.

For example, you have a flat file target definition with the following structure:

Port Name	**Datatype**
ITEM_ID	number
ITEM_NAME	string
PRICE	number

The column width for ITEM_ID is six. When you enable the Output Metadata For Flat File Target option, the Integration Service writes the following text to a flat file:

```
#ITEM_ITEM_NAME      PRICE
100001Screwdriver    9.50
100002Hammer         12.90
100003Small nails    3.00
```

Working with XML Targets in a Session

When you configure a session to load data to an XML target, you define writer properties on the Mapping tab of the session properties.

The following table describes the properties you define in the XML Writer:

XML Targets Options	Description
Output File Directory	Enter the directory name in this field. By default, the Integration Service writes output files in the service process variable directory, $PMTargetFileDir. You can enter the full path and file name. If you specify both the directory and file name in the Output Filename field, clear this field. The Integration Service concatenates this field with the Output Filename field when it runs the session. You can also use the $OutputFileName session parameter to specify the file directory.
Output Filename	Enter the file name or file name and path. By default, the Workflow Manager names the target file based on the target definition used in the mapping: *target_name*.xml. If the target definition contains a slash character, the Workflow Manager replaces the slash character with an underscore. Enter the file name, or file name and path. Optionally, use the $OutputFileName session parameter for the file name. If you specify both the directory and file name in the Output File Directory field, clear this field. The Integration Service concatenates this field with the Output File Directory field when it runs the session. If you specify an absolute path file name when using FTP, the Integration Service ignores the Default Remote Directory specified in the FTP connection. When you specify an absolute path file name, do not use single or double quotes.
Validate Target	Validates simple data types. The Integration Service does not validate the target XML structure against a schema.
Format Output	Format the XML target file so the XML elements and attributes indent. If you do not select Format Output, each line of the XML file starts in the same position.

XML Targets Options	Description
XML Datetime Format	Select local time, local time with time zone, or UTC. Local time with time zone is the difference in hours between the server time zone and Greenwich Mean Time. UTC is Greenwich Mean Time.
Null Content Representation	Choose how to represent null content in the target. Default is No Tag.
Empty String Content Representation	Choose how to represent empty string content in the target. Default is Tag with Empty Content.
Null Attribute Representation	Choose how to represent null attributes. Default is No Attribute.
Empty String Attribute Representation	Choose how to represent empty string attributes in the target. Default is Attribute Name with Empty String.

Integration Service Handling for XML Targets

You can configure some of the settings the Integration Service uses when it loads data to an XML target:

- **Character set**. Configure the Integration Service to run sessions with XML targets in either ASCII or Unicode data movement mode.
- **Null and empty string**. Choose how the Integration Service handles null data or empty strings when it writes data to an XML target.
- **Handling duplicate group rows**. Choose how the Integration Service handles duplicate rows of data.
- **DTD and schema referenc**e. Define a DTD or schema file name for the target XML file.
- **Flushing XML on commits**. Configure the Integration Service to periodically flush data to the target.
- **XML caching properties**. Define a cache directory for an XML target.
- **Session logs for XML targets**. View session logs for an XML session.
- **Multiple XML output.** Configure the Integration Service to output a new XML document when the data in the root changes.
- **Partitioning the XML Generator.** When you generate XML in multiple partitions, you always generate separate documents for each partition.
- **Generating XML files with no data.** Configure the WriteNullXMLFile custom property to skip creating an XML file when the XML Generator transformation receives no data.

Character Set

You can configure the Integration Service to run sessions with XML targets in either ASCII or Unicode data movement mode. XML files contain an encoding declaration that indicates the code page used in the file. The most commonly used code pages are UTF-8 and UTF-16. PowerCenter supports UTF-8 code pages for XML targets only. Use the same set of code pages for XML files as for relational databases and other files.

For XML targets, PowerCenter uses the code page declared in the XML file. When you run the Integration Service in Unicode data movement mode, the XML target code page must be a superset of the Integration Service code page and the source code page.

Special Characters

The Integration Service adds escape characters to the following special characters in XML targets:

```
< & > ”
```

Null and Empty Strings

You can choose how the Integration Service handles null data or empty strings when it writes elements and attributes to an XML target file. By default, the Integration Service does not output element tags or attribute names for null values. The Integration Service outputs tags and attribute names with no content for empty strings.

To change these defaults, you can change the Null Content Representation and Empty String Content Representation XML target properties. For attributes, change Null Attribute Representation and the Empty String Attribute Representation properties.

Choose one of the following values for each property:

Properties	Property Value	Integration Service Behavior
Null Content or Empty String Content	- No Tag - Tag with Empty Content	- Does not output a tag. - Outputs the XML tag with no content.
Null Attribute or Empty String Attribute	- No Attribute - Attribute Name with Empty String	- Does not output the attribute. - Outputs the attribute name with no content.

You can specify fixed or default values for elements and attributes. When an element in an XML schema or a DTD has a default value, the Integration Service inserts the value instead of writing empty content. When an element has a fixed value in the schema, the value is always inserted in the XML file. If the XML schema or DTD does not specify a value for an attribute and the attribute has a null value, the Integration Service omits the attribute.

If a required attribute does not have a fixed value, the attribute must be a projected field. The Integration Service does not output invalid attributes to a target. An error occurs when a prohibited attribute appears in an element tag. An error also occurs if a required attribute is not present in an element tag. The Integration Service writes these errors to the session log or the error log when you enable row error logging.

The following table describes the format of XML file elements and attributes that contain null values or empty strings:

Type of Output	Type of Data	Target File
Element	Null	<elem></elem>
-	Empty string	<elem></elem>

Type of Output	Type of Data	Target File
Attribute	Null	<elem>...</elem>
-	Empty string	<elem attrib=“”>...</elem>

Handling Duplicate Group Rows

Sometimes duplicate rows occur in source data. The Integration Service can pass one of these rows to an XML target. You can configure duplicate row handling in the XML target session properties. You can also configure the Integration Service to write warning messages in the session log when duplicate rows occur.

The Integration Service does not write duplicate rows to the reject file. The Integration Service writes duplicate rows to the session log. You can skip writing warning messages in the session log for the duplicate rows. Disable the XMLWarnDupRows Integration Service option in the Informatica Administrator.

The Integration Service handles duplicate rows passed to the XML target root group differently than it handles rows passed to other XML target groups:

- For the XML target root group, the Integration Service always passes the first row to the target. When the Integration Service encounters duplicate rows, it increases the number of rejected rows in the session load summary.
- For any XML target group other than the root group, you can configure duplicate group row handling in the XML target definition in the Mapping Designer.
- If you choose to warn about duplicate rows, the Integration Service writes all duplicate rows for the root group to the session log. Otherwise, the Integration Service drops the rows without logging any error messages.

You can select which row the Integration Service passes to the XML target:

- **First row.** The Integration Service passes the first row to the target. When the Integration Service encounters other rows with the same primary key, the Integration Service increases the number of rejected rows in the session load summary.
- **Last row.** The Integration Service passes the last duplicate row to the target. You can configure the Integration Service to write the duplicate XML rows to the session log by setting the Warn About Duplicate XML Rows option.

  For example, the Integration Service encounters five duplicate rows. If you configure the Integration Service to write the duplicate XML rows to the session log, the Integration Service passes the fifth row to the XML target and writes the first four duplicate rows to the session log. Otherwise, the Integration Service passes the fifth row to the XML target but does not write anything to the session log.

- **Error.** The Integration Service passes the first row to the target. When the Integration Service encounters a duplicate row, it increases the number of rejected rows in the session load summary and increments the error count.

  When the Integration Service reaches the error threshold, the session fails and the Integration Service does not write any rows to the XML target.

  The Integration Service sets an error threshold for each XML group.

DTD and Schema Reference

When you edit the XML target in the Target Designer, you can also specify a DTD or schema file name for the target XML file. The Integration Service adds a document type declaration or schema reference to the target XML file and inserts the name of the file you specify. For example, if you have a target XML file with the root element *TargetRoot* and you set the DTD Reference option to *TargetDoc.dtd*, the Integration Service adds the following document type declaration after the XML declaration:

```
<!DOCTYPE TargetRoot SYSTEM "TargetDoc.dtd">
```

The Integration Service does not check that the file you specify exists or that the file is valid. The Integration Service does not validate the target XML file against the DTD or schema file you specify.

Note: An XML instance document must refer to the full relative path of a schema if a midstream XML transformation is processing the file. Otherwise, the full path is not required.

Flushing XML on Commits

When you process an XML file or stream, the XML parser parses the entire XML file and writes target XML data at end of file. Use the On Commit attribute to periodically flush the data to the target before reaching end of file. You can flush data periodically into one target XML document, or you can generate multiple XML documents.

You might want to flush XML data in the following situations:

- **Large XML files.** If you are processing a large XML file of several gigabytes, the Integration Service may have reduced performance. You can set the On Commit attribute to Append to Doc. This flushes XML data periodically to the target document.
- **Real-time processing**. If you process real-time data that requires commits at specific times, use Append to Doc.

You can set the On Commit attribute to one of the following values:

- **Ignore commit.** Generate and write to the XML document at end of file.
- **Append to document.** Write to the same XML document at the end of each commit. The XML document closes at end of file. This option is not available for XML Generator transformations.
- **Create new document.** Create and write to a new document at each commit. You create multiple XML documents.

You can flush data if all groups in the XML target are connected to the same single commit or transaction point. The transformation at the commit point generates denormalized output. The denormalized output contains repeating primary key values for all but the lowest level node in the XML schema. The Integration Service extracts rows from this output for each group in the XML target.

You must have only one child group for the root group in the XML target.

Ignoring Commit

You can choose to generate the XML document after the session has read all the source records. This option causes the Integration Service to store all of the XML data in cache during a session. Use this option when you are not processing a lot of data.

Appending to Document on Commit

When you append data to an XML document, use a source-based or user-defined commit in the session. Use a single point in the mapping to generate transactions. All the projected groups of an XML target must belong to the same transaction control unit.

For sessions using source-based commits, the single transaction point might be a source or nearest active source to the XML target, such as the last active transformation before the target. For sessions with user-defined commits, the transaction point is a transaction generating transformation.

Creating XML Documents on Commit

You can choose to generate a separate XML document for each commit. To generate multiple XML output documents, set On Commit to Create New Document. To define the commit, you can turn on source-based commit in the session, or you can generate the commit from a transaction generating transformation in the mapping.

Warning: When you create new a document on commit, you need to provide a unique file name for each document. Otherwise, the Integration Service overwrites the document it created from the previous commit.

XML Caching Properties

The Integration Service uses a data cache to store XML row data while it generates an XML document. The cache size is the sum of all the groups in the XML target instance. The cache includes a primary key and a foreign key index cache for each XML group and one data cache for all groups.

You can configure the Integration Service to automatically determine the XML cache size, or you can configure the cache size. When the memory requirements exceed the cache size, the Integration Service pages data to index and data files in the cache directory. When the session completes, the Integration Service releases cache memory and deletes the cache files.

You can specify the cache directory and cache size for the XML target. The default cache directory is $PMCacheDir, which is a service process variable that represents the directory where the Integration Service stores cache files by default.

Session Logs for XML Targets

When you run a session with an XML target, the Integration Service writes the target name and group name into the session log. The session log lists the target and group names in the following format:

```
Target Name : : Group Name.
```

For example, the following session log entry contains target EMP_SALARY and group DEPARTMENT:

```
WRITER_1_1_1> WRT_8167 Start loading table [EMP_SALARY::DEPARTMENT] at: Wed Nov 05
08:01:35 2003
```

Multiple XML Document Output

The Integration Service generates a new XML document for each distinct primary key value in the root group of the target. To create separate XML files, you must pass data to the root node primary key. When the value of the key changes, the Integration Service creates a new target file. The Integration Service creates an .lst file that contains the file name and absolute path to each XML file it creates in the session.

The Integration Service creates multiple XML files when the root group has more than one distinct primary key value. If the Integration Service receives multiple rows with the same primary key value, the Integration Service chooses the first or last row based on the way you configure duplicate row handling.

If you pass data to a column in the root group, but you do not pass data to the primary key, the Integration Service does not generate a new XML document. The Integration Service writes a warning message to the session log indicating that the primary key for the root group is not projected, and the Integration Service is generating one document.

Example

The following example includes a mapping that contains a flat file source of country names, regions, and revenue dollars per region. The target is an XML file. The root view contains the primary key, XPK_COL_0, which is a string.

Each time the Integration Service passes a new country name to the root view the Integration Service generates a new target file. Each target XML file contains country name, region, and revenue data for one country.

The Integration Service passes the following rows to the XML target:

```
Country,Region,Revenue
USA,region1,1000
Canada,region1,100
USA,region2,200
USA,region3,300
USA,region4,400
France,region1,10
France,region2,20
France,region3,30
France,region4,40
```

The Integration Service builds the XML files in cache. The Integration Service creates one XML file for USA, one file for Canada, and one file for France. The Integration Service creates a file list that contains the file name and absolute path of each target XML file.

If you specify “revenue_file.xml” as the output file name in the session properties, the session produces the following files:

- **revenue_file.xml.** Contains the Canada rows.
- **revenue_file.1.xml.** Contains the France rows.
- **revenue_file.2.xml.** Contains the USA rows.
- **revenue_file.xml.lst.** Contains a list of each XML file the session created.

If the data has multiple root rows with circular references, but none of the root rows has a null foreign key, the Integration Service cannot find a root row. You can add a FileName column to XML targets to name XML output documents based on data values.

Working with Heterogeneous Targets

You can output data to multiple targets in the same session. When the target types or database types of those targets differ from each other, you have a session with heterogeneous targets.

To create a session with heterogeneous targets, you can create a session based on a mapping with heterogeneous targets. Or, you can create a session based on a mapping with homogeneous targets and select different database connections.

A heterogeneous target has one of the following characteristics:

- **Multiple target types.** You can create a session that writes to both relational and flat file targets.
- **Multiple target connection types.** You can create a session that writes to a target on an Oracle database and to a target on a DB2 database. Or, you can create a session that writes to multiple targets of the same type, but you specify different target connections for each target in the session.

All database connections you define in the Workflow Manager are unique to the Integration Service, even if you define the same connection information. For example, you define two database connections, Sales1 and Sales2. You define the same user name, password, connect string, code page, and attributes for both Sales1 and Sales2. Even though both Sales1 and Sales2 define the same connection information, the Integration Service treats them as different database connections. When you create a session with two relational targets and specify Sales1 for one target and Sales2 for the other target, you create a session with heterogeneous targets.

You can create a session with heterogeneous targets in one of the following ways:

- Create a session based on a mapping with targets of different types or different database types. In the session properties, keep the default target types and database types.
- Create a session based on a mapping with the same target types. However, in the session properties, specify different target connections for the different target instances, or override the target type to a different type.

You can specify the following target type overrides in a session:

- **Relational target to flat file.**
- **Relational target to any other relational database type.** Verify the datatypes used in the target definition are compatible with both databases.
- **SAP BW target to a flat file target type.**

Note: When the Integration Service runs a session with at least one relational target, it performs database transactions per target connection group. For example, it orders the target load for targets in a target connection group when you enable constraint-based loading.

Reject Files

During a session, the Integration Service creates a reject file for each target instance in the mapping. If the writer or the target rejects data, the Integration Service writes the rejected row into the reject file. The reject file and session log contain information that helps you determine the cause of the reject.

Each time you run a session, the Integration Service appends rejected data to the reject file. Depending on the source of the problem, you can correct the mapping and target database to prevent rejects in subsequent sessions.

Note: If you enable row error logging in the session properties, the Integration Service does not create a reject file. It writes the reject rows to the row error tables or file.

Locating Reject Files

The Integration Service creates reject files for each target instance in the mapping. It creates reject files in the session reject file directory. Configure the target reject file directory on the Mapping tab for the session. By default, the Integration Service creates reject files in the $PMBadFileDir process variable directory.

When you run a session that contains multiple partitions, the Integration Service creates a separate reject file for each partition. The Integration Service names reject files after the target instance name. The default name for reject files is filename_partitionnumber.bad. The reject file name for the first partition does not contain a partition number.

For example,

```
/home/directory/filename.bad
/home/directory/filename2.bad
/home/directory/filename3.bad
```

The Workflow Manager replaces slash characters in the target instance name with underscore characters.

To find a reject file name and path, view the target properties settings on the Mapping tab of session properties.

Reading Reject Files

After you locate a reject file, you can read it using a text editor that supports the reject file code page. Reject files contain rows of data rejected by the writer or the target database. Though the Integration Service writes the entire row in the reject file, the problem generally centers on one column within the row. To help you determine which column caused the row to be rejected, the Integration Service adds row and column indicators to give you more information about each column:

- **Row indicator.** The first column in each row of the reject file is the row indicator. The row indicator defines whether the row was marked for insert, update, delete, or reject.

  If the session is a user-defined commit session, the row indicator might indicate whether the transaction was rolled back due to a non-fatal error, or if the committed transaction was in a failed target connection group.
- **Column indicator.** Column indicators appear after every column of data. The column indicator defines whether the column contains valid, overflow, null, or truncated data.

The following sample reject file shows the row and column indicators:

```
0,D,1921,D,Nelson,D,William,D,415-541-5145,D
0,D,1922,D,Page,D,Ian,D,415-541-5145,D
0,D,1923,D,Osborne,D,Lyle,D,415-541-5145,D
0,D,1928,D,De Souza,D,Leo,D,415-541-5145,D
0,D,2001123456789,O,S. MacDonald,D,Ira,D,415-541-514566,T
```

Row Indicators

The first column in the reject file is the row indicator. The row indicator is a flag that defines the update strategy for the data row.

The following table describes the row indicators in a reject file:

Row Indicator	Meaning	Rejected By
0	Insert	Writer or target
1	Update	Writer or target
2	Delete	Writer or target
3	Reject. Marked for reject by an update strategy expression.	Writer
4	Rolled-back insert	Writer
5	Rolled-back update	Writer
6	Rolled-back delete	Writer
7	Committed insert	Writer
8	Committed update	Writer
9	Committed delete	Writer

Column Indicators

A column indicator appears after every column of data. A column indicator defines whether the data is valid, overflow, null, or truncated.

The column indicator “D” also appears after each row indicator.

The following table describes the column indicators in a reject file:

Column Indicator	Type of data	Writer Treats As
D	Valid data.	Good data. Writer passes it to the target database. The target accepts it unless a database error occurs, such as finding a duplicate key.
N	Null. The column contains a null value.	Good data. Writer passes it to the target, which rejects it if the target database does not accept null values.
T	Truncated. String data exceeded a specified precision for the column, so the value was truncated.	Bad data, if you configured the mapping target to reject overflow or truncated data.

Null columns appear in the reject file with commas marking their column. The following example shows a null column surrounded by good data:

```
0,D,5,D,,N,5,D
```

Either the writer or target database can reject a row. Consult the log to determine the cause for rejection.

Connection Objects

Connection Objects Overview

Before you create and run sessions, you must configure connections in the Workflow Manager. A connection object is a global object that defines a connection in the repository. You create and modify connection objects and assign permissions to connection objects in the Workflow Manager.

Connection Types

When you create a connection object, choose the connection type in the Connection Browser. Some connection types also have connection subtypes. For example, a relational connection type has subtypes such as Oracle and Microsoft SQL Server. Define the values for the connection based on the connection type and subtype.

When you configure a session, you can choose the connection type and select a connection to use. You can also override the connection attributes for the session or create a connection. Set the connection type on the mapping tab for each object.

The following table describes the connection types that you can create or choose when you configure a session:

Table 1. Connection Types

Connection Types	Description
Relational	Relational connection to source, target, lookup, or stored procedure database. When you configure a session, you cannot change the relational connection type.
FTP	FTP or SFTP connection to the FTP host. When you configure a session, select an FTP connection type to access flat files or XML files through FTP. Specify the FTP connection when you configure source or target options. Select an FTP connection in the Value column.
Loader	Relational connection to the external loader for the target, such as IBM DB2 Autoloader or Teradata FastLoad. When you configure a session, choose File as the writer type for the relational target instance. Select a Loader connection to load output files to teradata, Oracle, DB2, or Sybase IQ through an external loader. Select a loader connection in the Value column.
Queue	Database connection for message queues, such as WebSphere MQ or MSMQ. Select a Queue connection type to access an MSMQ or WebSphere MQ source, or if you want to write messages to a WebSphere MQ message queue. Select an MQ connection in the Value column. For static WebSphere MQ targets, set the connection type to FTP or Queue. For dynamic MQSeries targets, set the connection type to Queue.

Connection Types	Description
Application	Connection to source or target application, such as Netezza or SAP NetWeaver. Select an Application connection type to access PowerExchange sources and targets and Teradata FastExport sources. You can also access transformations such as HTTP, Salesforce Lookup, and BAPI/RFC transformations.
None	Connection type not available in the Connection Browser. When you configure a session, select None if the mapping contains a flat file or XML file source or target or an associated source for WebSphere MQ.

Note: For information about connections to PowerExchange see *PowerExchange Interfaces for PowerCenter*.

Database User Names and Passwords

The Workflow Manager requires a database user name and password when you configure a connection. The database user must have the appropriate read and write database permissions to access the database.

Session Parameters

You can enter session parameter $Param*Name* as the database user name and password, and define the user name and password in a parameter file. For example, you can use a session parameter, $ParamMyDBUser, as the database user name, and set $ParamMyDBUser to the user name in the parameter file.

To use a session parameter for the database password, enable the Use Parameter in Password option and encrypt the password by using the *pmpasswd* command line program. Encrypt the password by using the CRYPT_DATA encryption type. For example, to encrypt the database password “monday,” enter the following command:

```
pmpasswd monday -e CRYPT_DATA
```

Databases that Do Not Allow User Names and Passwords

Some database drivers, such as ISG Navigator, do not allow user names and passwords. Since the Workflow Manager requires a database user name and password, PowerCenter provides reserved words to register databases that do not allow user names and passwords:

- PmNullUser
- PmNullPasswd

Use the PmNullUser user name if you use one of the following authentication methods:

- **Oracle OS Authentication.** Oracle OS Authentication lets you log in to an Oracle database if you have a login name and password for the operating system. You do not need to know a database user name and password. PowerCenter uses Oracle OS Authentication when the connection user name is PmNullUser and the connection is for an Oracle database.

- **IBM DB2 client authentication.** IBM DB2 client authentication lets you log in to an IBM DB2 database without specifying a database user name or password if the IBM DB2 server is configured for external authentication or if the IBM DB2 server is on the same as the Integration Service process. PowerCenter uses IBM DB2 client authentication when the connection user name is PmNullUser and the connection is for an IBM DB2 database.

Use the PmNullUser user name with any of the following connection types:

- **Relational database connections.** Use for Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow user names,
- **External loader connections.** Use for Oracle OS Authentication or IBM DB2 client authentication.
- **HTTP connections.** Use if the HTTP server does not require authentication.
- **PowerChannel relational database connections.** Use for Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow user names.
- **Web Services connections.** Use if the web service does not require a user name.

User Permissions for Oracle

Oracle uses temporary tablespaces to store temporary LOB data (BLOB, CLOB, or NCLOB data). When you run a session that reads from or writes to Oracle LOB columns, PowerCenter uses the Oracle temporary tablespace available for the database user account to store temporary LOB data.

Grant the database user permission to access and create temporary tablespaces. If the user does not have sufficient permission, the Integration Service fails the session.

Native Connect Strings

When you configure a connection object, you must provide connection information. Use native connect string syntax for the following connection types:

- **Relational database connections.** Use to connect to all databases except Microsoft SQL Server and Sybase ASE.
- **External loader connection.** Use to connect to all databases.
- **PowerChannel relational database connections.** Use to connect with all databases except Microsoft SQL Server and Sybase ASE.
- **PeopleSoft application connections.** Use to connect to the underlying database of the PeopleSoft system for DB2, Oracle, and Informix databases.

The following table lists the native connect string syntax for each supported database when you create or update connections:

Database	Connect String Syntax	Example
IBM DB2	*dbname*	`mydatabase`

Microsoft SQL Server	*servername@dbname*
	If you want to use SSL encryption, use the following connect string syntax:
	`servername@dbname;Encrypt=Yes`
		`sqlserver@mydatabase`
		The following example includes SSL encryption:
		`sqlserver@mydatabase;Encrypt=Yes`

Database	Connect String Syntax	Example
Oracle	*dbname*.world (same as TNSNAMES entry)	oracle.world
Sybase ASE	*servername@dbname*	sambrown@mydatabase
Teradata[1]	*ODBC_data_source_name* or *ODBC_data_source_name@db_name* or *ODBC_data_source_name@db_user_name*	TeradataODBC TeradataODBC@mydatabase TeradataODBC@jsmith

[1]. *Use Teradata ODBC drivers to connect to source and target databases.*

Connection Variable Values

Enter the database connection you want the Integration Service to use for the $Source and $Target connection variables. You can select a connection object, or you can use the $DBConnection*Name* or $AppConnection*Name* session parameter if you want to define the connection value in a parameter file.

When you configure a mapping, you can specify the database location to use $Source or $Target variable for Lookup and Stored Procedure transformations. You can also configure the $Source variable to specify the source connection for relational sources and the $Target variable to specify the target connection for relational targets in the session properties.

If you use $Source or $Target in a Lookup or Stored Procedure transformation, you can configure the connection value on the Properties tab or Mapping tab of the session. When you configure $Source Connection Value or $Target Connection Value, the Integration Service uses that connection when it runs the session. If you do not configure $Source Connection Value or $Target Connection Value, the Integration Service determines the database connection to use when it runs the session.

The following table describes how the Integration Service determines the value of $Source when you do not configure $Source Connection Value:

Table 2. Connection Used for $Source Variable

Mapping Objects	Connection Used
One source	The database connection you specify for the source.
Joiner transformation is before a Lookup or Stored Procedure transformation	The database connection for the detail source.
Lookup or Stored Procedure transformation is before a Joiner transformation	The database connection for the source connected to the transformation.
Unconnected Lookup or Stored Procedure transformation	None. The session fails.

The following table describes how the Integration Services determines the value of $Target when you do not configure $Target Connection Value in the session properties:

Table 3. Connection Used for $Target

$Target	Connection Used
One target	The database connection you specify for the target.
Multiple relational targets	None. The session fails.
Unconnected Lookup or Stored Procedure transformation.	None. The session fails.

Configuring a Session to Use Connection Variables

If the source or target is a database, you can use connection variables.

To enter the database connection for the $Source and $Target connection variables:

1. In the session properties, select the Properties tab or the Mapping tab, Connections node.
2. Click the Open button in $Source Connection Value or $Target Connection Value field.

   The Connection Browser dialog box appears.
3. Select a connection variable or session parameter.

   You can enter the $Source or $Target connection variable, or the $DBConnection*Name* or $AppConnection*Name* session parameter. If you enter a session parameter, define the parameter in the parameter file. If you do not define a value for the session parameter, the Integration Service determines which database connection to use when it runs the session.
4. Click OK.

Connection Attribute Overrides

When you configure the source and target instances, you can override connection attributes and define some attributes that are not in the connection object. You can override connection attributes based on how you configure the source or target instances.

You can override connection attributes when you configure the source or target session properties in the following ways:

- You use an FTP, queue, external loader, or application connection for a non-relational source or target.
- You use an FTP, queue, or external loader connection for a relational target.
- You use an application connection for a relational source.

You configure connections in the Connections settings on the Mapping tab.

You can override connection attributes in the session or in the parameter file:

- **Session.** Select the connection object and override attributes in the session.
- **Parameter file.** Use a session parameter to define the connection and override connection attributes in the parameter file.

Overriding Connection Attributes

You can override the connection attributes on the Mapping tab of the session properties.t

1. On the Mapping tab, select the source or target instance in the Connections node.
2. Select the connection type.
3. Click the Open button in the value field to select a connection object.
4. Choose the connection object.
5. Click Override.
6. Update the attributes you want to change.
7. Click OK.

Connection Object Code Pages

Code pages must be compatible for accurate data movement. You must select a code page for most types of connection objects. The code page of a database connection must be compatible with the database client code page. If the code pages are not compatible, sessions may hang, data may become inconsistent, or you might receive a database error, such as:

```
ORA-00911: Invalid character specified.
```

The Workflow Manager filters the list of code pages for connections to ensure that the code page for the connection is a subset of the code page for the repository. It lists the five code pages you have most recently selected. Then it lists all remaining code pages in alphabetical order.

If you configure the Integration Service for code page validation, the Integration Service enforces code page compatibility at run time. The Integration Service ensures that the target database code page is a superset of the source database code page.

When you change the code page in a connection object, you must choose one that is compatible with the previous code page. If the code pages are incompatible, the Workflow Manager invalidates all sessions using that connection.

If you configure the PowerCenter Client and Integration Service for relaxed code page validation, you can select any supported code page for source and target connections. If you are familiar with the data and are confident that it will convert safely from one code page to another, you can run sessions with incompatible source and target data code pages. It is your responsibility to ensure your data will convert properly.

SSL Authentication Certificate Files

Before you configure an HTTP connection or a Web Services Consumer connection to use SSL authentication, you may need to configure certificate files. If the Integration Service authenticates the HTTP server or web service provider, you configure the trust certificates file. If the HTTP server or web service provider authenticates the Integration Service, you configure the client certificate file and the corresponding private key file, passwords, and file types. You can generate client certificate and private key files by running the OpenSSL commands.

The trust certificates file (ca-bundle.crt) contains certificate files from major, trusted certificate authorities. If the certificate bundle does not contain a certificate from a certificate authority that the session uses, you can convert the certificate of the HTTP server or web service provider to PEM format and append it to the ca-bundle.crt file.

The private key for a client certificate must be in PEM format.

Generate Client Certificate and Private Key Files

If the web service provider authenticates the PowerCenter Integration Service, you must configure the client certificate and private key files. You can generate client certificate and private key files and use these files to configure the Web Service Consumer application connection.

You can generate the client certificate and private key files in a single file or as separate files.

Generate One Certificate File

To generate the client certificate file and private key file in a single file, use the following command:

```
openssl pkcs12 -in <certificate authority file>.p12 -out test1.pem -clcerts
```

The command generates a single certificate file in the PEM format. In the Web Service Consumer application connection, use the single certificate file while configuring both the client certificate file and the private key file. Use the password that you provide after running the OpenSSL command to configure the Web Service Consumer application connection.

Generate Keys in Separate Files

- To generate the client certificate file, use the following command:

  ```
  openssl pkcs12 -in <certificate authority file>.p12 -nokeys -out clientcert.pem
  ```
- To generate the private key file, use the following command:

  ```
  openssl pkcs12 -in <certificate authority file>.p12 -nocerts -out pk.pem
  ```

The command generates certificate files in the PEM format. In the Web Service Consumer application connection, specify the fully qualified path along with the client certificate and private key files. Use the passwords that you provide after running the OpenSSL commands to configure the Web Service Consumer application connection.

Configure the Web Service Consumer Application Connection

Use the client certificate file, the corresponding private key file, and the passwords to configure a Web Service application connection to use SSL authentication. You can access the Web Service application connection from the Application type connection in the Workflow Manager.

Converting Certificate Files from Other Formats

Certificate files have the following formats:

- **DER.** Files with the .cer or .der extension.
- **PEM.** Files with the .pem extension.
- **PKCS12.** Files with the .pfx or .P12 extension.

When you append certificates to the ca-bundle.crt file, the HTTP server certificate files must use the PEM format. Use the OpenSSL utility to convert certificates from one format to another. You can get OpenSSL at http://www.openssl.org.

For example, to convert the DER file named server.der to PEM format, use the following command:

```
openssl x509 -in server.der -inform DER -out server.pem -outform PEM
```

If you want to convert the PKCS12 file named server.pfx to PEM format, use the following command:

```
openssl pkcs12 -in server.pfx -out server.pem
```

To convert a private key named key.der from DER to PEM format, use the following command:

```
openssl rsa -in key.der -inform DER -outform PEM -out keyout.pem
```

For more information, refer to the OpenSSL documentation. After you convert certificate files to the PEM format, you can append them to the trust certificates file. Also, you can use PEM format private key files with the HTTP transformation or PowerExchange for Web Services.

Adding Certificates to the Trust Certificates File

If the HTTP server or web service provider uses a certificate that is not included in the ca-bundle.crt file, you can add the certificate to the ca-bundle.crt file.

1. Use Internet Explorer to locate the certificate and create a copy:
    - Access the HTTP server or web service provider using HTTPS.
    - Double-click the padlock icon in the status bar of Internet Explorer.
    - In the Certificate dialog box, click the Details tab.
    - Select the Authority Information Access field.
    - Click Copy to File.
    - Use the Certificate Export Wizard to copy the certificate in DER format.
2. Convert the certificate from DER to PEM format.
3. Append the PEM certificate file to the certificate bundle, ca-bundle.crt.

    The ca-bundle.crt file is located in the following directory: <PowerCenter Installation Directory>/server/bin

For more information about adding certificates to the ca-bundle.crt file, see the curl documentation at http://curl.haxx.se/docs/sslcerts.html.

Connection Object Permissions

You can access global connection objects from all folders in the repository and use them in any session. The Workflow Manager assigns owner permissions to the user who creates the connection object. The owner has all permissions. You can change the owner but you cannot change the owner permissions. You can assign permissions on a connection object to users, groups, and all others for that object.

The Workflow Manager assigns default permissions for connection objects to users, groups, and all others if you enable enhanced security.

You can specify read, write, and execute permissions for each user and group. You can perform the following types of tasks with different connection object permissions in combination with user privileges and folder permissions:

- **Read.** View the connection object in the Workflow Manager and Repository Manager. When you have read permission, you can perform tasks in which you view, copy, or edit repository objects associated with the connection object.
- **Write.** Edit the connection object.
- **Execute.** Run sessions that use the connection object.

To assign or edit permissions on a connection object, select an object from the Connection Object Browser, and click Permissions.

You can perform the following tasks to manage permissions on a connection object:

- Change connection object permissions for users and groups.
- Add users and groups and assign permissions to users and groups on the connection object.
- List all users to see all users that have permissions on the connection object.
- List all groups to see all groups that have permissions on the connection object.
- List all, to see all users, groups, and others that have permissions on the connection object.
- Remove each user or group that has permissions on the connection object.
- Remove all users and groups that have permissions on the connection object.
- Change the owner of the connection object.

If you change the permissions assigned to a user that is currently connected to a repository in a PowerCenter Client tool, the changed permissions take effect the next time the user reconnects to the repository.

Environment SQL

The Integration Service runs environment SQL in auto-commit mode and closes the transaction after it issues the SQL. Use SQL commands that do not depend on a transaction being open during the entire read or write process. For example, if a source database is set to read only mode and you create an environment SQL statement in the source connection to set the transaction to read only, the Integration Service issues a commit after it runs the SQL and cannot read the source in read only mode.

You can configure connection environment SQL or transaction environment SQL.

Use environment SQL for source, target, lookup, and stored procedure connections. If the SQL syntax is not valid, the Integration Service does not connect to the database, and the session fails.

Note: When a connection object has “environment SQL,” the connection uses “connection environment SQL.”

Connection Environment SQL

This custom SQL string sets up the environment for subsequent transactions. The Integration Service runs the connection environment SQL each time it connects to the database. If you configure connection environment SQL in a target connection, and you configure three partitions for the pipeline, the

Integration Service runs the SQL three times, once for each connection to the target database. Use SQL commands that do not depend on a transaction being open during the entire read or write process.

For example, use the following SQL statement to set the quoted identifier parameter for the duration of the connection:

```
SET QUOTED_IDENTIFIER ON
```

Use the SQL statement in the following situations:

- You want to set up the connection environment so that double quotation marks are object identifiers.
- You configure the target load type to Normal and the Microsoft SQL Server target name includes spaces.

Transaction Environment SQL

This custom SQL string also sets up the environment, but the Integration Service runs the transaction environment SQL at the beginning of each transaction.

Use SQL commands that depend on a transaction being open during the entire read or write process. For example, you might use the following statement as transaction environment SQL to modify how the session handles characters:

```
ALTER SESSION SET NLS_LENGTH_SEMANTICS=CHAR
```

This command must be run before each transaction. The command is not appropriate for connection environment SQL because setting the parameter once for each connection is not sufficient.

Guidelines for Configuring Environment SQL

Consider the following guidelines when creating the SQL statements:

- You can enter any SQL command that is valid in the database associated with the connection object. The Integration Service does not allow nested comments, even though the database might.
- When you enter SQL in the SQL Editor, you type the SQL statements.
- Use a semicolon (;) to separate multiple statements.
- The Integration Service ignores semicolons within /*...*/.
- If you need to use a semicolon outside of comments, you can escape it with a backslash (\).
- You can use parameters and variables in the environment SQL. Use any parameter or variable type that you can define in the parameter file. You can enter a parameter or variable within the SQL statement, or you can use a parameter or variable as the environment SQL. For example, you can use a session parameter, $ParamMyEnvSQL, as the connection or transaction environment SQL, and set $ParamMyEnvSQL to the SQL statement in a parameter file.
- You can configure the table owner name using sqlid in the connection environment SQL for a DB2 connection. However, the table owner name in the target instance overrides the SET sqlid statement in environment SQL. To use the table owner name specified in the SET sqlid statement, do not enter a name in the target name prefix.

Connection Resilience

Connection resilience is the ability of the Integration Service to tolerate temporary network failures when connecting to a relational database, an application, or the PowerExchange Listener. The Integration Service can also tolerate the temporary unavailability of the relational database, application, or PowerExchange Listener. The Integration Services is resilient to failures when it initializes the connection to the source or target and when it reads data from a source or writes data to a target.

You configure the resilience retry period in the connection object. You can configure the retry period for source, target, SQL transformation, and Lookup transformation connections. When a network failure occurs or the source or target becomes unavailable, the Integration Service attempts to reconnect for the amount of time configured for the **Connection Retry Period** property. If the Integration Service cannot reconnect to the source or target within the retry period, the session fails.

PowerExchange does not support runtime connection resilience for database connections other than those used for PowerExchange Express CDC for Oracle. Configure the workflow for automatic recovery of terminated tasks if recovery from a dropped PowerExchange connection is required. PowerExchange also does not support runtime resilience of connections between the Integration Service and PowerExchange Listener after the initial connection attempt. However, you can configure resilience for the initial connection attempt by setting the **Connection Retry Period** property to a value greater than 0 when you define PowerExchange Client for PowerCenter (PWXPC) relational and application connections. The Integration Service then retries the connection to the PowerExchange Listener after the initial connection attempt fails. If the Integration Service cannot connect to the PowerExchange Listener within the retry period, the session fails.

The Integration Service will *not* attempt to reconnect to a source or target in the following situations:

- The database connection object is for an Informix connection.
- The transformation associated with the connection object is not configured for deterministic and repeatable output.
- The value for the DTM buffer size is less than what the session requires.
- The truncate the target table option is enabled for a target and the connection fails during execution of the truncate query.
- The database connection fails during a commit or rollback.

Use the retry period with the following connection types:

- Relational database connections
- FTP connections
- JMS connections
- WebSphere MQ queue connections
- HTTP application connections
- Web Services Consumer application connections

Note: For a database connection to be resilient, the source or target must be a highly available database and you must have the high availability option or the real-time option.

Relational Database Connections

Use a relational connection object for each source, target, lookup, and stored procedure database that you want to access.

The following table describes the properties that you configure for a relational database connection:

Property	Description
Name	Name you want to use for this connection. The connection name cannot contain spaces or other special characters, except for the underscore.
Type	Type of database.
Use Kerberos Authentication	Indicates that the database to connect to runs on a network that uses Kerberos authentication. If this option is selected, you cannot set the user name and password in the connection object. The connection uses the credentials of the user account that runs the session that connects to the database. The user account must have a user principal on the Kerberos network where the database runs. Informatica supports Kerberos authentication for native relational connections to the following databases: Oracle, DB2, SQL Server, and Sybase.
User Name	Database user name with the appropriate read and write database permissions to access the database. For Oracle connections that process BLOB, CLOB, or NCLOB data, the user must have permission to access and create temporary tablespaces. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters. If you use Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow user names, enter PmNullUser. For Teradata connections, this overrides the default database user name in the ODBC entry. Not available if the **Use Kerberos Authentication** option is selected.
Use Parameter in Password	Indicates that the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. For Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow passwords, enter PmNullPassword. For Teradata connections, this overrides the database password in the ODBC entry. Passwords must be in 7-bit ASCII. Not available if the **Use Kerberos Authentication** option is selected.
Connect String	Connect string used to communicate with the database. For syntax, see "Native Connect Strings" on page 118. Required for all databases except Microsoft SQL Server and Sybase ASE.
Provider Type	The connection provider that you want to use to connect to the Microsoft SQL Server database. You can select the following provider types: - ODBC - Oledb(Deprecated) Default is ODBC.

Property	Description
Use DSN	Enables the PowerCenter Integration Service to use the Data Source Name for the connection. If you select the Use DSN option, the PowerCenter Integration Service retrieves the database and server names from the DSN. If you do not select the Use DSN option, you must provide the database and server names.
Code Page	Code page the Integration Service uses to read from a source database or write to a target database or file.
Connection Environment SQL	Runs an SQL command with each database connection. Default is disabled.
Transaction Environment SQL	Runs an SQL command before the initiation of each transaction. Default is disabled.
Enable Parallel Mode	Enables parallel processing when loading data into a table in bulk mode. Default is enabled.
Database Name	Name of the database. For Teradata connections, this overrides the default database name in the ODBC entry. Also, if you do not enter a database name for a Teradata or Sybase ASE connection, the Integration Service uses the default database name in the ODBC entry. If you do not enter a database name, connection-related messages do not show a database name when the default database is used.
Data Source Name	Name of the Teradata ODBC data source.
Server Name	Database server name. Use to configure workflows.
Packet Size	Use to optimize the native drivers for Sybase ASE and Microsoft SQL Server.
Domain Name	The name of the domain. Used for Microsoft SQL Server on Windows.
Use Trusted Connection	If selected, the Integration Service uses Windows authentication to access the Microsoft SQL Server database. The user name that starts the Integration Service must be a valid Windows user with access to the Microsoft SQL Server database.
Connection Retry Period	Number of seconds the Integration Service attempts to reconnect to the database if the connection fails. If the Integration Service cannot connect to the database in the retry period, the session fails. Default value is 0.
Impersonate User	The name of the impersonate user to connect to Oracle. The user name specified in the Oracle connection must have the impersonate user privileges. Applicable only for Oracle connections.

RELATED TOPICS:

- "Target Connections" on page 80
- "FTP Connections" on page 130

Copying a Relational Database Connection

When you make a copy of a relational database connection, the Workflow Manager retains the connection properties that apply to the relational database type you select. The copy of the connection is invalid if a required connection property is missing. Edit the connection properties manually to validate the connection.

The Workflow Manager appends an underscore and the first three letters of the relational database type to the name of the new database connection. For example, you have lookup table in the same database as your source definition. You you make a copy of the Microsoft SQL Server database connection called Dev_Source. The Workflow Manager names the new database connection Dev_Source_Mic. You can edit the copied connection to use a different name.

To copy a relational database connection:

1. Click Connections > Relational.

   The Relational Connection Browser appears.
2. Select the connection you want to copy.

   Tip: Hold the shift key to select more than one connection to copy.
3. Click Copy As.

   The Select Subtype dialog box appears.
4. Select a relational database type for the copy of the connection.

   If you copy one database connection object as a different type of database connection, you must reconfigure the connection properties for the copied connection.
5. Click OK.

   The Workflow Manager retains connection properties that apply to the database type. If a required connection property does not exist, the Workflow Manager displays a warning message. This happens when you copy a connection object as a different database type or copy a connection object that is already invalid.
6. Click OK to close the warning dialog box.

   The copy of the connection appears in the Relational Connection Browser.
7. If the copied connection is invalid, click the Edit button to enter required connection properties.
8. Click Close to close the Relational Connection Browser dialog box.

Relational Database Connection Replacement

You can replace a relational database connection with another relational database connection. For example, you might have several sessions that you want to write to another target database. Instead of editing the properties for each session, you can replace the relational database connection for all sessions in the repository that use the connection.

When you replace database connections, the Workflow Manager replaces the relational database connections in the following locations for all sessions using the connection:

- Source connection
- Target connection
- Connection Information property in Lookup and Stored Procedure transformations
- $Source Connection Value session property
- $Target Connection Value session property

When the repository contains both relational and application connections with the same name, the Workflow Manager replaces the relational connections only if you specified the connection type as relational in all locations.

The Integration Service uses the updated connection information the next time the session runs.

You must close all folders before replacing a relational database connection.

Replacing a Connection Object

Replace a connection object when you want to update for all sessions in the repository that use the connection.

1. Close all folders in the repository.
2. Click Connections > Replace.

   The Replace Connections dialog box appears.
3. Click the Add button to replace a connection.
4. In the From list, select a relational database connection you want to replace.
5. In the To list, select the replacement relational database connection.
6. Click Replace.

   All sessions in the repository that use the From connection now use the connection you select in the To list.

FTP Connections

Use an FTP connection object for each source or target that you want to access through FTP or SFTP.

To connect to an SFTP server, create an FTP connection and enable SFTP. SFTP uses the SSH2 authentication protocol. Configure the authentication properties to use the SFTP connection. You can configure publickey or password authentication. The Integration Service connects to the SFTP server with the authentication properties you configure. If the authentication does not succeed, the session fails.

The following table describes the properties that you configure for an FTP connection:

Property	Description
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
User Name	User name necessary to access the host machine. Must be in 7-bit ASCII only. Required to connect to an SFTP server with password based authentication. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.

Property	Description
Password	Password for the user name. Must be in 7-bit ASCII only. Required to connect to an SFTP server with password based authentication. **Note:** When you specify *pmnullpasswd*, the PowerCenter Integration Service authenticates the user directly based on public key without performing the password authentication.

Host Name	Host name or dotted IP address of the FTP connection.
	Optionally, you can specify a port number between 1 and 65535, inclusive. Default for FTP is 21. Use the following syntax to specify the host name:
	`hostname:port_number`
	Or,
	`IP address:port_number`
	When you specify a port number, enable that port number for FTP on the host machine.
	If you enable SFTP, specify a host name or port number for an SFTP server. Default for SFTP is 22.

Default Remote Directory	Default directory on the FTP host used by the Integration Service. Do not enclose the directory in quotation marks. You can enter a parameter or variable for the directory. Use any parameter or variable type that you can define in the parameter file. Depending on the FTP server you use, you may have limited options to enter FTP directories. In the session, when you enter a file name without a directory, the Integration Service appends the file name to this directory. This path must contain the appropriate trailing delimiter. For example, if you enter c:\staging\ and specify data.out in the session, the Integration Service reads the path and file name as c:\staging\data.out. For SAP, you can leave this value blank. SAP sessions use the Source File Directory session property for the FTP remote directory. If you enter a value, the Source File Directory session property overrides it.
Retry Period	Number of seconds the Integration Service attempts to reconnect to the FTP host if the connection fails. If the Integration Service cannot reconnect to the FTP host in the retry period, the session fails. Default value is 0 and indicates an infinite retry period.
Use SFTP	Enables SFTP.
Public Key File Name	Public key file path and file name. Required if the SFTP server uses publickey authentication. Enabled for SFTP.
Private Key File Name	Private key file path and file name. Required if the SFTP server uses publickey authentication. Enabled for SFTP.
Private Key File Password	Private key file password used to decrypt the private key file. Required if the SFTP server uses public key authentication and the private key is encrypted. Enabled for SFTP.

External Loader Connections

Use a loader connection object for each target that you want load through an external loader.

The following table describes the properties that you configure for an external loader connection:

Property	Description
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
User Name	Database user name with the appropriate read and write database permissions to access the database. If you use Oracle OS Authentication or IBM DB2 client authentication, enter PmNullUser. PowerCenter uses Oracle OS Authentication when the connection user name is PmNullUser and the connection is to an Oracle database. PowerCenter uses IBM DB2 client authentication when the connection user name is PmNullUser and the connection is to an IBM DB2 database. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters. You can connect to a database runs on a network that uses Kerberos authentication. To use Kerberos authentication for the database connection, set the user name to the reserved word *PmKerberosUser*. If you use Kerberos authentication, the connection uses the credentials of the user account that runs the session that connects to the database. The user account must have a user principal on the Kerberos network where the database runs.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. For Oracle OS Authentication or IBM DB2 client authentication, enter PmNullPassword. For Teradata connections, you can enter PmNullPasswd to prevent the password from appearing in the control file. Instead, the Integration Service writes an empty string for the password in the control file. Passwords must be in 7-bit ASCII. If you set the user name to *PmKerberosUser* to use Kerberos authentication for the database connection, set the password to the reserved word *PmKerberosPassword*. The connection uses the credentials of the user account that runs the session that connects to the database.
Connect String	Connect string used to communicate with the database. For syntax, see "Native Connect Strings" on page 118.

HTTP Connections

Use an application connection object for each HTTP server that you want to connect to.

Configure connection information for an HTTP transformation in an HTTP application connection. The Integration Service can use HTTP application connections to connect to HTTP servers. HTTP application connections enable you to control connection attributes, including the base URL and other parameters.

If you want to connect to an HTTP proxy server, configure the HTTP proxy server settings in the Integration Service.

Configure an HTTP application connection in the following circumstances:

- The HTTP server requires authentication.
- You want to configure the connection timeout.

- You want to override the base URL in the HTTP transformation.

Note: Before you configure an HTTP connection to use SSL authentication, you may need to configure certificate files. For information about SSL authentication, see "SSL Authentication Certificate Files" on page 121.

The following table describes the properties that you configure for an HTTP connection:

Property	Description
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
User Name	Authenticated user name for the HTTP server. If the HTTP server does not require authentication, enter PmNullUser. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the authenticated user is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the authenticated user. If the HTTP server does not require authentication, enter PmNullPasswd.
Base URL	URL of the HTTP server. This value overrides the base URL defined in the HTTP transformation. You can use a session parameter to configure the base URL. For example, enter the session parameter $ParamBaseURL in the Base URL field, and then define $ParamBaseURL in the parameter file.
Timeout	Number of seconds the Integration Service waits for a connection to the HTTP server before it closes the connection.
Domain	Authentication domain for the HTTP server. This is required for NTLM authentication.
Trust Certificates File	File containing the bundle of trusted certificates that the client uses when authenticating the SSL certificate of a server. You specify the trust certificates file to have the Integration Service authenticate the HTTP server. By default, the name of the trust certificates file is ca-bundle.crt. For information about adding certificates to the trust certificates file, see "SSL Authentication Certificate Files" on page 121.
Certificate File	Client certificate that an HTTP server uses when authenticating a client. You specify the client certificate file if the HTTP server needs to authenticate the Integration Service.
Certificate File Password	Password for the client certificate. You specify the certificate file password if the HTTP server needs to authenticate the Integration Service.
Certificate File Type	File type of the client certificate. You specify the certificate file type if the HTTP server needs to authenticate the Integration Service. The file type can be PEM or DER. For information about converting certificate file types to PEM or DER, see "SSL Authentication Certificate Files" on page 121. Default is PEM.
Private Key File	Private key file for the client certificate. You specify the private key file if the HTTP server needs to authenticate the Integration Service.

Property	Description
Key Password	Password for the private key of the client certificate. You specify the key password if the web service provider needs to authenticate the Integration Service.
Key File Type	File type of the private key of the client certificate. You specify the key file type if the HTTP server needs to authenticate the Integration Service. The HTTP transformation uses the PEM file type for SSL authentication.
Authentication Type	Select one of the following authentication types to use when the HTTP server does not return an authentication type to the Integration Service: - Auto. The Integration Service attempts to determine the authentication type of the HTTP server. - Basic. Based on a non-encrypted user name and password. - Digest. Based on an encrypted user name and password. - NTLM. Based on encrypted user name, password, and domain. Default is Auto.

PowerExchange for Amazon Redshift Connections

When you configure an Amazon Redshift connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Amazon Redshift.

The following table describes the application connection properties:

Property	Description
Name	Name of the Amazon Redshift connection.
Type	The AmazonRSCloudAdapter connection type.
User Name	User name to access the Amazon Redshift database.
Password	Password for the Amazon Redshift database user name.

The following table describes the Amazon Redshift connection attributes:

Property	Description
Schema	Schema name for the Amazon Redshift tables. When you import objects from **AmazonRSCloudAdapter** in the PowerCenter Designer, the table names are listed in an alphabetical order. Default is public. **Note:** The public schema might not work for all the Amazon Redshift tables.
AWS Access Key ID	Amazon S3 bucket access key ID.
AWS Secret Access Key	Amazon S3 bucket secret access key ID.
Master Symmetric Key	Optional. Amazon S3 encryption key. Provide a 256-bit AES encryption key in the Base64 format.

Property	Description
Customer Master Key ID	Optional. Specify the customer master key ID or alias name generated by AWS Key Management Service (AWS KMS). You must generate the customer master key ID for the same region where Amazon S3 bucket reside. You can specify any of the following values: **Customer Generated Customer Master Key** Enables client-side or server-side encryption. **Default Customer Master Key** Enables client-side or server-side encryption. Only the administrator user of the account can use the default customer master key ID to enable client-side encryption.
JDBC URL	Amazon Redshift connection URL.
Number of bytes needed to support multibytes for varchar	Not applicable. This property is not supported as you cannot use the **Create Target** option from the target session properties on an Amazon Redshift target using the PowerCenter Client.

PowerExchange for Amazon S3 Connections

When you configure an Amazon S3 connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to Amazon S3.

The following table describes the Amazon S3 connection properties:

Property	Description
Name	The name of the Amazon S3 connection.
Type	The Amazon S3 connection type.
Access Key	The access key ID used to access the Amazon account resources. Required if you do not use AWS Identity and Access Management (IAM) authentication. **Note:** Ensure that you have valid AWS credentials before you create a connection.
Secret Key	The secret access key used to access the Amazon account resources. This value is associated with the access key and uniquely identifies the account. You must specify this value if you specify the access key ID. Required if you do not use AWS Identity and Access Management (IAM) authentication.
Folder Path	The complete path to the Amazon S3 objects and must include the bucket name and any folder name. Ensure that you do not use a forward slash at the end of the folder path. For example, `<bucket name>/<my folder name>`
Master Symmetric Key	Optional. Provide a 256-bit AES encryption key in the Base64 format when you enable client-side encryption. You can generate a key using a third-party tool. If you specify a value, ensure that you specify the **Encryption Type** as **Client Side Encryption** in the target session properties.

Property	Description
Customer Master Key ID	Optional. Specify the customer master key ID or alias name generated by AWS Key Management Service (AWS KMS). You must generate the customer master key for the same region where Amazon S3 bucket reside. You can specify any of the following values: **Customer Generated Customer Master Key** Enables client-side or server-side encryption. **Default Customer Master Key** Enables client-side or server-side encryption. Only the administrator user of the account can use the default customer master key ID to enable client-side encryption.
Code Page	The code page compatible with the Amazon S3 source. Select one of the following code pages: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode and non-Unicode data. - Shift-JIS. Select for double-byte character data. - ISO 8859-15 Latin 9 (Western European). - ISO 8859-2 Eastern European. - ISO 8859-3 Southeast European. - ISO 8859-5 Cyrillic. - ISO 8859-9 Latin 5 (Turkish). - IBM EBCDIC International Latin-1.
Region Name	The name of the region where the Amazon S3 bucket is available. Select one of the following regions: - Asia Pacific (Mumbai) - Asia Pacific (Seoul) - Asia Pacific (Singapore) - Asia Pacific (Sydney) - Asia Pacific (Tokyo) - AWS GovCloud - Canada (Central) - China (Beijing) - EU (Ireland) - EU (Frankfurt) - South America (Sao Paulo) - US East (Ohio) - US East (N. Virginia) - US West (N. California) - US West (Oregon) Default is US East (N. Virginia).
Formatting Options	Select a delimiter, text qualifier, or an escape character. If you want to specify a delimiter other than comma, tab, colon, and semi-colon, you must choose **Other** in File Formatting options. When you specify a delimiter in the **Other** option, the PowerCenter Integration Service might display an error message or the fields or metadata are not fetched as expected. You must follow the below guidelines when you use the **Other** option for delimiter in File Formatting Options: - You must use a single special character excluding comma as the delimiter such as, #, @, &, or ~. - You must not use multibyte characters as the delimiter. - You must not use text qualifiers such as, ' and " as the delimiter. **Note:** You cannot use this property when you configure an Amazon S3 connection in the Workflow Manager.

PowerChannel Relational Database Connections

Use a relational connection object for each database that you want to access through PowerChannel. If you have configured a relational database connection, and you want to create a PowerChannel connection, you can copy the connection.

The following table describes the properties that you configure for a PowerChannel relational database connection:

Property	**Description**
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
Type	Type of database.
User Name	Database user name with the appropriate read and write database permissions to access the database. If you use Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow user names, enter PmNullUser. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. For Oracle OS Authentication, IBM DB2 client authentication, or databases such as ISG Navigator that do not allow passwords, enter PmNullPassword. For Teradata connections, this overrides the database password in the ODBC entry. Passwords must be in 7-bit ASCII.
Connect String	Connect string used to communicate with the database. For syntax, see “Native Connect Strings” on page 118. Required for all databases except Microsoft SQL Server.
Code Page	Code page the Integration Service uses to read from a source database or write to a target database or file.
Database Name	Name of the database. If you do not enter a database name, connection-related messages do not show a database name when the default database is used.
Environment SQL	Runs an SQL command with each database connection. Default is disabled.
Rollback Segment	Name of the rollback segment.
Server Name	Database server name. Use to configure workflows.
Packet Size	Use to optimize the native drivers for Sybase ASE and Microsoft SQL Server.
Domain Name	The name of the domain. Used for Microsoft SQL Server on Windows.

Property	Description
Use Trusted Connection	If selected, the Integration Service uses Windows authentication to access the Microsoft SQL Server database. The user name that starts the Integration Service must be a valid Windows user with access to the Microsoft SQL Server database.
Remote PowerChannel Host Name	Host name or IP address for the remote PowerChannel Server that can access the database data.
Remote PowerChannel Port Number	Port number for the remote PowerChannel Server. Make sure the PORT attribute of the ACTIVE_LISTENERS property in the *PowerChannel*.properties file uses a value that other applications on the PowerChannel Server do not use.
Use Local PowerChannel	Select to use compression or encryption while extracting or loading data. When you select this option, you need to specify the local PowerChannel Server address and port number. The Integration Service uses the local PowerChannel Server as a client to connect to the remote PowerChannel Server and access the remote database.
Local PowerChannel Host Name	Host name or IP address for the local PowerChannel Server. Enter this option when you select the Use Local PowerChannel option.
Local PowerChannel Port Number	Port number for the local PowerChannel Server. Specify this option when you select the Use Local PowerChannel option. Make sure the PORT attribute of the ACTIVE_LISTENERS property in the *PowerChannel*.properties file uses a value that other applications on the PowerChannel Server do not use.
Encryption Level	Encryption level for the data transfer. Encryption levels range from 0 to 3. 0 indicates no encryption and 3 is the highest encryption level. Default is 0. Use this option only if you have selected the Use Local PowerChannel option.
Compression Level	Compression level for the data transfer. Compression levels range from 0 to 9. 0 indicates no compression and 9 is the highest compression level. Default is 2. Use this option only if you have selected the Use Local PowerChannel option.
Certificate Account	Certificate account to authenticate the local PowerChannel Server to the remote PowerChannel Server. Use this option only if you have selected the Use Local PowerChannel option. If you use the sample PowerChannel repository that the installation program set up, and you want to use the default certificate account in the repository, you can enter "default" as the certificate account.

PowerExchange for Db2 Warehouse Connections

The following table describes the Db2 Warehouse connection properties that you must configure:

Property	Description
User Name	Database user name with the appropriate read and write database permissions to access Db2 Warehouse.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name.

Property	Description
Connect String	ODBC data source to connect to Db2 Warehouse.
Database Name	Database name of Db2 Warehouse that you want to connect to.
Schema Name	The schema name in Db2 Warehouse from where you want to fetch the metadata.
Server Name	Host name of Db2 Warehouse.
Port Number	Network port number used to connect to the Db2 Warehouse server.
Driver Name	Specify the name of the IBM Data Server driver that you configured in the `odbcinst.ini` file. For example, IBM DB2 ODBC DRIVER - IBMDBCL1.
Advanced connection properties	Optional. Additional connection parameters that you want to use. Specify the connection parameters as key-value pairs in the following format, and separate each key-value pair with a semicolon: `<param1>=<value>;<param2>=<value>;<param3>=<value>....`

PowerExchange for Essbase Connections

When you configure a Essbase connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to Essbase.

The following table describes the Essbase connection properties that you must configure:

Connection Attribute	Description
User Name	User name to connect to Essbase.
Password	Password to connect to Essbase.
ServerHost	Essbase server name.
Application	Name of the application. Default is none. **Note:** For a Unicode application, specify the name of the Unicode application.
Database	Name of the database. Default is none.

PowerExchange for Greenplum Connections

When you configure a Greenplum connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to Greenplum.

The following table describes the Greenplum connection properties that you must configure:

Connection Attribute	Description
Name	Name for the Greenplum relational connection.
User Name	User name with permissions to access the Greenplum database. You can connect to a database that runs on a network that uses Kerberos authentication. To configure Kerberos authentication for the database connection, set the user name to the reserved word *PmKerberosUser*. If you use Kerberos authentication, the connection uses the credentials of the user account that runs the session to connect to the database. The user account must have a user principal on the Kerberos network where the database runs.
Password	Password to connect to the Greenplum database. If you set the user name to *PmKerberosUser* to use Kerberos authentication for the database connection, set the password to the reserved word *PmKerberosPassword*. The connection uses the credentials of the user account that runs the session to connect to the database.
Host Name	Host name or IP address of the Greenplum server.
Port	Greenplum server port number. If you enter 0, the gpload utility reads from the environment variable $PGPORT. Default is 5432.
Database	Name of the database.
Enable SSL	Select this option to establish secure communication between the gpload utility and the Greenplum server over SSL. Applicable for Greenplum connections used for loading data to Greenplum. Not applicable for Greenplum reader. Not applicable for Greenplum writer on the Windows platform.
Certificate Path	Path where the SSL certificates for the Greenplum server are stored. For information about the files that need to be present in the certificates path, see the gpload documentation.
Schema	Name of the schema that contains the metadata for Greenplum targets. Default is public.

PowerExchange for Google Analytics Connections

When you configure a Google Analytics connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Google Analytics.

The following table describes the Google Analytics connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
APIVersion	API that PowerExchange for Google Analytics uses to read from Google Analytics reports. Select **Core Reporting API v3**. **Note:** PowerExchange for Google Analytics does not support Analytics Reporting API v4.

PowerExchange for Google BigQuery Connections

When you configure a Google BigQuery connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Google BigQuery database.

The following table describes the Google BigQuery connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Connection mode	The mode that you want to use to read data from or write data to Google BigQuery. Select one of the following connection modes: - Simple. Flattens each field within the Record data type field as a separate field in the mapping. - Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the mapping. - Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping. Default is Simple.
Schema Definition File Path	Specifies a directory on the client machine where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. The JSON file name is the same as the Google BigQuery table name. Alternatively, you can specify a storage path in Google Cloud Storage where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. You can download the JSON file from the specified storage path in Google Cloud Storage to a local machine.

Property	Description
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.
Storage Path	This property applies when you read or write large volumes of data. Path in Google Cloud Storage where the PowerCenter Integration Service creates a local stage file to store the data temporarily. You can either enter the bucket name or the bucket name and folder name. For example, enter `gs://<bucket_name>` or `gs://<bucket_name>/<folder_name>`
Dataset ID	Not applicable for PowerExchange for Google BigQuery.
Use Legacy SQL For Custom Query	Uses Legacy SQL to define a custom query. You can clear this option, you must use Standard SQL to define a custom query.
Dataset Name for Custom Query	When you define a custom query, you must specify a Google BigQuery dataset.
Region id	The region name where the Google BigQuery dataset resides. For example, if you want to connect to a Google BigQuery dataset that resides in Las Vegas region, specify **us-west4** as the **Region ID**. **Note:** In the **Storage Path** connection property, ensure that you specify a bucket name or the bucket name and folder name that resides in the same region as the dataset in Google BigQuery. For more information about the regions supported by Google BigQuery, see the following Google BigQuery documentation:https://cloud.google.com/bigquery/docs/locations

PowerExchange for Google Cloud Spanner Connections

When you configure a Google Cloud Spanner connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Google Cloud Spanner.

The following table describes the Google Cloud Spanner connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the PowerCenter Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or fewer and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.

Property	Description
Type	The connection type. Select **Google Cloud Spanner**.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the bucket that you want to connect to.
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Instance ID	Name of the instance that you created in Google Cloud Spanner.

PowerExchange for Google Cloud Storage Connections

When you configure a Google Cloud Storage connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Google Cloud Storage database.

The following table describes the Google Cloud Storage connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.

PowerExchange for Hadoop Connections

Use a Hadoop HDFS application connection object for each Hadoop source or target that you want to access.

You connect to a Hadoop cluster through an HDFS host that runs the name node service for a Hadoop cluster.

The following table describes the properties that you configure for a Hadoop HDFS application connection:

Property	Description
Name	The connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore character.
User Name	The name of the user in the Hadoop group that is used to access the HDFS host.

Property	Description
Password	Password to access the HDFS host. Reserved for future use.

HDFS Connection URI	The URI to access HDFS. Use the value for the `fs.default.name` property for the NameNode URI. You can find the value for the property for the NameNode URI. You can find the value for the `fs.default.name` property in the `core-site.xml` configuration set.
	Syntax for Hadoop distributions:
	`hdfs://<namenode>:<port>`
	Where
	- <namenode> is the host name or IP address of the NameNode.
	- <port> is the port that the NameNode listens for remote procedure calls (RPC).
	Syntax for the MapR distribution:
	`maprfs:///`
	Syntax for the HDInsight distribution:
	- `adl:// <nameservices>`
	- `wasb://<nameservices>`

Hive Driver Name	The name of the Hive driver. By default, the driver name is: `org.apache.hive.jdbc.HiveDriver`

Hive URL	The URL to the Hive host.
	For MapR Ticket cluster, specify the URL in the following format:
	`jdbc:hive2://`*`hostname`*`:`*`portnumber`*`/default;auth=MAPRSASL/default;auth=MAPRSASL`
	For MapR Kerberos cluster, specify the URL in the following format:
	`jdbc:hive2://`*`hostname`*`:`*`portnumber`*`/default;auth=MAPRSASL/default;principal=<spn>`

Hive User Name	The Hive user name. Reserved for future use.
Hive Password	The password for the Hive user. Reserved for future use.
Hadoop Distribution	The name of the Hadoop distribution. Default is cloudera_cdh.

HANA ODBC Connection Properties

You must configure a HANA ODBC data source before you can import HANA sources.

The following table describes the HANA ODBC connection properties:

Property	Decription
Name	Name you want to use for this connection. The connection name cannot contain spaces or other special characters, except for the underscore.
Type	Read-only. Type of database. For SAP HANA, this property is set to ODBC.

Property	Decription
User Name	Database user name with the appropriate read and write database permissions to access the database. To define the user name in the parameter file, enter session parameter $ParamName as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates that the password for the database user name is a session parameter, $ParamName. Define the password in the workflow or session parameter file, and encrypt it by using the pmpasswd CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. Must be in 7-bit ASCII.
Connect String	Connect string used to communicate with the SAP HANA database.
Code Page	Code page the Integration Service uses to read from a source database or write to a target database.
Connection Environment SQL	Runs an SQL command with each database connection. Default is disabled.
Transaction Environment SQL	Runs an SQL command before the initiation of each transaction. Default is disabled.
Connection Retry Period	Number of seconds the Integration Service attempts to reconnect to the database if the connection fails. If the Integration Service cannot connect to the database in the retry period, the session fails. Default value is 0.
ODBC Subtype	Type of database to which ODBC connects. Select **SAP HANA**.

PowerExchange for JD Edwards EnterpriseOne Connections

When you configure a JD Edwards EnterpriseOne connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the JD Edwards EnterpriseOne.

The following table describes the connection properties:

Property	Description
User Name	User name to connect to JD Edwards EnterpriseOne database.
Password	Password for the user name.
Connect String	Name of the ODBC data source. **Note:** Use Informatica certified ODBC drivers for ODBC data source connection.

Property	Description
Connection Retry Period	Number of seconds that the PowerCenter Integration Service waits after making a request to connect to the database. If the PowerCenter Integration Service does not receive any response, the session fails. Default value is 0.
Control Table Name Prefix	Owner of the F0005 control table that contains UDC values. If the database user specified in the database connection is not the owner of the F0005 control table and the session is configured for UDC validation, specify the owner of the F0005 control table as the control table name prefix. You can use a parameter for this connection attribute.

PowerExchange for JMS Connections

Use an application connection object for each JMS source or target that you want to access.

You must configure two types of JMS application connections:

- JNDI application connection
- JMS application connection

JNDI Application Connection

Use a JNDI application connection for each JNDI server that you want to access.

When the Integration Service connects to the JNDI server, it retrieves information from JNDI about the JMS provider during the session. When you configure a JNDI application connection, you must specify connection properties in the Connection Object Definition dialog box.

The following table describes the properties that you configure for a JNDI application connection:

Property	Description
JNDI Context Factory	Name of the context factory that you specified when you defined the context factory for your JMS provider.
JNDI Provider URL	Provider URL that you specified when you defined the provider URL for your JMS provider.
JNDI UserName	User name.
JNDI Password	Password.

JMS Application Connection

Use a JMS application connection for each JMS provider you want to access.

When you configure a JMS application connection, you specify connection properties the Integration Service uses to connect to JMS providers during a session. Specify the JMS application connection properties in the Connection Object Definition dialog box.

The following table describes the properties that you configure for a JMS application connection:

Property	Description
JMS Destination Type	Select QUEUE or TOPIC for the JMS Destination Type. Select QUEUE if you want to read source messages from a JMS provider queue or write target messages to a JMS provider queue. Select TOPIC if you want to read source messages based on the message topic or write target messages with a particular message topic.
JMS Connection Factory Name	Name of the connection factory. The name of the connection factory must be the same as the connection factory name you configured in JNDI. The Integration Service uses the connection factory to create a connection with the JMS provider.
JMS Destination	Name of the destination. The destination name must match the name you configured in JNDI. Optionally, you can use the $Param*Name* session parameter for the destination name.
JMS UserName	User name.
JMS Password	Password.
JMS Recovery Destination	Recovery queue or recovery topic name, based on what you configure for the JMS Destination Type. Configure this option when you enable recovery for a real-time session that reads from a JMS or WebSphere MQ source and writes to a JMS target. **Note:** The session fails if the recovery destination does not match a recovery queue or topic name in the JMS provider.
Connection Retry Period	Number of seconds the Integration Service attempts to reconnect to JMS if the connection fails. If the Integration Service cannot connect to JMS in the retry period, the session fails. Default value is 0.
Retry Connection Error Code File Name	Name of the properties file that contains error codes that identify JMS connection errors. Default is pmjmsconnerr.properties.

PowerExchange for Kafka Connections

When you configure a Kafka connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Kafka cluster. You can configure SSL properties to connect to a Kafka Broker over SSL.

The following table describes the Kafka connection properties:

Property	Description
Kafka Broker List	The IP address and port combinations of the Kafka messaging system broker list. The IP address and port combination has the following format: `<IP Address>:<port>` You can enter multiple comma-separated IP address and port combinations.
Retry Timeout in seconds	Number of seconds the Integration Service attempts to reconnect to the Kafka broker to write data. If the source or target is not available for the time you specify, the mapping execution stops to avoid any data loss. Default is 180 seconds.

Property	Description
Kafka Broker Version	Select **Apache 0.10.1.1 and above** as the Kafka messaging broker version.
Additional Connection Properties	Comma-separated list of connection properties to connect to the Kafka broker. To connect to a Kafka broker over SSL, ensure that you set the following properties: - security.protocol - ssl.truststore.location - ssl.truststore.password If the `ssl.client.auth` parameter is set to `required` or `requested`, ensure that you set the following properties: - ssl.keystore.location - ssl.keystore.password - ssl.key.password For example, `security.protocol=SSL,ssl.truststore.location=<complete path and name of the client.truststore.jks file>,ssl.truststore.password=<client truststore password>,ssl.keystore.location=<complete path and name of the client.keystore.jks file>,ssl.keystore.password=<client keystore password>,ssl.key.password=<password for the individual key>` You can also specify the configuration properties required to connect to a Kerberised Kafka cluster. Specify the following property: `security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI`
SSL Mode	Specifies whether the PowerCenter Integration Service establishes a secure connection to the Kafka broker.You can select one of the following options: - disabled. The PowerCenter Integration Service establishes an unencrypted connection to the Kafka broker. - require. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker without verifying the identity of the server. - one-way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password. - two-way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password.
SSL TrustStore File Path	Applicable only if you select **one-way** or **two-way** as the SSL mode. The complete path and file name of the truststore file. The truststore file contains the SSL certificate that the Kafka cluster validates against the Kafka broker certificate.
SSL TrustStore Password	Applicable only if you select **one-way** or **two-way** as the SSL mode. The password for the truststore file.
SSL KeyStore File Path	Applicable only if you select **two-way** as the SSL mode. The complete path and name of the Java keystore file. The keystore file contains the certificate that the Kafka broker validates against the Kafka cluster certificate

Property	Description
SSL KeyStore Password	Applicable only if you select **two-way** as the SSL mode. The password for the keystore file.
Additional Security Properties	Optional. Comma-separated list of connection properties to connect to the Kafka broker in a secured way. For example, `security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=<kerberos name>,sasl.mechanism=GSSAPI,sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required useKeyTab=true doNotPrompt=true storeKey=true client=true keyTab="<Keytab Location>" principal="<principal>";`

PowerExchange for LDAP Connections

When you configure a LDAP connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the LDAP.

The following table describes the connection properties:

Property	Description
User Name	User name to connect to the LDAP directory server.
Password	Password to connect to the LDAP directory server. If the user name does not require the password, enter `infa_blank`.
Host Name	LDAP directory server host name. Default is localhost.
Port Number	LDAP directory server port number. Default is 389.
Anonymous Access	Select this option to establish an anonymous connection with the LDAP directory server. If you select this option, enter the user name and password as `anonymous`.
Security	Type of security used to establish secure connection with SSL or TLS. Default is None. If you do not select the security type or select the SSL option to establish a secure connection, the PowerCenter Integration Service ignores the TLS options.
TLS Options	TLS options used to establish secure connection or transfer data, or both, with the LDAP directory server. Default is None.

Microsoft Azure Blob Storage Connection Properties

When you configure a Microsoft Azure Blob Storage connection, you define the connection properties that the PowerCenter Integration Service uses to connect to Microsoft Azure Blob Storage.

The following table describes the Microsoft Azure Blob Storage connection properties:

Connection Property	Description
Account Name	Name of the Microsoft Azure Blob Storage account.
Account Key	Microsoft Azure Storage access key.
Container Name	Microsoft Azure Blob Storage container name.
File Delimiter	Character used to separate fields in the file. Default is a comma (,). Use a printable single-byte character delimiter that is not present in the data. You cannot use multibyte characters as delimiters.

PowerExchange for Microsoft Azure SQL Data Warehouse V3 Connections

A Microsoft Azure SQL Data Warehouse connection extracts data from and loads data to the Microsoft Azure SQL Data Warehouse. PowerExchange for Microsoft Azure SQL Data Warehouse V3 uses SOAP to connect to Microsoft Azure SQL Data Warehouse.

The following table describes PowerExchange for Microsoft Azure SQL Data Warehouse V3 connection properties:

Connection Properties	Description
Azure DW JDBC URL	Microsoft Azure SQL Data Warehouse JDBC connection string. For example, you can enter the following connection string: jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>
Azure DW JDBC Username	User name to connect to the Microsoft Azure SQL Data Warehouse account.
Azure DW JDBC Password	Password to connect to the Microsoft Azure SQL Data Warehouse account.
Azure DW Schema Name	Name of the schema in Microsoft Azure SQL Data Warehouse.
Azure Blob Account Name	Name of the Microsoft Azure Storage account to stage the files.
Azure Blob Account Key	Microsoft Azure Storage access key to stage the files.
Blob End-point	Type of Microsoft Azure end-points. You can select any of the following end-points: - `core.windows.net`: Default - `core.usgovcloudapi.net`: To select the US government Microsoft Azure end-points - `core.chinacloudapi.cn`: Not applicable

Microsoft Dynamics 365 for Sales Connection Properties

When you configure a Microsoft Dynamics 365 for Sales connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Microsoft Dynamics 365 for Sales database.

The following table describes the Microsoft Dynamics 365 for Sales connection properties:

Property	Description
Runtime Environment	The name of the runtime environment where you want to run the tasks.
Authentication Type	The authentication method that the connector must use to login to the web application. Select one of the following authentication types: OAuth 2.0 Password Grant. Not Supported. OAuth 2.0 Client Certificate Grant. Requires you to select web API url, application id, tenant id, keystore file, keystore password, key alias, and key password.
Web API url	The URL of the Microsoft Dynamics 365 for Sales endpoint.
Username	The user name to connect to the Microsoft Dynamics 365 for Sales account.
Password	The password to connect to the Microsoft Dynamics 365 for Sales account.
Application ID	The Azure application ID for Microsoft Dynamics 365 for Sales.
Tenant ID	The directory ID for Azure Active Directory.
Keystore File	The location and the file name of the key store. Not applicable when you use the Hosted Agent.
Keystore Password	The password for the keystore file required for secure communication.
Key Alias	The alias name for the individual key.
Key Password	The password for the individual keys in the keystore file required for secure communication. Not applicable when you use the Hosted Agent.
Retry Error Codes	The comma-separated http error codes for which the retries are made.
Retry Count	The number of retries to get the response from an endpoint based on the retry interval. The default value is 5.
Retry Interval	The time in seconds to wait before Microsoft Dynamics 365 for Sales Connector retries for a response. The default value is 60 seconds.

PowerExchange for MongoDB JDBC Connections

When you configure a PowerExchange for MongoDB connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the MongoDB database.

The following table describes the MongoDB connection properties:

Property	Description
Connection Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
Type	The connection type. Select **MongoDB**.
Host Name	Host name or IP address of the MongoDB server.
Port	MongoDB server port number. Default is 9042.
User Name	User name to access the MongoDB server.
Password	Password corresponding to the user name to access the MongoDB server.
Database Name	Name of the MongoDB database to connect to.
Additional Connection Properties	Enter one or more JDBC connection parameters in the following format: `<param1>=<value>&<param2>=<value>&<param3>=<value>` You must provide the JDBC parameters as ampersand-separated key-value pairs. You can configure the following JDBC connection parameters in a MongoDB connection: - AuthSource - BatchSize - connectTimeoutMS - DefaultStringColumnLength - DmlBatchSize - EnableDoubleBuffer - EnableTransaction - LogLevel - LogPath - SamplingLimit - SamplingStepSize - SamplingStrategy - useJSONColumn For example, DefaultStringColumnLength=512&DmlBatchSize=1000& EnableDoubleBuffer=false&EnableTransaction=true& SamplingLimit=200&SamplingStepSize=2&SamplingStrategy=Backwards **Note:** If you specify the host name, port number, user name, and password of the MongoDB server in the **Additional Connection Properties**, the values specified in the **Additional Connection Properties** takes precedence.
SSL Mode	Options applicable for MongoDB Connector are: - **disabled** - **require** or **one-way**

Note: If you select the **Enable Reading/Writing as JSON** option, a column **documentAsJSON** appears in the collection when you read data from MongoDB through which you can read data as JSON. Default is disabled. To enable reading or writing as JSON, select **useJSONColumn**=true.

PowerExchange for MSMQ Connections

Use a queue connection object for each MSMQ source or target that you want to access.

The following table describes the properties that you configure for an MSMQ application connection:

Property	Description
Queue Name	Name of the MSMQ queue.
Machine Name	Name of the MSMQ machine. If MSMQ is running on the same machine as the Integration Service, you can enter a period (.).
Queue Type	Select public if the MSMQ queue is a public queue. Select private if the MSMQ queue is a private queue.
Is Transactional	Define whether the MSMQ queue is transactional or not. When a session writes to a remote private queue, the Integration Service cannot determine whether the queue is transactional or not. Configure the Is Transactional attribute to match the queue configuration. Choose one of the following options: - Auto. The Integration Service determines if the queue is transactional or not transactional. Choose Auto for a local queue or a remote queue that is not private. - Yes. The queue is transactional. - No. The queue is not transactional. Default is Auto. If you configure this property incorrectly, the session will not fail, but the target queue will not persist the data.

PowerExchange for Netezza Connections

Use a relational connection object for each Netezza source or target that you want to access.

The relational database connection defines how the Integration Service accesses the underlying database for Netezza Performance Server. When you configure a Netezza connection, you specify the connection attributes that the Integration Service uses to connect to Netezza.

The following table describes the properties that you configure for a Netezza connection:

Property	Description
User Name	Database user name with the appropriate read and write database permissions to access Netezza Performance Server.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name.
Connect String	ODBC data source to connect to Netezza Performance Server.

Property	Description
Code Page	Code page associated with Netezza Performance Server.
Connection Environment SQL	Runs an SQL command with each database connection. Default is disabled.
Transaction Environment SQL	Runs an SQL command before the initiation of each transaction. Default is disabled.
Connection Retry Period	Number of seconds the Integration Service attempts to reconnect to the database if the connection fails. If the Integration Service cannot connect to the database in the retry period, the session fails. Default value is 0.

PowerExchange for Oracle E-Business Suite Connection Properties

When you configure an Oracle E-Business Suite connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Oracle E-Business Suite.

The following table describes the connection properties:

Property	Description
User Name	User name to connect to Oracle E-Business Suite. If you are configuring a connection for an E-Business Suite target, the user name must be apps for the PowerCenter Integration Service to be able to execute the concurrent program.
Password	Password for the user name. You cannot use a parameter to specify the password.
Connect String	ODBC data source name. Use Informatica-certified ODBC drivers for ODBC data source connections.
Apps Schema Name	Name of the application schema that contains metadata for Oracle E-Business Suite. Default is apps.

PowerExchange for PeopleSoft Connections

Use an application connection object for each PeopleSoft source that you want to access. The application connection defines how the Integration Service accesses the underlying database for the PeopleSoft system.

The following table describes the properties that you configure for a PeopleSoft application connection:

Property	Description
Name	Name you want to use for this connection.
User Name	Database user name with SELECT permission on physical database tables in the PeopleSoft source system. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.

Property	Description
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. Must be in US-ASCII.
Connect String	Connect string for the underlying database of the PeopleSoft system. This option appears for DB2, Oracle, and Informix.
Code Page	Code page the Integration Service uses to extract data from the source database. When using relaxed code page validation, select compatible code pages for the source and target data to prevent data inconsistencies.
Language Code	PeopleSoft language code. Enter a language code for language-sensitive data. When you enter a language code, the Integration Service extracts language-sensitive data from related language tables. If no data exists for the language code, the PowerCenter extracts data from the base table. When you do not enter a language code, the Integration Service extracts all data from the base table.
Database Name	Name of the underlying database of the PeopleSoft system. This option appears for Sybase ASE and Microsoft SQL Server.
Server Name	Name of the server for the underlying database of the PeopleSoft system. This option appears for Sybase ASE and Microsoft SQL Server.
Domain Name	Domain name for Microsoft SQL Server on Windows.
Packet Size	Packet size used to transmit data. This option appears for Sybase ASE and Microsoft SQL Server.
Use Trusted Connection	If selected, the Integration Service uses Windows authentication to access the Microsoft SQL Server database. The user name that enables the Integration Service must be a valid Windows user with access to the Microsoft SQL Server database. This option appears for Microsoft SQL Server.
Rollback Segment	Name of the rollback segment for the underlying database of the PeopleSoft system. This option appears for Oracle.
Environment SQL	SQL commands used to set the environment for the underlying database of the PeopleSoft system.

PowerExchange for PostgreSQL Connection Properties

When you configure a PowerExchange for PostgreSQL connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the PostgreSQL database.

The following table describes the PostgreSQL connection properties:

Connection Property	Description
Host Name	Host name of the PostgreSQL server to which you want to connect.
Port	Port number for the PostgreSQL server to which you want to connect. Default is 5432.
Database	The PostgresSQL database name.
User Name	User name to access the PostgreSQL database.
Password	Password for the PostgreSQL database user name.
Encryption Method	Determines whether the data exchanged between the PowerCenter Integration Service and the PostgreSQL database server is encrypted: Select one of the following encryption methods: - noEncryption. Establishes a connection without using SSL. Data is not encrypted. - SSL. Establishes a connection using SSL. Data is encrypted using SSL. If the PostgreSQL database server does not support SSL, the connection fails. - requestSSL. Attempts to establish a connection using SSL. If the PostgreSQL database server does not support SSL, the PowerCenter Integration Service establishes an unencrypted connection. Default is noEncryption.
Validate Server Certificate	Applicable if you enable the encryption method to SSL or requestSSL. Select the **Validate Server Certificate** option so that the PowerCenter Integration Service validates the server certificate that is sent by the PostgreSQL database server. If you specify the **Hostname In Certificate** parameter, the PowerCenter Integration Service also validates the host name in the certificate.
TrustStore	Applicable if you select SSL or requestSSL as the encryption method and the Validate Server Certificate option. The path and name of the truststore file, which contains the list of the Certificate Authorities (CAs) that the PostgreSQL client trusts.
TrustStore Password	Applicable if you select SSL or requestSSL as the encryption method and the Validate Server Certificate option. The password to access the truststore file that contains the SSL certificate.
Host Name In Certificate	Optional when you select SSL or requestSSL as the encryption method and the Validate Server Certificate option. Specifying a host name ensures additional security and the PowerCenter Integration Service validates the host name included in the connection with the host name in the SSL certificate.

Connection Property	Description
KeyStore	Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server. The path and the file name of the key store. The keystore file contains the certificates that the PostgreSQL client sends to the PostgreSQL server in response to the server's certificate request.
KeyStore Password	Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server. The password for the keystore file required for secure communication.
Key Password	Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server. Required when individual keys in the keystore file have a different password than the keystore file.
Additional Connection Properties	Additional connection parameters that you want to use. You must provide the connection parameters as semicolon-separated key-value pairs. For example, `ConnectionRetryCount=2; ConnectionRetryDelay=5`
Crypto Protocol Versions	Required if you enable the encryption method to SSL or requestSSL. Specifies a cryptographic protocol or a list of cryptographic protocols when you use an encrypted connection. You can select from the following protocols: - SSLv3 - TLSv1 - TLSv1_1 - TLSv1_2

PowerExchange for Salesforce Analytics Connections

Use an application connection object for each Salesforce Analytics source or target that you want to access.

The following table describes the connection attributes for a Salesforce Analytics application connection:

Attribute Name	Description
Username	Salesforce Analytics user name for the application connection.
Password	Password for the Salesforce Analytics user name. The password is case sensitive.
Security Token	The token used to login to Salesforce Analytics from an untrusted network.
Service URL	URL of the Salesforce Analytics service that you want to access. In a test or development environment, you might want to access the Salesforce Analytics Sandbox testing environment. For more information about the Salesforce Analytics Sandbox, see the Salesforce documentation.

Attribute Name	Description
Temp Folder Name	The directory where the JSON files are stored.
Default Date Format	The date format to read date columns in the JSON file. Use the hyphen (-) delimiter for the Windows platform, and the forward slash (/) delimiter for the Linux platform.

PowerExchange for Salesforce Connections

Use an application connection object for each Salesforce source, target, or lookup that you want to access.

The following table describes the connection attributes for a Salesforce application connection:

Connection Property	Description
Name	Name of the Salesforce connection.
Type	Standard Salesforce connection.
User Name	User name for the Salesforce account.
Password	Password for the Salesforce account.

Service URL	URL of the Salesforce service.
	`https://login.salesforce.com/services/Soap/u/45.0`
	URL of the Salesforce service while using Salesforce Picklist, Merge, and Lookup transformation.
	`https://login.salesforce.com/services/Soap/u/33.0`

You can also create an OAuth type connection to access to Salesforce using the Salesforce API. OAuth is a standard protocol that allows for secure API authorization. A benefit of OAuth is that users do not need to disclose their Salesforce credentials and the Salesforce administrator can revoke the consumer's access at any time.

The following table lists the properties for an OAuth connection:

Connection Property	Description
Name	Name of the OAuth connection.
Type	Select the Use OAuth checkbox to use the OAuth connection.
Refresh Token	The Refresh Token of Salesforce.
Consumer Key	The Consumer Key obtained from Salesforce, required to generate the Refresh Token.
Consumer Secret	The Consumer Secret obtained from Salesforce, required to generate the Refresh Token.

PowerExchange for SAP NetWeaver Connections

Depending on the method of integration with mySAP applications, configure the following types of connections:

- **SAP R/3 application connection.** Configure SAP R/3 application connections to access the SAP system when you run an RFC stream or file mode session.
- **SAPTableReader application connection.** Configure SAPTableReader application connections to read data from SAP tables and ABAP CDS views through ABAP by using the HTTP/HTTPS protocol.
- **FTP connection.** Configure FTP connections to access the staging file through FTP. When you run a file mode session, you can configure the session to access the staging file on the SAP system through FTP.
- **SAP_ALE _IDoc_Reader and SAP_ALE _IDoc_Writer application connection.** Configure SAP_ALE_IDoc_Reader application connections to receive IDocs and business content integration documents using ALE. Configure SAP_ALE_IDoc_Writer application connections to send IDocs using ALE.
- **SAP RFC/BAPI interface application connection.** Configure SAP RFC/BAPI Interface application connections if you want to process data in SAP using BAPI/RFC transformations.

The following table describes the type of connection you need depending on the method of integration with mySAP applications:

Connection Type	Integration Method
SAP R/3 application connection	ABAP integration with RFC stream and RFC file mode sessions.
SAPTableReader application connection	ABAP integration with HTTP stream mode sessions.
FTP connection	ABAP integration with file mode sessions.
SAP_ALE _IDoc_Reader application connection	IDoc ALE and business content integration.
SAP_ALE _IDoc_Writer application connection	IDoc ALE and business content integration.
BCI Metadata Connection	IDoc ALE and business content integration for segments in SAP longer than 1,000 characters.
SAP RFC/BAPI interface application connection	BAPI/RFC integration.

SAP R/3 Application Connection for ABAP Integration

The application connections for SAP sources use one of the following connections:

File Mode

Use an RFC file mode connection when you extract data through file mode. The connection information for RFC is stored in the `sapnwrfc.ini` file. You must also have authorizations on the SAP system to read SAP tables and to run file mode sessions.

Stream Mode (RFC/HTTP)

In stream mode, you can use an SAP R/3 application connection.

To extract data through stream mode by using the RFC protocol, use an SAP R/3 application connection The connection information for RFC is stored in the `sapnwrfc.ini` file. You must also have authorizations on the SAP system to read SAP tables and to run stream mode sessions. RFC stream mode sessions use foreground processing.

You cannot use an SAP R/3 application connection to extract data through stream mode by using the HTTP protocol. Use an SAPTableReader application connection to extract data through stream mode by using the HTTP protocol.

Application Connection for an RFC Stream Mode Session

When you configure an application connection for an RFC stream mode session, the connect string you use in the SAP R/3 application connection must match the connect string in the `sapnwrfc.ini` file. For example, if the connect string in the `sapnwrfc.ini` file is defined in lowercase, use lowercase to enter the connect string parameter in the application connection configuration.

Application Connection for Stream and File Mode Sessions

You can create separate application connections for file and stream mode, or you can create one connection for both file and stream mode. Create separate entries if the SAP administrator creates separate authorization profiles.

To create one connection for both modes, the SAP administrator must have created a single profile with authorizations for both file and stream mode sessions.

The following table describes the properties that you configure for an SAP ECC connection:

Property	Values for RFC File Mode and RFC Stream Mode
Name	Connection name used by the Workflow Manager.
User Name	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the SAP user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the SAP user name.
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for an SAP load balancing connection.
Code Page	Code page compatible with the SAP server. The code page must correspond to the Language Code.

Property	Values for RFC File Mode and RFC Stream Mode
Client Code	SAP client number.
Language Code	Language code that corresponds to the SAP language.

Application Connection for HTTP Stream Mode Sessions

To read data from SAP tables and ABAP CDS views through ABAP by using the HTTP protocol, create an SAPTableReader connection.

The following table describes the properties that you configure for an SAPTableReader connection:

Property	Values for HTTP Stream Mode
Port Range	HTTP port range that the PowerCenter Integration Service must use to read data from the SAP server in streaming mode. Enter the minimum and maximum port numbers with a hyphen as the separator. The minimum and maximum port number can range between 10000 and 65535. You can also specify the port range according to your organization. Default is 10000-65535.
Use HTTPS	Enables you to read data from SAP tables and ABAP CDS views through HTTPS streaming. By default, the **Use HTTPS** check box is not selected.
Key store file path	Path to the keystore file that contains the private or public key pairs and the associated certificates. Required if you enable HTTPS.
Key store Password	Password for the keystore file. Required if you enable HTTPS.
Private Key Password	Password to decrypt the private key file. Required if you enable HTTPS, and the SAP server uses public key authentication and the private key is encrypted.

Application Connections for ALE Integration

To receive outbound IDocs and business content integration documents from SAP using ALE, create an SAP_ALE_IDoc_Reader application connection in the Workflow Manager. To send inbound IDocs to SAP using ALE, create an SAP_ALE_IDoc_Writer application connection in the Workflow Manager.

SAP_ALE_IDoc_Reader Application Connection

Configure the SAP_ALE_IDoc_Reader connection properties with the destination entry defined in the `sapnwrfc.ini` file for a connection to an RFC server program. Verify that the Program ID for this destination entry matches the Program ID for the logical system you defined in SAP to receive IDocs or consume business content data. For business content integration, set to INFACONTNT.

The following table describes the properties that you configure for an SAP_ALE_IDoc_Reader application connection:

Property	Description
Name	Connection name used by the Workflow Manager.
Code Page	Code page compatible with the SAP server.
Destination Entry	DEST entry defined in the `sapnwrfc.ini` file for a connection to an RFC server program. The Program ID for this destination entry must be the same as the Program ID for the logical system you defined in SAP to receive IDocs or consume business content data. For business content integration, set to INFACONTNT.

SAP_ALE_IDoc_Writer and BCI Metadata Application Connection

Configure the SAP_ALE_IDoc_Writer or the BCI Metadata Connection connection properties with the destination entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server.

The following table describes the properties that you configure for an SAP_ALE_IDoc_Writer or a BCI Metadata Connection application connection:

Property	Description
Name	Connection name used by the Workflow Manager.
User Name	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the SAP user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the SAP user name.
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server.
Code Page	Code page compatible with the SAP server. Must also correspond to the Language Code.
Language Code	Language code that corresponds to the SAP language.
Client Code	SAP client number.

Application Connection for BAPI/RFC Integration

To process BAPI/RFC data in SAP, you must create an application connection of the SAP RFC/BAPI Interface type in the Workflow Manager. The Integration Service uses this connection to connect to SAP and make BAPI/RFC function calls to extract, transform, or load data.

The following table describes the properties that you configure for an SAP RFC/BAPI application connection:

Property	Description
Name	Connection name used by the Workflow Manager.
User Name	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the password for the SAP user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the SAP user name. **Note:** If you want to run a session on 32-bit Linux, and you want to connect to SAP 4.60, enter the password in uppercase. The SAP system must also use uppercase passwords.
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server.
Code Page	Code page compatible with the SAP server. Must also correspond to the Language Code.
Language Code	Language code that corresponds to the SAP language.
Client Code	SAP client number.

PowerExchange for SAP NetWeaver BI Connections

You can configure the following types of connection objects to connect to SAP NetWeaver BI:

- SAP BW OHS application connection to connect to SAP NetWeaver BI sources.
- SAP BW application connection to connect to SAP NetWeaver BI targets.

SAP BW OHS Application Connection

Use an SAP BW OHS application connection object for each SAP NetWeaver BI source that you want to access.

The following table describes the properties that you configure for an SAP BW OHS application connection:

Property	Description
Name	Connection name used by the Workflow Manager.
User Name	SAP NetWeaver BI user name. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.

Property	Description
Use Parameter in Password	Indicates the SAP NetWeaver BI password is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	SAP NetWeaver BI password.
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server. The Integration Service uses the `sapnwrfc.ini` file to connect to the SAP NetWeaver BI system.
Code Page	Code page compatible with the SAP NetWeaver BI server.
Client Code	SAP NetWeaver BI client. Must match the client you use to log on to the SAP NetWeaver BI server.
Language Code	Language code that corresponds to the code page.

SAP BW Application Connection

Use an SAP BW application connection object for each SAP NetWeaver BI target that you want to access.

The following table describes the properties that you configure for an SAP BW application connection:

Property	Description
Name	Connection name used by the Workflow Manager.
User Name	SAP NetWeaver BI user name. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the SAP NetWeaver BI password is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	SAP NetWeaver BI password.
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server. The Integration Service uses the `sapnwrfc.ini` file to connect to the SAP NetWeaver BI system. If you do not enter a connection string, the Integration Service obtains the connection parameters from the SAP BW Service.
Code Page	Code page compatible with the SAP NetWeaver BI server.
Client Code	SAP NetWeaver BI client. Must match the client you use to log in to the SAP NetWeaver BI server.
Language Code	Language code that corresponds to the code page.

PowerExchange for Siebel Connections

You can configure the following types of connection objects to connect to Siebel:

- Siebel Application Connections for Sources, Targets, and EIM Invoker Transformations
- Siebel Application Connection for EIM Read and Load Transformations

Siebel Application Connections for Sources, Targets, and EIM Invoker Transformations

The Siebel Sources, Targets, and EIM Invoker transformations use the Siebel connection application connection to connect to the Siebel repository. When you configure an application connection, you must specify the connection attributes for the Siebel repository.

The following table describes the application connection properties:

Connection Parameter	Description
Protocol	Protocol used to connect to Siebel. Specify the following protocol parameters: - Transport. Enter HTTP or TCP/IP. Default is TCP/IP. - Encryption. Enter NONE or RSA. Default is NONE. - Compression. Enter NONE or ZLIB. Default is ZLIB. Specify the parameters in the following format: `siebel[[.transport][.[encryption][.[compression]]]]`
User Name	User name to connect to Siebel.
Password	Password for the user name.
Siebel Server Host	Host name or IP address of the Siebel server. If you configure native load balancing, specify the virtual host name.
SCBroker Port	Siebel Connection Broker port number.
Enterprise Server	Enterprise Server name.
Application Object Manager	Siebel application business object manager.
Encoding	Encoding defined in the code page the PowerCenter Integration Service uses to communicate with the Siebel Server. Default is UTF-8.

Siebel Application Connection for EIM Read and Load Transformations

The Siebel EIM Read and Load transformations use the Siebel Database Connection application connection to connect to the Siebel database. When you configure an application connection for the Siebel EIM Read or Load transformation, you must specify the connection attributes for the Siebel database.

The following table describes the Siebel EIM Read or Load transformations:

Connection Parameter	Description
User Name	User name to connect to Siebel.
Password	Password for the user name.
Connect String	ODBC data source name. **Note:** Use Informatica certified ODBC drivers for ODBC data source connection.
Connection Retry Period	Number of seconds the PowerCenter Integration Service attempts to reconnect to the database if the connection fails. If the PowerCenter Integration Service fails to connect to the database in the retry period, the session fails. If you set the connection retry period to 0, the PowerCenter Integration Service does not attempt to reconnect to the database if the connection fails. Default is 0.
Table Name Prefix	If required, configure the table name prefix to establish connection with the database. Default is Blank. **Note:** Enter the name of the Siebel database schema as the table name prefix when Oracle is the target database.

PowerExchange for Tableau Connections

The following table describes the properties for a Tableau connection:

Property	Description
Tableau Product	The name of the Tableau product to which you want to connect. You can choose one of the following Tableau products to publish the TDE or TWBX file: - Tableau Desktop. Creates a TDE file in the Data Integration Service machine. You can then manually import the TDE file to Tableau Desktop. **Note:** Tableau Desktop is not applicable for TWBX file. - Tableau Server. Publishes the generated TDE or TWBX file to Tableau Server. - Tableau Online. Publishes the generated TDE or TWBX file to Tableau Online.
Connection URL	URL of Tableau Server or Tableau Online to which you want to publish the TDE or TWBX file. The URL has the following format: `http://<Host name of Tableau Server or Tableau Online>:<port>`
User Name	User name of the Tableau Server or Tableau Online account.
Password	Password for the Tableau Server or Tableau Online account.
Content URL	The name of the site on Tableau Server or Tableau Online where you want to publish the TDE or TWBX file. Contact the Tableau administrator to provide the site name.
Template File Path	The path to a sample TDE file from where the Integration Service imports the Tableau metadata. Enter one of the following options for the template file path: - Absolute path to the TDE file. - Directory path for the TDE files. - Empty directory path. The path you specify for the template file becomes the default path for the target TDE file. If you do not specify a file path, the Integration Service uses the following default file path for the target TDE file: `<Data Integration Installation Directory>/main/java/lib`

PowerExchange for Tableau V3 Connections

When you set up a Tableau V3 connection, you must configure the connection properties.

The following table describes the Tableau V3 connection properties:

Property	Description
Name	Name of the Tableau V3 connection.
ID	String that the PowerCenter Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Description of the connection. The description cannot exceed 765 characters.
Location	The Informatica domain where you want to create the connection.
Type	Type of connection. Select Tableau V3.

The following table describes the properties to connect to Tableau:

Connection Property	Description
Tableau Product	The name of the Tableau product to which you want to connect. You can choose one of the following Tableau products to publish the `.hyper` or TWBX file: **Tableau Desktop** Creates a `.hyper` file in the PowerCenter Integration Service machine. You can then manually import the `.hyper` file to Tableau Desktop. **Tableau Server** Publishes the generated `.hyper` or TWBX file to Tableau Server. **Tableau Online** Publishes the generated `.hyper` or TWBX file to Tableau Online.
Connection URL	The URL of Tableau Server or Tableau Online to which you want to publish the `.hyper` or TWBX file. Enter the URL in the following format: `http://<Host name of Tableau Server or Tableau Online>:<port>`
User Name	The user name of the Tableau Server or Tableau Online account.
Password	The password for the Tableau Server or Tableau Online account.

Connection Property	Description
Site ID	The ID of the site on Tableau Server or Tableau Online where you want to publish the or TWBX file. **Note:** Contact the Tableau administrator to provide the site ID.
Schema File Path	The path to a sample `.hyper` file from where the PowerCenter Integration Service imports the Tableau metadata. Enter one of the following options for the schema file path: - Absolute path to the `.hyper` file. - Directory path for the `.hyper` files. - Empty directory path. The path you specify for the schema file becomes the default path for the target `.hyper` file. If you do not specify a file path, the PowerCenter Integration Service uses the following default file path for the target `.hyper` file: `<PowerCenter Integration Service installation directory>/apps/PowerCenter_Integration_Server/<latest version>/bin/rtdm`

PowerExchange for Teradata Parallel Transporter Connections

When you configure a Teradata Parallel Transporter API connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to Teradata.

The following table describes the Teradata Parallel Transporter API connection properties that you must configure:

Property	Description
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
User Name	Database user name with the appropriate read and write database permissions to access the database. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The PowerCenter Integration Service interprets user names that start with $Param as session parameters. You can connect to a database that runs on a network that uses Kerberos authentication. To configure Kerberos authentication for the database connection, set the user name to the reserved word *PmKerberosUser*. If you use Kerberos authentication, the connection uses the credentials of the user account that runs the session to connect to the database. The user account must have a user principal on the Kerberos network where the database runs.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, $Param*Name*. If you enable this option, define the password in the workflow or session parameter file, and encrypt it using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. If you set the user name to *PmKerberosUser* to use Kerberos authentication for the database connection, set the password to the reserved word *PmKerberosPassword*. The connection uses the credentials of the user account that runs the session to connect to the database.
Code Page	Code page associated with the database. When you run a session that extracts from a Teradata source, the code page of the Teradata PT API connection must be the same as the code page of the Teradata source.

The following table describes the Teradata connection properties that you must configure:

Attribute	Description
TDPID	The name of the Teradata database machine.
Database Name	Teradata database name. If you do not enter a database name, Teradata PT API uses the default login database name.
Tenacity	Amount of time, in hours, that Teradata PT API continues trying to log on when the maximum number of operations runs on the Teradata database. Must be a positive integer. Default is 4.
Max Sessions	Maximum number of sessions that Teradata PT API establishes with the Teradata database. Must be a positive, non-zero integer. Default is 4.
Min Sessions	Minimum number of Teradata PT API sessions required for the Teradata PT API job to continue. Must be a positive integer between 1 and the Max Sessions value. Default is 1.
Sleep	Amount of time, in minutes, that Teradata PT API pauses before it retries to log on when the maximum number of operations runs on the Teradata database. Must be a positive, non-zero integer. Default is 6.
Data Encryption	Enables full security encryption of SQL requests, responses, and data. Default is disabled.
Block Size	Maximum block size, in bytes, Teradata PT API uses when it returns data to the PowerCenter Integration Service. Minimum is 256. Maximum is 64,000. Default is 64,000.
Authentication Type	Method to authenticate the user. Select one of the following authentication types: - Native. Authenticates your user name and password against the Teradata database specified in the connection. - LDAP. Authenticates user credentials against the external LDAP directory service. - KRB5. Authenticates the credentials of the user account that runs the session against the Kerberos network where the database runs. Default is Native.
System Operator	Teradata PT API operator type: - Export. Extracts data from Teradata. Select Export if the session uses a Teradata Parallel Transporter Reader. - Load. Bulk loads data into an empty Teradata database table. - Update. Performs update, insert, upsert, and delete operations against Teradata database tables. - Stream. Performs update, insert, upsert, and delete operations against Teradata database tables in near real-time mode. Select Stream if you want to enable recovery for sessions that load data to Teradata. Default is Stream.

PowerExchange for TIBCO Connections

Use an application connection object for each TIBCO source or target that you want to access.

You can configure the following TIBCO application connection types:

- **TIB/Rendezvous.** Configure to read or write messages in TIB/Rendezvous format.
- **TIB/Adapter SDK.** Configure read or write messages in AE format.

Connection Properties for TIB/Rendezvous Application Connections

Use a TIB/Rendezvous application connection to read source messages or write target messages in TIB/Rendezvous format. When you configure a TIB/Rendezvous application connection, you specify connection properties for the Integration Service to connect to a TIBCO daemon.

The following table describes the properties you configure for a TIB/Rendezvous application connection:

Property	Description
Name	Name you want to use for this connection.
Code Page	Code page the Integration Service uses to extract data from the TIBCO. When using relaxed code page validation, select compatible code pages for the source and target data to prevent data inconsistencies.
Subject	Default subject for source and target messages. During a session, the Integration Service reads messages with this subject from TIBCO sources. It also writes messages with this subject to TIBCO targets. You can overwrite the default subject for TIBCO targets when you link the SendSubject port in a TIBCO target definition in a mapping.
Service	Service attribute value. Enter a value if you want to include a service name, service number, or port number.
Network	Network attribute value. Enter a value if your machine contains more than one network card.
Daemon	TIBCO daemon you want to connect to during a session. If you leave this option blank, the Integration Service connects to the local daemon during a session. If you want to specify a remote daemon, which resides on a different host than the Integration Service, enter the following values: `<remote hostname>:<port number>` For example, you can enter `host2:7501` to specify a remote daemon.
Certified	Select if you want the Integration Service to read or write certified messages.
CmName	Unique CM name for the CM transport when you choose certified messaging.
Relay Agent	Enter a relay agent when you choose certified messaging and the node running the Integration Service is not constantly connected to a network. The Relay Agent name must be fewer than 127 characters.
Ledger File	Enter a unique ledger file name when you want the Integration Service to read or write certified messages. The ledger file records the status of each certified message. Configure a file-based ledger when you want the TIBCO daemon to send unconfirmed certified messages to TIBCO targets. You also configure a file-based ledger with Request Old when you want the Integration Service to receive unconfirmed certified messages from TIBCO sources.
Synchronized Ledger	Select if you want PowerCenter to wait until it writes the status of each certified message to the ledger file before continuing message delivery or receipt.

Property	Description
Request Old	Select if you want the Integration Service to receive certified messages that it did not confirm with the source during a previous session run. When you select Request Old, you should also specify a file-based ledger for the Ledger File attribute.
User Certificate	Register the user certificate with a private key when you want to connect to a secure TIB/Rendezvous daemon during the session. The text of the user certificate must be in PEM encoding or PKCS #12 binary format.
Username	Enter a user name for the secure TIB/Rendezvous daemon.
Password	Enter a password for the secure TIB/Rendezvous daemon.

Connection Properties for TIB/Adapter SDK Connections

Use a TIB/Adapter SDK application connection to read source messages or write target messages in AE format. When you configure a TIB/Adapter SDK connection, you specify properties for the TIBCO adapter instance through which you want to connect to TIBCO.

Note: The adapter instances you specify in TIB/Adapter SDK connections should only contain one session.

The following table describes the connection properties you configure for a TIB/Adapter SDK application connection:

Property	Description
Name	Name you want to use for this connection.
Code Page	Code page the Integration Service uses to extract data from the TIBCO. When using relaxed code page validation, select compatible code pages for the source and target data to prevent data inconsistencies.
Subject	Default subject for source and target messages. During a workflow, the Integration Service reads messages with this subject from TIBCO sources. It also writes messages with this subject to TIBCO targets. You can overwrite the default subject for TIBCO targets when you link the SendSubject port in a TIBCO target definition in a mapping.
Application Name	Name of an adapter instance.
Repository URL	URL for the TIB/Repository instance you want to connect to. You can enter the server process variable $PMSourceFileDir for the Repository URL.
Configuration URL	URL for the adapter instance.
Session Name	Name of the TIBCO session associated with the adapter instance.
Validate Messages	Select Validate Messages when you want the Integration Service to read and write messages in AE format.

PowerExchange for Web Services Connections

Use a Web Services Consumer application connection for each web service source or target that you want to access. Use a Web Services Consumer application connection for each Web Services Consumer transformation as well. Web Services Consumer application connections allow you to control connection properties, including the endpoint URL and authentication parameters.

To connect to a web service, the Integration Service requires an endpoint URL. If you do not configure a Web Services Consumer application connection or if you configure one without providing an endpoint URL, the Integration Service uses the endpoint URL contained in the WSDL file on which the source, target, or Web Services Consumer transformation is based.

Use the following guidelines to determine when to configure a Web Services Consumer application connection:

- Configure a Web Services Consumer application connection with an endpoint URL if the web service you connect to requires authentication or if you want to use an endpoint URL that differs from the one contained in the WSDL file.
- Configure a Web Services Consumer application connection without an endpoint URL if the web service you connect to requires authentication but you want to use the endpoint URL contained in the WSDL file.
- You do not need to configure a Web Services Consumer application connection if the web service you connect to does not require authentication and you want to use the endpoint URL contained in the WSDL file.

If you need to configure SSL authentication, enter values for the SSL authentication-related properties in the Web Services Consumer application connection.

The following table describes the properties that you configure for a Web Services Consumer application connection:

Property	Description
User Name	User name that the web service requires. If the web service does not require a user name, enter PmNullUser. To define the user name in the parameter file, enter session parameter $Param*Name* as the user name, and define the value in the session or workflow parameter file. The Integration Service interprets user names that start with $Param as session parameters.
Use Parameter in Password	Indicates the web service password is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Password	Password that the web service requires. If the web service does not require a password, enter PmNullPasswd.
Code Page	Connection code page. The Repository Service uses the character set encoded in the repository code page when writing data to the repository.

Property	Description
End Point URL	Endpoint URL for the web service that you want to access. The WSDL file specifies this URL in the location element. You can use session parameter $Param*Name*, a mapping parameter, or a mapping variable as the endpoint URL. For example, you can use a session parameter, $ParamMyURL, as the endpoint URL, and set $ParamMyURL to the URL in the parameter file.
Domain	Domain for authentication.
Timeout	Number of seconds the Integration Service waits for a connection to the web service provider before it closes the connection and fails the session. Also, the number of seconds the Integration Service waits for a SOAP response after sending a SOAP request before it fails the session. Default is 60 seconds.
Trust Certificates File	File containing the bundle of trusted certificates that the Integration Service uses when authenticating the SSL certificate of the web services provider. Default is ca-bundle.crt.
Certificate File	Client certificate that a web service provider uses when authenticating a client. You specify the client certificate file if the web service provider needs to authenticate the Integration Service.
Certificate File Password	Password for the client certificate. You specify the certificate file password if the web service provider needs to authenticate the Integration Service.
Certificate File Type	File type of the client certificate. You specify the certificate file type if the web service provider needs to authenticate the Integration Service. The file type can be either PEM or DER.
Private Key File	Private key file for the client certificate. You specify the private key file if the web service provider needs to authenticate the Integration Service.
Key Password	Password for the private key of the client certificate. You specify the key password if the web service provider needs to authenticate the Integration Service.
Key File Type	File type of the private key of the client certificate. You specify the key file type if the web service provider needs to authenticate the Integration Service. PowerExchange for Web Services requires the PEM file type for SSL authentication.
Authentication Type	Select one of the following authentication types to use when the web service provider does not return an authentication type to the Integration Service: - Auto. The Integration Service attempts to determine the authentication type of the web service provider. - Basic. Based on a non-encrypted user name and password. - Digest. Based on a non-encrypted user name and encrypted password. - NTLM. Based on encrypted user name, password, and domain. Default is Auto.

PowerExchange for webMethods Connections

Use a webMethods application connection for each webMethods source and target that you want to access. Use a webMethods broker connection to read from webMethods source documents and write to webMethods target documents that do not have special characters. Use a webMethods Integration Server connection to read webMethods sources documents that have special characters.

Note: You cannot write to webMethods target documents that have special characters.

webMethods Broker Connection

The following table describes the properties that you configure for a webMethods broker connection:

Property	Description
Name	Name you want to use for this connection.
Broker Host	Enter the host name of the Broker you want the PowerCenter Integration Service to connect to. If the port number for the Broker is not the default port number, also enter the port number. Default port number is 6849. Enter the host name and port number in the following format: `<host name:port>`
Broker Name	Enter the name of the Broker. If you do not enter a Broker name, the PowerCenter Integration Service uses the default Broker.
Client ID	Enter a client ID for the PowerCenter Integration Service to use when it connects to the Broker during the session. If you do not enter a client ID, the Broker generates a random client ID. If you select Preserve Client State, enter a client ID.
Client Group	Enter the name of the group to which the client belongs.
Application Name	Enter the name of the application that will run the Broker Client.
Automatic Reconnection	Select this option to enable the PowerCenter Integration Service to reconnect to the Broker if the connection to the Broker is lost.
Preserve Client State	Select this option to maintain the client state across sessions. The client state is the information the Broker keeps about the client, such as the client ID, application name, and client group. Preserving the client state enables the webMethods Broker to retain documents it sends when a subscribing client application, such as the PowerCenter Integration Service, is not listening for documents. Preserving the client state also allows the Broker to maintain the publication ID sequence across sessions when writing documents to webMethods targets. If you select this option, configure a Client ID in the application connection. You should also configure guaranteed storage for your webMethods Broker. If you do not select this option, the PowerCenter Integration Service destroys the client state when it disconnects from the Broker.

webMethods Integration Server Connection

The following table describes the properties that you configure for a webMethods Integration Server connection:

Property	Description
Name	Name for the connection.
User Name	User name of a user with read access in the webMethods Integration Server.
Password	Password for the user name.
Use Parameter in Password	Enables the PowerCenter Integration Service to parameterize the password. Password for the webMethods Integration Server user name is a session parameter, $ParamName. Define the password in the workflow or session parameter file, and encrypt it by using the pmpasswd CRYPT_DATA option. Default is disabled.
IS Host	Host name and port number of the webMethods Integration Server in the following format: `<host name:port>`
Certificate Files	Client certificate that the webMethods Integration Server uses to authenticate a client. Specify the client certificate file if the webMethods Integration Server is configured as HTTPS. Use a semicolon (;) to separate multiple certificate files.
Certificate File Type	The file type of the client certificate. You specify the certificate file type if the webMethods Integration Server needs to authenticate the Integration Service. Supported file type is DER.
Private Key File	Private key file for the client certificate. Specify the private key file if the webMethods Integration Server is configured as HTTPS.
Key File Type	File type of the private key of the client certificate. You specify the key file type if the webMethods Integration Server is configured as HTTPS. Supported file type is DER.

PowerExchange for WebSphere MQ Connections

Use a Message Queue queue connection for each webMethods queue that you want to access.

Before you use PowerExchange for WebSphere MQ to extract data from message queues or load data to message queues, you can test the queue connections configured in the Workflow Manager.

The following table describes the properties that you configure for a Message Queue queue connection:

Property	Description
Name	Name you want to use for this connection.
Code Page	Code page that is the same as or a a subset of the code page of the queue manager coded character set identifier (CCSID).
Queue Manager	Name of the queue manager for the message queue.
Queue Name	Name of the message queue.

Property	Description
Connection Retry Period	Number of seconds the Integration Service attempts to reconnect to the WebSphere MQ queue if the connection fails. If the Integration Service cannot reconnect to the WebSphere MQ queue in the retry period, the session fails. Default is 0.
Recovery Queue Name	Name of the recovery queue. The recovery queue enables message recovery for a session that writes to a queue target.

Testing a Queue Connection on Windows

1. From the command prompt of the WebSphere MQ server machine, go to the <WebSphere MQ>\bin directory.
2. Use one of the following commands to test the connection for the queue:
   - **amqsputc.** Use if you installed the WebSphere MQ client on the Integration Service node.
   - **amqsput.** Use if you installed the WebSphere MQ server on the Integration Service node.

   The *amqsputc* and *amqsput* commands put a new message on the queue. If you test the connection to a queue in a production environment, terminate the command to avoid writing a message to a production queue.

   For example, to test the connection to the queue "production," which is administered by the queue manager "QM_s153664.informatica.com," enter one of the following commands:

   ```
   amqsputc production QM_s153664.informatica.com
   amqsput production QM_s153664.informatica.com
   ```

   If the connection is valid, the command returns a connection acknowledgment. If the connection is not valid, it returns an WebSphere MQ error message.
3. If the connection is successful, press Ctrl+C at the prompt to terminate the connection and the command.

Testing a Queue Connection on UNIX

1. On the WebSphere MQ Server system, go to the <WebSphere MQ>/samp/bin directory.
2. Use one of the following commands to test the connection for the queue:
   - **amqsputc.** Use if you installed the WebSphere MQ client on the Integration Service node.
   - **amqsput.** Use if you installed the WebSphere MQ server on the Integration Service node.

   The *amqsputc* and *amqsput* commands put a new message on the queue. If you test the connection to a queue in a production environment, make sure you terminate the command to avoid writing a message to a production queue.

   For example, to test the connection to the queue "production," which is administered by the queue manager "QM_s153664.informatica.com," enter one of the following commands:

   ```
   amqsputc production QM_s153664.informatica.com
   amqsput production QM_s153664.informatica.com
   ```

   If the connection is valid, the command returns a connection acknowledgment. If the connection is not valid, it returns an WebSphere MQ error message.

3. If the connection is successful, press Ctrl+C at the prompt to terminate the connection and the command.

Connection Object Management

You can create, edit, and delete connection objects.

Creating a Connection Object

1. In the Workflow Manager, click Connections and select the type of connection you want to create.

   The Connection Browser dialog box appears, listing all the source and target connections available for the selected connection type.

2. Click New.

   If you selected FTP as the connection type, the Connection Object dialog box appears. Go to step 5.

   If you selected Relational, Queue, Application, or Loader connection type, the Select Subtype dialog box appears.

3. In the Select Subtype dialog box, select the type of database connection you want to create.
4. Click OK.
5. Enter the properties for the type of connection object you want to create.

   The Connection Object Definition dialog box displays different properties depending on the type of connection object you create. For more information about connection object properties, see the section for each specific connection type in this chapter.

6. Click OK.

   The database connection appears in the Connection Browser list.

7. To add more database connections, repeat steps 2 through 6.
8. Click OK to save all changes.

Editing a Connection Object

You can change connection information at any time. When you edit a connection object, the Integration Service uses the updated connection information the next time the session runs.

1. Open the Connection Browser dialog box for the connection object. For example, click Connections > Relational to open the Connection Browser dialog box for a relational database connection.
2. Click Edit.

   The Connection Object Definition dialog box appears.

3. Enter the values for the properties you want to modify.

   The connection properties vary depending on the type of connection you select. For more information about connection properties, see the section for each specific connection type in this chapter.

4. Click OK.

Deleting a Connection Object

When you delete a connection object, the Workflow Manager invalidates all sessions that use these connections. To make a session valid, you must edit it and replace the missing connection.

1. Open the Connection Browser dialog box for the connection object. For example, click Connections > Relational to open the Connection Browser dialog box for a relational database connection.
2. Select the connection object you want to delete in the Connection Browser dialog box.

   Tip: Hold the shift key to select more than one connection to delete.
3. Click Delete, and then click Yes.

Validation

Workflow Validation

Before you can run a workflow, you must validate it. When you validate the workflow, you validate all task instances in the workflow, including nested worklets.

When you validate a workflow, you validate worklet instances, worklet objects, and all other nested worklets in the workflow. You validate task instances and worklets, regardless of whether you have edited them.

The Workflow Manager validates the worklet object using the same validation rules for workflows. The Workflow Manager validates the worklet instance by verifying attributes in the Parameter tab of the worklet instance.

If the workflow contains nested worklets, you can select a worklet to validate the worklet and all other worklets nested under it. To validate a worklet and its nested worklets, right-click the worklet and choose Validate.

The Workflow Manager validates the following properties:

- **Expressions.** Expressions in the workflow must be valid.
- **Tasks.** Non-reusable task and Reusable task instances in the workflow must follow validation rules.
- **Scheduler.** If the workflow uses a reusable scheduler, the Workflow Manager verifies that the scheduler exists. The Workflow Manager marks the workflow invalid if the scheduler you specify for the workflow does not exist in the folder.

The Workflow Manager also verifies that you linked each task properly.

Note: The Workflow Manager validates Session tasks separately. If a session is invalid, the workflow may still be valid.

Example

You have a workflow that contains a non-reusable worklet called Worklet_1. Worklet_1 contains a nested worklet called Worklet_a. The workflow also contains a reusable worklet instance called Worklet_2. Worklet_2 contains a nested worklet called Worklet_b.

The Workflow Manager validates links, conditions, and tasks in the workflow. The Workflow Manager validates all tasks in the workflow, including tasks in Worklet_1, Worklet_2, Worklet_a, and Worklet_b.

You can validate a part of the workflow. Right-click Worklet_1 and choose Validate. The Workflow Manager validates all tasks in Worklet_1 and Worklet_a.

Validating Multiple Workflows

You can validate multiple workflows or worklets without fetching them into the workspace. To validate multiple workflows, you must select and validate the workflows from a query results view or a view dependencies list. When you validate multiple workflows, the validation does not include sessions, nested worklets, or reusable worklet objects in the workflows. You can save and optionally check in workflows that change from invalid to valid status.

To validate multiple workflows:

1. Select workflows from either a query list or a view dependencies list.
2. Check out the objects you want to validate.
3. Right-click one of the selected workflows and choose Validate.

   The Validate Objects dialog box appears.
4. Choose to save objects and check in objects that you validate.

Worklet Validation

The Workflow Manager validates worklets when you save the worklet in the Worklet Designer. In addition, when you use worklets in a workflow, the Integration Service validates the workflow according to the following validation rules at run time:

- If the parent workflow is configured to run concurrently, each worklet instance in the workflow must be configured to run concurrently.
- Each worklet instance in the workflow can run once.

When a worklet instance is invalid, the workflow using the worklet instance remains valid.

The Workflow Manager displays a red invalid icon if the worklet object is invalid. The Workflow Manager validates the worklet object using the same validation rules for workflows. The Workflow Manager displays a blue invalid icon if the worklet instance in the workflow is invalid. The worklet instance may be invalid when any of the following conditions occurs:

- The parent workflow or worklet variable you assign to the user-defined worklet variable does not have a matching datatype.
- The user-defined worklet variable you used in the worklet properties does not exist.
- You do not specify the parent workflow or worklet variable you want to assign.

For non-reusable worklets, you may see both red and blue invalid icons displayed over the worklet icon in the Navigator.

Task Validation

The Workflow Manager validates each task in the workflow as you create it. When you save or validate the workflow, the Workflow Manager validates all tasks in the workflow except Session tasks. It marks the workflow as not valid if it detects that any task in the workflow is not valid.

The Workflow Manager verifies that attributes in the tasks follow validation rules. For example, the user-defined event you specify in an Event task must exist in the workflow. The Workflow Manager also verifies that you linked each task properly. For example, you must link the Start task to at least one task in the workflow.

When you delete a reusable task, the Workflow Manager removes the instance of the deleted task from each workflow that contains the task. The Workflow Manager also marks the workflow as not valid when you delete a reusable task that a workflow uses.

The Workflow Manager verifies that a folder does not contain duplicate task names, and it verfies that a workflow does not contain duplicate task instances.

You can validate reusable tasks in the Task Developer. Or, you can validate task instances in the Workflow Designer. When you validate a task, the Workflow Manager validates the task attributes and the links. For example, the user-defined event you specify in an Event tasks must exist in the workflow.

The Workflow Manager uses the following rules to validate tasks:

- **Assignment.** The Workflow Manager validates the expression that you enter for the Assignment task. For example, the Workflow Manager verifies that you assigned a matching datatype value to the workflow variable in the assignment expression.
- **Command.** The Workflow Manager does not validate the shell command you enter for the Command task.
- **Event-Wait.** If you choose to wait for a predefined event, the Workflow Manager verifies that you specified a file to watch. If you choose to use the Event-Wait task to wait for a user-defined event, the Workflow Manager verifies that you specified an event.
- **Event-Raise.** The Workflow Manager verifies that you specified a user-defined event for the Event-Raise task.
- **Human Task.** The Workflow Manager verifies that a Human task has a potential owner. The task must also have a business administrator and an escalation user. The Workflow Manager verifies that a task notification has a recipient. It also verifies that the Human task receives the results of a mapping task in the workflow.
- **Timer.** The Workflow Manager verifies that the variable you specified for the Absolute Time setting has the Date/Time datatype.
- **Start.** The Workflow Manager verifies that you linked the Start task to at least one task in the workflow.

When a task instance is not valid, the workflow running the task instance becomes not valid. When a reusable task is not valid, it does not affect the validity of the task instance in the workflow. However, if a Session task instance is not valid, the workflow might still be valid. The Workflow Manager validates sessions differently.

To validate a task, select the task in the workspace and click **Tasks** > **Validate**. Or, right-click the task in the workspace and choose **Validate**.

Session Validation

The Workflow Manager validates a Session task when you save it. You can also manually validate Session tasks and session instances. Validate reusable Session tasks in the Task Developer. Validate non-reusable sessions and reusable session instances in the Workflow Designer.

The Workflow Manager marks a reusable session or session instance invalid if you perform one of the following tasks:

- Edit the mapping in a way that might invalidate the session. You can edit the mapping used by a session at any time. When you edit and save a mapping, the repository might invalidate sessions that already use the mapping. The Integration Service does not run invalid sessions.

  You must reconnect to the folder to see the effect of mapping changes on Session tasks.

  When you edit a session based on an invalid mapping, the Workflow Manager displays a warning message:

  ```
  The mapping [mapping_name] associated with the session [session_name] is invalid.
  ```
- Delete a database, FTP, or external loader connection used by the session.
- Leave session attributes blank. For example, the session is invalid if you do not specify the source file name.
- Change the code page of a session database connection to an incompatible code page.

If you delete objects associated with a Session task such as session configuration object, Email, or Command task, the Workflow Manager marks a reusable session invalid. However, the Workflow Manager does not mark a non-reusable session invalid if you delete an object associated with the session.

If you delete a shortcut to a source or target from the mapping, the Workflow Manager does not mark the session invalid.

The Workflow Manager does not validate SQL overrides or filter conditions entered in the session properties when you validate a session. You must validate SQL override and filter conditions in the SQL Editor.

If a reusable session task is invalid, the Workflow Manager displays an invalid icon over the session task in the Navigator and in the Task Developer workspace. This does not affect the validity of the session instance and the workflows using the session instance.

If a reusable or non-reusable session instance is invalid, the Workflow Manager marks it invalid in the Navigator and in the Workflow Designer workspace. Workflows using the session instance remain valid.

To validate a session, select the session in the workspace and click Tasks > Validate. Or, right-click the session instance in the workspace and choose Validate.

Related Topics:

- "Editing a Session" on page 37
- "Session Properties Reference" on page 242

Validating Multiple Sessions

You can validate multiple sessions without fetching them into the workspace. You must select and validate the sessions from a query results view or a view dependencies list. You can save and optionally check in sessions that change from invalid to valid status.

Note: If you use the Repository Manager, you can select and validate multiple sessions from the Navigator.

To validate multiple sessions:

1. Select sessions from either a query list or a view dependencies list.
2. Right-click one of the selected sessions and choose Validate.

   The Validate Objects dialog box appears.
3. Choose whether to save objects and check in objects that you validate.

Expression Validation

The Workflow Manager validates all expressions in the workflow. You can enter expressions in the Assignment task, Decision task, and link conditions. The Workflow Manager writes any error message to the Output window.

Expressions in link conditions and Decision task conditions must evaluate to a numerical value. Workflow variables used in expressions must exist in the workflow.

The Workflow Manager marks the workflow invalid if a link condition is invalid.

Scheduling and Running Workflows

Workflow Schedulers

Each workflow has an associated scheduler. A workflow scheduler is a repository object that contains a set of schedule settings. It contains information about how and when to run a workflow.

You can schedule a workflow to run continuously, repeat at a specified time or interval, or you can manually start a workflow. By default, workflows run on demand. You can create a non-reusable scheduler for an individual workflow. Or, you can create a reusable scheduler to use the same schedule settings for all workflows in a folder.

If you configure multiple instances of a workflow, and you schedule the workflow run time, the Integration Service runs all instances at the scheduled time. You cannot schedule workflow instances to run at different times.

Scheduling for Time Zones and Time Shifts

Consider the following rules and guidelines for workflow scheduling around time zones and time shifts:

- On Windows, the Integration Service does not run a scheduled workflow during the last hour of Daylight Saving Time (DST). If a workflow is scheduled to run between 1:00 a.m. and 1:59 a.m. DST, the Integration Service resumes the workflow after 1:00 a.m. Standard Time (ST). If you try to schedule a workflow during the last hour of DST or the first hour of ST, you receive an error. Wait until 2:00 a.m. to create a scheduler.
- The Integration Service schedules the workflow in the time zone of the Integration Service node. For example, the PowerCenter Client is in the local time zone and the Integration Service is in a time zone two hours later. If you schedule the workflow to start at 9:00 a.m., it starts at 9:00 a.m. in the time zone of the Integration Service node and 7:00 a.m. local time.

Scheduling with Team-based Development

If you have team-based development, workflow schedules are updated based on the type of scheduler:

Non-reusable scheduler

When you configure or edit a non-reusable scheduler, check in the workflow to allow the schedule to take effect. You can update the schedule manually with the workflow checked out. Note that the changes are applied to the latest checked-in version of the workflow.

Reusable scheduler

When you create a reusable scheduler for a workflow, you must check in the workflow and the scheduler to enable the schedule to take effect.

When you edit a reusable scheduler and check it in, workflows are updated with the latest schedule. Note that the workflow schedule is updated even for workflows that are checked out.

When you edit a reusable scheduler and do not check it in, you must manually update a workflow to update the workflow schedule. Note that the workflow schedule is updated only for workflows that are checked in.

Workflow Scheduler Properties

Configure the **Schedule** tab of the scheduler to set run options, schedule options, start options, and end options for the schedule.

You can configure the following options on the **Schedule** tab of the scheduler:

Run Options

Indicates the how to run the workflow. You can choose one of the following options:

- Run On Integration Service Initialization. The Integration Service runs the workflow as soon as the service is initialized. The Integration Service then starts the next run of the workflow according to settings in **Schedule Options**.
- Run On Demand. The Integration Service runs the workflow when you start the workflow manually.

- Run Continuously. The Integration Service runs the workflow as soon as the service initializes. The Integration Service then starts the next run of the workflow as soon as it finishes the previous run. If you edit a workflow that is set to run continuously, you must stop or unschedule the workflow, save the workflow, and then restart or reschedule the workflow.

Schedule Options

Indicates the type of schedule. Required if you select **Run On Integration Service Initialization**, or if you do not choose any setting in **Run Options**. You can choose one of the following options:

- Run Once. The Integration Service runs the workflow once, as scheduled in the scheduler.
- Run Every. The Integration Service runs the workflow at regular intervals, as configured.
- Customized Repeat. The Integration Service runs the workflow on the dates and times specified in the Repeat dialog box. When you choose **Customized Repeat**, you can schedule specific dates and times to run the workflow. The selected scheduler appears at the bottom of the page.

Start Options

Indicates when to start the workflow schedule. You can choose one of the following options:

- Start Date. The date that the Integration Service begins the workflow schedule.
- Start Time. The time when the Integration Service begins the workflow schedule.

End Options

Indicates when to end the workflow schedule. Required if the workflow schedule is **Run Every** or **Customized Repeat**. You can choose one of the following options:

- End On. The Integration Service stops scheduling the workflow on the selected date.
- End After. The Integration Service stops scheduling the workflow after the configured number of workflow runs.
- Forever. The Integration Service schedules the workflow as long as the workflow does not fail.

Repeat Options for Schedulers

If you choose the **Customized Repeat** schedule option, you can configure the repeat options for the workflow runs.

You can configure the following options:

Repeat Every

Enter the numeric interval you would like the Integration Service to schedule the workflow. You can choose one of the following frequencies:

- Days. Select the daily frequency settings.
- Weeks. Select the weekly and daily frequency settings.
- Months. Select the monthly, and daily frequency settings.

Weekly

Required to enter a weekly schedule. Select the day or days of the week on that you want to run the workflow.

Monthly

Required to enter a monthly schedule. You can choose one of the following options:

- Run On Day. Select the dates on which you want the workflow scheduled on a monthly basis. The Integration Service schedules the workflow to run on the selected dates. If you select a numeric date exceeding the number of days within a particular month, the Integration Service schedules the workflow for the last day of the month, including leap years. For example, if you schedule the workflow to run on the 31st of every month, the Integration Service schedules the session on the 30th of April, June, September, and November.
- Run On The. Select the week or weeks of the month, and then select the day of the week on which you want the workflow to run. For example, if you select Second and Last, and then select Wednesday, the Integration Service schedules the workflow to run on the second and last Wednesday of every month.

Daily Frequency

The number of times you want the workflow to run on any day the session is scheduled. Choose one of the following options:

- Run Once. The Integration Service runs the workflow one time on the selected day, at the time entered on the **Start Time** setting on the **Time** tab.
- Run Every. The Integration Service runs the workflow on the hour and minute interval that you configure. The Integration Service then schedules the workflow at regular intervals on the selected day. The Integration Service uses the **Start Time** setting for the first scheduled workflow of the day. If you choose an interval that is greater than the start time, the workflow runs one time each day. The Integration Service then schedules the workflow at regular intervals on the selected day.

Scheduled States

The scheduled state of a workflow includes historical run-time information such as the last time the workflow ran and how many times a repeating workflow has run. A workflow can get removed from the schedule based on changes to the workflow status or the Integration Service state.

When a workflow is removed from the schedule, the Integration Service either discards or maintains the scheduled state. If the Integration Service discards the scheduled state, it resets the state when the workflow is rescheduled. If the Integration Service maintains the scheduled state, it restores the state when the workflow is rescheduled.

When the Integration Service resets the scheduled state, it maintains the scheduler configuration. It does not check for missed schedules, and it schedules the workflow as though the workflow never ran. For example, you configure a workflow to run five times, and it stops during the second run. When you reschedule the workflow, the Integration Service resets the schedule to run five times.

The Integration Service can restore the scheduled state of a workflow in a highly available environment when it successfully recovers a terminated workflow or when you restart a workflow. When the Integration Service restores the scheduled state, it reschedules the workflow based on the scheduler configuration and the schedule frequency.

The Integration Service maintains or discards the scheduled state based on the following situations:

You disable a workflow.

When you enable a workflow, the Integration Service resets the schedule.

You remove a workflow from the schedule.

When you reschedule a workflow, the Integration Service resets the schedule.

You change the schedule settings.

The Integration Service reschedules the workflow according to the updated settings. If you change a schedule that is configured to run at repeated intervals, the Integration Service resets the frequency counter.

You copy a folder.

The Integration Service resets the schedule for all workflows in the folder.

You choose a different Integration Service to run a workflow.

The Integration Service resets the schedule for the workflow if it is unscheduled or is scheduled to run continuously but the start time has passed. You must reschedule the workflow if the start time is passed and the workflow is not scheduled to run continuously.

You recycle the Integration Service or enable it in normal mode.

The Integration Service resets the schedule for all workflows that are unscheduled or are scheduled to run continuously but the start time has passed. If a workflow is not configured to run on service initialization, you must reschedule it if the start time is passed and it is not scheduled to run continuously. If a workflow is configured to run on service initialization, you do not need to reschedule it.

You enable the Integration Service in safe mode.

In safe mode, workflows remain scheduled, but the Integration Service does not run them, including workflows that are scheduled to run continuously or run on service initialization.

A workflow becomes suspended.

A workflow can become suspended when you configure it to suspend on error. The Integration Service removes a suspended workflow from the schedule and it maintains the state of operation. You can recover a suspended workflow to restore the schedule.

A workflow fails.

To re-establish the schedule, you can reschedule the workflow. In a highly available domain, if you restart the workflow, and the workflow succeeds, the Integration Service restores the scheduled state and determines whether a scheduled run was missed.

A workflow stops or aborts.

To re-establish the schedule, you can recover or reschedule the workflow. If the domain is not highly available, the Integration Service resets the schedule. If the domain is highly available, the Integration Service restores the schedule. If you restart the workflow, and the workflow succeeds, the Integration Service restores the scheduled state and determines whether a scheduled run was missed.

A workflow terminates.

The Integration Service terminates all running workflows when it shuts down unexpectedly. If the domain is not highly available, the Integration Service resets the schedule when you reschedule the workflow. If the domain is highly available, and the workflow is recoverable, you can recover the workflow to restore the scheduled state. If the workflow is not recoverable, you can reset the schedule by rescheduling the workflow. If you restart the workflow, and the workflow succeeds, the Integration Service restores the scheduled state and determines whether a scheduled run was missed.

Important: If you manually start a failed, terminated, stopped, or aborted workflow in a highly available domain, Informatica recommends that you unschedule it first. If you do not unschedule the workflow, and the Integration Service detects that the scheduled run time was missed, it immediately runs the workflow again. This can result in errors such as key violations and invalid data. When you unschedule the workflow first and reschedule it after the manual run completes, the Integration Service does not run the workflow based on the missed schedule.

Restored State and Schedule Frequencies

In a highly available domain, the Integration Service restores the scheduled state of workflows based on the scheduled frequency.

The following scheduler configurations determine how the Integration Service restores the scheduled state:

The workflow is configured to run at a specific time.

When the Integration Service restores the scheduled state, it determines whether a scheduled run was missed. If the workflow did not miss a scheduled run, it runs at the next scheduled time. If the workflow missed a scheduled run, the Integration Service schedules it to run immediately.

The workflow is configured with a repeat frequency.

When the Integration Service restores the scheduled state, it determines how many more times the workflow is scheduled to run and begins the schedule at that point. It does not determine missed workflow runs. For example, you configure a workflow to run five times, and the workflow stops or aborts after it runs two times. When the Integration Service restores the schedule, the workflow runs three more times beginning with the next scheduled time.

The workflow is configured to run continuously.

The Integration Service restores the scheduled state and begins running the workflow immediately.

If you restart the Integration Service or choose a different Integration Service for a workflow, you must reschedule workflows that are not scheduled to run continuously. The Integration Service reschedules workflows that are scheduled to run continuously. The Integration Service also reschedules workflows in a folder if you copy the folder.

Scheduling a Workflow

You can schedule a workflow to run continuously, repeat at a given time or interval, or you can manually start a workflow.

1. In the Workflow Designer, open the workflow.

2. Click Workflows > Edit.
3. Click the Scheduler tab.
4. Select Non-reusable to create a non-reusable set of schedule settings for the workflow.

   -or-

   Select Reusable to select an existing reusable scheduler for the workflow.
5. Click the right side of the Scheduler field to edit scheduling settings for the scheduler.
6. If you select Reusable, choose a reusable scheduler from the Scheduler Browser dialog box.
7. Click OK.

To reschedule a workflow on its original schedule, right-click the workflow in the Navigator window and choose Schedule Workflow.

Creating a Reusable Scheduler

You can create a reusable scheduler in a folder to use the same set of scheduling settings for all workflows in the folder.

1. In the Workflow Designer, click **Workflow** > **Schedulers**.
2. Click **Add** to add a new scheduler.
3. On the **General** tab, enter a name for the scheduler.
4. Configure the scheduler settings in the **Scheduler** tab.

Note: When you delete a reusable scheduler, all workflows that use the deleted scheduler becomes invalid. To make the workflows valid, you must edit them and replace the missing scheduler

Unscheduling a Workflow

To remove a workflow from its schedule, right-click the workflow in the Navigator and choose Unschedule Workflow.

To permanently remove a workflow from a schedule, configure the workflow schedule to run on demand.

Note: When the Integration Service restarts, it reschedules all unscheduled workflows that are scheduled to run continuously.

Disabling a Workflow

You might want to disable the workflow while you edit it. When you disable a workflow, the Integration Service does not run the workflow until you enable it.

To disable a workflow select **Disable Workflows** on the General tab of the workflow properties.

Manual Workflow Runs

You can manually start a workflow configured to run on demand or to run on a schedule. Use the Workflow Manager, Workflow Monitor, or *pmcmd* to run a workflow. You can choose to run the entire workflow, part of a workflow, or a task in the workflow.

Before you can run a workflow, you must select an Integration Service to run the workflow. You can select an Integration Service when you edit a workflow or from the **Assign Integration Service** dialog box. If you select an Integration Service from the **Assign Integration Service** dialog box, the Workflow Manager overwrites the Integration Service assigned in the workflow properties.

You can also use advanced options to override the Integration Service or operating system profile assigned to the workflow and select concurrent workflow run instances.

Running an Entire Workflow

When you run an entire workflow, you can choose to run the workflow with the current configuration, or you can choose to override some options, such as the Integration Service or operating system profile.

1. Open the folder that contains the workflow.
2. From the Navigator, select the workflow that you want to start.
3. Right-click the workflow in the Navigator and choose the way that you want to start the workflow.
    - Start Workflow. When you choose to start the workflow, the Integration Service runs the workflow with the configured options.
    - Start Advanced Workflow. When you choose to start the workflow with advanced options, you can configure the following advanced options:
        - Integration Service. Overrides the Integration Service configured for the workflow.
        - Operating System Profile. Overrides the operating system profile assigned to the folder.
        - Workflow Run Instances. Choose the workflow instances that you want to run. This option appears if the workflow is configured for concurrent execution.

    The Integration Service runs the entire workflow.

    Note: You can also manually start a workflow from the Workflows menu or by right-clicking in the Workflow Designer workspace and choosing **Start Workflow**.

Running a Workflow with Advanced Options

Use the advanced options to override the Integration Service or operating system profile assigned to the workflow and select concurrent workflow run instances.

1. Open the folder containing the workflow.
2. From the Navigator, select the workflow that you want to start.
3. Right-click the workflow in the Navigator and click Start Workflow Advanced.

    The Start Workflow - Advanced options dialog box appears.

4. Configure the following options:

Advanced Options	Description
Integration Service	Overrides the Integration Service configured for the workflow.
Operating System Profile	Overrides the operating system profile assigned to the folder.
Workflow Run Instances	The workflow instances you want to run. Appears if the workflow is configured for concurrent execution.

5. Click OK.

Running Part of a Workflow

To run part of the workflow, right-click the task that you want the Integration Service to run and choose Start Workflow From Task. The Integration Service runs the workflow from the selected task to the end of the workflow.

1. Connect to the folder containing the workflow.
2. In the Navigator, drill down the Workflow node to show the tasks in the workflow.
3. Right-click the task for which you want the Integration Service to begin running the workflow.
4. Click Start Workflow From Task.

Running a Task in the Workflow

When you start a task in the workflow, the Workflow Manager locks the entire workflow so another user cannot start the workflow. The Integration Service runs the selected task. It does not run the rest of the workflow.

To run a task using the Workflow Manager, select the task in the Workflow Designer workspace. Right-click the task and choose Start Task.

You can also use menu commands in the Workflow Manager to start a task. In the Navigator, drill down the Workflow node to locate the task. Right-click the task you want to start and choose Start Task.

To start a task in a workflow from *pmcmd*, use the starttask command.

Sending Email

Sending Email Overview

You can send email to designated recipients when the Integration Service runs a workflow. For example, if you want to track how long a session takes to complete, you can configure the session to send an email containing the time and date the session starts and completes. Or, if you want the Integration Service to notify you when a workflow suspends, you can configure the workflow to send email when it suspends.

To send email when the Integration Service runs a workflow, perform the following steps:

- **Configure the Integration Service to send email.** Before creating Email tasks, configure the Integration Service to send email.

  If you use a grid or high availability in a Windows environment, you must use the same Microsoft Outlook profile on each node to ensure the Email task can succeed.
- **Create Email tasks.** Before you can configure a session or workflow to send email, you need to create an Email task.
- **Configure sessions to send post-session email.** You can configure the session to send an email when the session completes or fails. You create an Email task and use it for post-session email.

  When you configure the subject and body of post-session email, use email variables to include information about the session run, such as session name, status, and the total number of rows loaded. You can also use email variables to attach the session log or other files to email messages.
- **Configure workflows to send suspension email.** You can configure the workflow to send an email when the workflow suspends. You create an Email task and use it for suspension email.

The Integration Service sends the email based on the locale set for the Integration Service process running the session.

You can use parameters and variables in the email user name, subject, and text. For Email tasks and suspension email, you can use service, service process, workflow, and worklet variables. For post-session email, you can use any parameter or variable type that you can define in the parameter file. For example, you can use the $PMSuccessEmailUser or $PMFailureEmailUser service variable to specify the email recipient for post-session email.

Configuring Email on UNIX

The PowerCenter Integration Service can send email on AIX and Linux. The PowerCenter Integration Service uses rmail to send email on AIX. The PowerCenter Integration Service uses sendmail to send email on Linux. To send email, the user who starts Informatica services must have the rmail or sendmail tool installed in the path.

If you want to send email to more than one person, separate the email address entries with a comma. Do not put spaces between addresses.

Verifying rmail on AIX

Before you configure email in a session or workflow, verify that the rmail tool is accessible on AIX.

1. Log in to the UNIX system as the PowerCenter user who starts the Informatica services.
2. Type the following lines at the prompt and press Enter:

   ```
   rmail <your fully qualified email address>,<second fully qualified email address>
   From <your_user_name>
   ```
3. To indicate the end of the message, type `^D`.

   You should receive a blank email from the email account of the user you specify in the From line. If not, locate the directory where rmail resides and add that directory to the path.

Verifying sendmail on Linux

The PowerCenter Integration Service uses sendmail to send email on Linux. Before you configure email in a session or workflow, verify that the sendmail tool is accessible on the Linux machines.

1. Log in to the Linux machine as the PowerCenter user who starts the Informatica services.
2. Add `/usr/sbin` to the $PATH environment variable to send emails.
3. Type the following line at the prompt and press Enter:

   ```
   sendmail <your fully qualified email address>,<second fully qualified email
   address>
   ```
4. To indicate the end of the message, enter a period (.) on a separate line and press Enter. Or, type `^D`.

   You should receive a blank email from the email account of the PowerCenter user. If not, find the directory where sendmail resides and add that directory to the path.

Configuring MAPI on Windows

The Integration Service on Windows can send email using SMTP or MAPI. By default, the Integration Service uses Microsoft Outlook to send email using the MAPI interface.

To send email using MAPI on Windows, you must meet the following requirements:

- Install the Microsoft Outlook mail client on each node configured to run the Integration Service.
- Run Microsoft Outlook on a Microsoft Exchange Server.

Complete the following steps to configure the Integration Service on Windows to send email:

1. Configure a Microsoft Outlook profile.
2. Configure Logon network security.
3. Create distribution lists in the Personal Address Book in Microsoft Outlook.
4. Verify the Integration Service is configured to send email using the Microsoft Outlook profile you created in step 1.

The Integration Service on Windows sends email in MIME format. You can include characters in the subject and body that are not in 7-bit ASCII. For more information about the MIME format or the MIME decoding process, see the email documentation.

Step 1. Configure a Microsoft Outlook User

You must set up a profile for a Microsoft Outlook user before you can configure the Integration Service to send email. The user profile must have a Personal Address Book and a Microsoft Exchange Server.

Note: If you have high availability or if you use a grid, use the same profile for each node configured to run a service process.

To configure a Microsoft Outlook user:

1. Open the Control Panel on the node running the Integration Service process.
2. Double-click the Mail icon.
3. In the Mail Setup - Outlook dialog box, click Show Profiles.

   The Mail dialog box displays the list of profiles configured for the computer.

4. Click Add.
5. In the New Profile dialog box, enter a profile name. Click OK.

   The E-mail Accounts wizard appears.
6. Select Add a new e-mail account. Click Next.
7. Select Microsoft Exchange Server for the server type. Click Next.
8. Enter the Microsoft Exchange Server name and the mailbox name. Click Next.
9. Click Finish.
10. In the Mail dialog box, select the profile you added and click Properties.
11. In the Mail Setup dialog box, click E-mail Accounts.

    The E-mail Accounts wizard appears.
12. Select Add a new directory or address book. Click Next.
13. Select Additional Address Books. Click Next.
14. Select Personal Address Book. Click Next.
15. Enter the path to a personal address book. Click OK.
16. Click Close to close the Mail Setup dialog box.
17. Click OK to close the Mail dialog box.

Step 2. Configure Logon Network Security

You must configure the Logon Network Security before you run the Microsoft Exchange Server.

1. In Microsoft Outlook, click Tools > E-mail Accounts.

   The E-mail Accounts wizard appears.
2. Select View or change existing e-mail accounts. Click Next.
3. Select the Microsoft Exchange Server e-mail account. Click Change.
4. Click More Settings.

   The Microsoft Exchange Server dialog box appears.
5. Click the Security tab.
6. Set the Logon network security option to Kerberos/NTLM Password Authentication.
7. Click OK.

Step 3. Create Distribution Lists

When the Integration Service runs on Windows, you can enter one email address in the Workflow Manager. If you want to send email to multiple recipients, create a distribution list containing these addresses in the Personal Address Book in Microsoft Outlook. Enter the distribution list name as the recipient when configuring email.

For more information about working with a Personal Address Book, refer to Microsoft Outlook documentation.

Step 4. Verify the Integration Service Settings

After you create the Microsoft Outlook profile, verify the Integration Service is configured to send email as that Microsoft Outlook user. You may need to verify the profile with the domain administrator.

1. From the Administrator tool, click the Properties tab for the Integration Service.
2. In the Configuration Properties tab, select Edit.
3. In the MSExchangeProfile field, verify that the name of Microsoft Exchange profile matches the Microsoft Outlook profile you created.

Configuring SMTP on Windows

The Integration Service can send email using SMTP if authentication is disabled on the SMTP server. To configure the Integration Service to send email using SMTP on Windows, set the following custom properties:

Property	Description
SMTPServerAddress	The server address for the SMTP outbound mail server, for example, *powercenter.mycompany.com*.
SMTPPortNumber	The port number for the SMTP outbound mail server, for example, 25.
SMTPFromAddress	Email address the Service Manager uses to send email, for example, *PowerCenter@MyCompany.com*.
SMTPServerTimeout	Amount of time in seconds the Integration Service waits to connect to the SMTP server before it times out. Default is 20.

If you omit the SMTPServerAddress, SMTPPortNumber, or SMTPFromAddress custom property, the Integration Service sends email using the MAPI interface.

For more information about setting custom properties for the Integration Service, see the *PowerCenterAdministrator Guide*.

Working with Email Tasks

You can send email during a workflow using the Email task on the Workflow Manager. You can create reusable Email tasks in the Task Developer for any type of email. Or, you can create non-reusable Email tasks in the Workflow and Worklet Designer.

Use Email tasks in any of the following locations:

- **Session properties.** You can configure the session to send email when the session completes or fails.
- **Workflow properties.** You can configure the workflow to send email when the workflow is interrupted.
- **Workflows or worklets.** You can include an Email task anywhere in the workflow or worklet to send email based on a condition you define.

Using Email Tasks in a Workflow or Worklet

Use Email tasks anywhere in a workflow or worklet. For example, you might configure a workflow to send an email if a certain number of rows fail for a session.

For example, you may have a Session task in the workflow and you want the Integration Service to send an email if more than 20 rows are dropped. To do this, you create a condition in the link, and create a non-reusable Email task. The workflow sends an email if the session fails more than 20 rows are dropped.

Email Address Tips and Guidelines

Consider the following tips and guidelines when you enter the email address in an Email task:

- Enter the email address using 7-bit ASCII characters only.
- You can use service, service process, workflow, and worklet variables in the email address.
- You can send email to any valid email address. On Windows, the mail recipient does not have to have an entry in the Global Address book of the Microsoft Outlook profile.
- If the Integration Service is configured to send email using MAPI on Windows, you can send email to multiple recipients by creating a distribution list in the Personal Address book. All recipients must also be in the Global Address book. You cannot enter multiple addresses separated by commas or semicolons.
- If the Integration Service is configured to send email using SMTP on Windows, you can enter multiple email addresses separated by a semicolon.
- If the Integration Service runs on UNIX, you can enter multiple email addresses separated by a comma. Do not include spaces between email addresses.

Creating an Email Task

You can create Email tasks in the Task Developer, Worklet Designer, and Workflow Designer.

1. In the Task Developer, click Tasks > Create.

   The Create Task dialog box appears.
2. Select an Email task and enter a name for the task. Click Create.

   The Workflow Manager creates an Email task in the workspace.
3. Click Done.
4. Double-click the Email task in the workspace.

   The Edit Tasks dialog box appears.
5. Click Rename to enter a name for the task.
6. Enter a description for the task in the Description field.
7. Click the Properties tab.
8. Enter the fully qualified email address of the mail recipient in the Email User Name field.
9. Enter the subject of the email in the Email Subject field. You can use a service, service process, workflow, or worklet variable in the email subject. Or, you can leave this field blank.
10. Click the Open button in the Email Text field to open the Email Editor.

11. Enter the text of the email message in the Email Editor. You can use service, service process, workflow, and worklet variables in the email text. Or, you can leave the Email Text field blank.

    Note: You can incorporate format tags and email variables in a post-session email. However, you cannot add them to an Email task outside the context of a session.

12. Click OK twice to save the changes.

Working with Post-Session Email

You can configure a session to send email when it fails or succeeds. You can create separate email tasks for success and failure email.

The Integration Service sends post-session email at the end of a session, after executing post-session shell commands or stored procedures. When the Integration Service encounters an error sending the email, it writes a message to the Log Service. It does not fail the session.

You can specify a reusable Email that task you create in the Task Developer for either success email or failure email. Or, you can create a non-reusable Email task for each session property. When you create a non-reusable Email task for a session, you cannot use the Email task in a workflow or worklet.

You cannot specify a non-reusable Email task you create in the Workflow or Worklet Designer for post-session email.

You can use parameters and variables in the email user name, subject, and text. Use any parameter or variable type that you can define in the parameter file. For example, you can use the service variable $PMSuccessEmailUser or $PMFailureEmailUser for the email recipient. Ensure that you specify the values of the service variables for the Integration Service that runs the session. You can also enter a parameter or variable within the email subject or text, and define it in the parameter file.

Email Variables and Format Tags

Use email variables and format tags in an email message for post-session emails. You can use some email variables in the subject of the email. With email variables, you can include important session information in the email, such as the number of rows loaded, the session completion time, or read and write statistics. You can also attach the session log or other relevant files to the email. Use format tags in the body of the message to make the message easier to read.

Note: The Integration Service does not limit the type or size of attached files. However, since large attachments can cause problems with the email system, avoid attaching excessively large files, such as session logs generated using verbose tracing. The Integration Service generates an error message in the email if an error occurs attaching the file.

The following table describes the email variables that you can use in a post-session email:

Email Variable	Description
%a<filename>	Attach the named file. The file must be local to the Integration Service. The following file names are valid: %a<c:\data\sales.txt> or %a</users/john/data/sales.txt>. The email does not display the full path for the file. Only the attachment file name appears in the email. **Note:** The file name cannot include the greater than character (>) or a line break.
%b	Session start time.
%c	Session completion time.
%d	Name of the repository containing the session.
%e	Session status.
%g	Attach the session log to the message. The Integration Service attaches a session log if you configure the session to create a log file. If you do not configure the session to create a log file or if you run a session on a grid, the Integration Service creates a temporary file in the PowerCenter Services installation directory and attaches the file. If the Integration Service does not use operating system profiles, verify that the user that starts Informatica Services has permissions on PowerCenter Services installation directory to create a temporary log file. If the Integration Service uses operating system profiles, verify that the operating system user of the operating system profile has permissions on PowerCenter Services installation directory to create a temporary log file.
%i	Session elapsed time.
%l	Total rows loaded.
%m	Name of the mapping used in the session.
%n	Name of the folder containing the session.
%r	Total rows rejected.
%s	Session name.
%t	Source and target table details, including read throughput in bytes per second and write throughput in rows per second. The Integration Service includes all information displayed in the session detail dialog box.
%u	Repository user name.
%v	Integration Service name.
%w	Workflow name.
%y	Session run mode (normal or recovery).
%z	Workflow run instance name.
Note: The Integration Service ignores %a, %g, and %t when you include them in the email subject. Include these variables in the email message only.	

The following table lists the format tags you can use in an Email task:

Formatting	Format Tag
tab	\t
new line	\n

Post-Session Email

You can configure post-session email to use a reusable or non-reusable Email task.

Using a Reusable Email Task

Complete the following steps to configure post-session email to use a reusable Email task:

1. Open the session properties and click the Components tab.
2. Select Reusable in the Type column for the success email or failure email field.
3. Click the Open button in the Value column to select the reusable Email task.
4. Select the Email task in the Object Browser dialog box and click OK.
5. Optionally, edit the Email task for this session property by clicking the Edit button in the Value column.

   If you edit the Email task for either success email or failure email, the edits only apply to this session.
6. Click OK to close the session properties.

Using a Non-Reusable Email Task

Complete the following steps to configure success email or failure email to use a non-reusable Email task:

1. Open the session properties and click the Components tab.
2. Select Non-Reusable in the Type column for the success email or failure email field.
3. Open the email editor using the Open button.
4. Edit the Email task and click OK.
5. Click OK to close the session properties.

Sample Email

The following example shows a user-entered text from a sample post-session email configuration using variables:

```
Session complete.
Session name: %s
Integration Service name: %v
%l
%r
%e
%b
%c
```

```
%i
%g
```

The following is sample output from the configuration above:

```
Session complete.
Session name: sInstrTest
Integration Service name: Node01IS
Total Rows Loaded = 1
Total Rows Rejected = 0
Completed
Start Time: Tue Nov 22 12:26:31 2005
Completion Time: Tue Nov 22 12:26:41 2005
Elapsed time: 0:00:10 (h:m:s)
```

Suspension Email

You can configure a workflow to send email when the Integration Service suspends the workflow. For example, when a task fails, the Integration Service suspends the workflow and sends the suspension email. You can fix the error and recover the workflow.

If another task fails while the Integration Service is suspending the workflow, you do not get the suspension email again. However, the Integration Service sends another suspension email if another task fails after you recover the workflow.

Configure suspension email on the General tab of the workflow properties. You can use service, service process, workflow, and worklet variables in the email user name, subject, and text. For example, you can use the service variable $PMSuccessEmailUser or $PMFailureEmailUser for the email recipient. Ensure that you specify the values of the service variables for the Integration Service that runs the session. You can also enter a parameter or variable within the email subject or text, and define it in the parameter file.

Configuring Suspension Email

Configure a workflow to send an email when the Integration Service suspends the workflow.

1. In the Workflow Designer, open the workflow.
2. Click Workflows > Edit to open the workflow properties.
3. On the General tab, select Suspend on Error.
4. Click the Browse Emails button to select a reusable Email task.

   Note: The Workflow Manager returns an error if you do not have any reusable Email task in the folder. Create a reusable Email task in the folder before you configure suspension email.
5. Choose a reusable Email task and click OK.
6. Click OK to close the workflow properties.

Using Service Variables to Address Email

Use service variables to address email in Email tasks, post-session email, and suspension email. When you configure the Integration Service, you configure the service variables. You may need to verify these variables with the domain administrator. You can use the following service variables as the email recipient:

- **$PMSuccessEmailUser.** Defines the email address of the user to receive email when a session completes successfully. Use this variable with post-session email. You can also use it to address email in standalone Email tasks or suspension email.
- **$PMFailureEmailUser.** Defines the email address of the user to receive email when a session completes with failure or when the Integration Service suspends a workflow. Use this variable with post-session or suspension email. You can also use it to address email in standalone Email tasks.

When you use one of these service variables, the Integration Service sends email to the address configured for the service variable. $PMSuccessEmailUser and $PMFailureEmailUser are optional process variables. Verify that you define a variable before using it to address email.

You might use this functionality when you have an administrator who troubleshoots all failed sessions. Instead of entering the administrator email address for each session, use the email variable $PMFailureEmailUser as the recipient for post-session email. If the administrator changes, you can correct all sessions by editing the $PMFailureEmailUser service variable, instead of editing the email address in each session.

You might also use this functionality when you have different administrators for different Integration Services. If you deploy a folder from one repository to another or otherwise change the Integration Service that runs the session, the new service sends email to users associated with the new service when you use process variables instead of hard-coded email addresses.

Tips for Sending Email

When the Integration Service runs on Windows, configure a Microsoft Outlook profile for each node.

If you run the Integration Service on multiple nodes in a Windows environment, create a Microsoft Outlook profile for each node. To use the profile on multiple nodes for multiple users, create a generic Microsoft Outlook profile, such as “PowerCenter,” and use this profile on each node in the domain. Use the same profile on each node to ensure that the Microsoft Exchange Profile you configured for the Integration Service matches the profile on each node.

Use service variables to address email.

Use service variables to address email in Email tasks, post-session email, and suspension email. When the service variables $PMSuccessEmailUser and $PMFailureEmailUser are configured for the Integration Service, use them to address email. You can change the email recipient for all sessions the service runs by editing the service variables. It is easier to deploy sessions into production if you define service variables for both development and production servers.

Generate and send post-session reports.

Use a post-session success command to generate a report file and attach that file to a success email. For example, you create a batch file called Q3rpt.bat that generates a sales report, and you are running Microsoft Outlook on Windows.

Use other mail programs.

If you do not have Microsoft Outlook and you do not configure the Integration Service to send email using SMTP, use a post-session success command to invoke a command line email program, such as Windmill. In this case, you do not have to enter the email user name or subject, since the recipients, email subject, and body text will be contained in the batch file, sendmail.bat.

Workflow Monitor

Workflow Monitor Overview

You can monitor workflows and tasks in the Workflow Monitor. A workflow is a set of instructions that tells an Integration Service how to run tasks. Integration Services run on nodes or grids. The nodes, grids, and services are all part of a domain.

With the Workflow Monitor, you can view details about a workflow or task in Gantt Chart view or Task view. You can also view details about the Integration Service, nodes, and grids.

The Workflow Monitor displays workflows that have run at least once. You can run, stop, abort, and resume workflows from the Workflow Monitor. The Workflow Monitor continuously receives information from the Integration Service and Repository Service. It also fetches information from the repository to display historic information.

The Workflow Monitor consists of the following windows:

- **Navigator window.** Displays monitored repositories, Integration Services, and repository objects.
- **Output window.** Displays messages from the Integration Service and the Repository Service.
- **Properties window.** Displays details about services, workflows, worklets, and tasks.
- **Time window.** Displays progress of workflow runs.
- **Gantt Chart view.** Displays details about workflow runs in chronological (Gantt Chart) format.
- **Task view.** Displays details about workflow runs in a report format, organized by workflow run.

The Workflow Monitor displays time relative to the time configured on the Integration Service node. For example, a folder contains two workflows. One workflow runs on an Integration Service in the local time zone, and the other runs on an Integration Service in a time zone two hours later. If you start both workflows at 9 a.m. local time, the Workflow Monitor displays the start time as 9 a.m. for one workflow and as 11 a.m. for the other workflow.

Toggle between Gantt Chart view and Task view by clicking the tabs on the bottom of the Workflow Monitor.

You can view and hide the Output and Properties windows in the Workflow Monitor. To view or hide the Output window, click View > Output. To view or hide the Properties window, click View > Properties View.

You can also dock the Output and Properties windows at the bottom of the Workflow Monitor workspace. To dock the Output or Properties window, right-click a window and select Allow Docking. If the window is floating, drag the window to the bottom of the workspace. If you do not allow docking, the windows float in the Workflow Monitor workspace.

Using the Workflow Monitor

The Workflow Monitor provides options to view information about workflow runs. After you open the Workflow Monitor and connect to a repository, you can view dynamic information about workflow runs by connecting to an Integration Service.

You can customize the Workflow Monitor display by configuring the maximum days or workflow runs the Workflow Monitor shows. You can also filter tasks and Integration Services in both Gantt Chart and Task view.

Complete the following steps to monitor workflows:

1. Open the Workflow Monitor.
2. Connect to the repository containing the workflow.
3. Connect to the Integration Service.
4. Select the workflow you want to monitor.
5. Select Gantt Chart view or Task view.

Opening the Workflow Monitor

You can open the Workflow Monitor in the following ways:

1. Select Start > Programs > Informatica PowerCenter [version] > Client > Workflow Monitor from the Windows Start menu.

   -or-

   Configure the Workflow Manager to open the Workflow Monitor when you run a workflow from the Workflow Manager.

   -or-

   Click Tools > Workflow Monitor from the Designer, Workflow Manager, or Repository Manager.

   -or-

   Click the Workflow Monitor icon on the Tools toolbar. When you use a Tools button to open the Workflow Monitor, PowerCenter uses the same repository connection to connect to the repository and opens the same folders.

   -or-

   From the Workflow Manager, right-click an Integration Service or a repository, and select Run Monitor.

You can open multiple instances of the Workflow Monitor on one machine using the Windows Start menu.

Connecting to a Repository

When you open the Workflow Monitor, you must connect to a repository. Connect to repositories by clicking Repository > Connect. Enter the repository name and connection information.

After you connect to a repository, the Workflow Monitor displays a list of Integration Services available for the repository. The Workflow Monitor can monitor multiple repositories, Integration Services, and workflows at the same time.

Note: If you are not connected to a repository, you can remove the repository from the Navigator. Select the repository in the Navigator and click Edit > Delete. The Workflow Monitor displays a message verifying that you want to remove the repository from the Navigator list. Click Yes to remove the repository. You can connect to the repository again at any time.

Connecting to an Integration Service

When you connect to a repository, the Workflow Monitor displays all Integration Services associated with the repository. This includes active and deleted Integration Services. To monitor tasks and workflows that run on an Integration Service, you must connect to the Integration Service. In the Navigator, the Workflow Monitor displays a red icon over deleted Integration Services.

To connect to an Integration Service, right-click it and select Connect. When you connect to an Integration Service, you can view all folders that you have permission for. To disconnect from an Integration Service, right-click it and select Disconnect. When you disconnect from an Integration Service, or when the Workflow Monitor cannot connect to an Integration Service, the Workflow Monitor displays disconnected for the Integration Service status.

The Workflow Monitor is resilient to the Integration Service. If the Workflow Monitor loses connection to the Integration Service, LMAPI tries to reestablish the connection for the duration of the PowerCenter Client resilience time-out period.

After the connection is reestablished, the Workflow Monitor retrieves the workflow status from the repository. Depending on your Workflow Monitor advanced settings, you may have to reopen the workflow to view the latest status of child tasks.

You can also ping an Integration Service to verify that it is running. Right-click the Integration Service in the Navigator and select Ping Integration Service. You can view the ping response time in the Output window.

Note: You can also open an Integration Service in the Navigator without connecting to it. When you open an Integration Service, the Workflow Monitor gets workflow run information stored in the repository. It does not get dynamic workflow run information from currently running workflows.

Filtering Tasks and Integration Services

You can filter tasks and Integration Services in both Gantt Chart view and Task view. Use the Filters menu to hide tasks and Integration Services you do not want to view in the Workflow Monitor.

Filtering Tasks

You can view all or some workflow tasks. You can filter tasks you do not want to view. For example, if you want to view only Session tasks, you can hide all other tasks. You can view all tasks at any time.

To filter tasks:

1. Click Filters > Tasks.

   -or-

   Click Filters > Deleted Tasks.

   The Filter Tasks dialog box appears.
2. Clear the tasks you want to hide, and select the tasks you want to view.

3. Click OK.

   Note: When you filter a task, the Gantt Chart view displays a red link between tasks to indicate a filtered task. You can double-click the link to view the tasks you hid.

Filtering Integration Services

When you connect to a repository, the Workflow Monitor displays all Integration Services associated with the repository. You can filter out Integration Services to view only Integration Services you want to monitor.

When you hide an Integration Service, the Workflow Monitor hides the Integration Service from the Navigator for the Gantt Chart and Task views. You can show the Integration Service again at any time.

You can hide unconnected Integration Services. When you hide a connected Integration Service, the Workflow Monitor asks if you want to disconnect from the Integration Service and then filter it. You must disconnect from an Integration Service before hiding it.

To filter Integration Services:

1. In the Navigator, right-click a repository to which you are connected and select Filter Integration Services.

   The Filter Integration Services dialog box appears.
2. Select the Integration Services you want to view and clear the Integration Services you want to filter. Click OK.

   If you are connected to an Integration Service that you clear, the Workflow Monitor prompts you to disconnect from the Integration Service before filtering.
3. Click Yes to disconnect from the Integration Service and filter it.

   -or-

   Click No to remain connected to the Integration Service.

   Tip: To filter an Integration Service in the Navigator, right-click it and select Filter Integration Service.

Opening and Closing Folders

You can select which folders to open and close in the Workflow Monitor. When you open a folder, the Workflow Monitor displays the number of workflow runs that you configured in the Workflow Monitor options.

You can open and close folders in the Gantt Chart and Task views. When you open a folder, it opens in both views. To open a folder, right-click it in the Navigator and select Open. Or, you can double-click the folder.

Viewing Statistics

You can view statistics about the objects you monitor in the Workflow Monitor. Click View > Statistics. The Statistics window displays the following information:

- **Number of opened repositories.** Number of repositories you are connected to in the Workflow Monitor.

- **Number of connected Integration Services.** Number of Integration Services you connected to since you opened the Workflow Monitor.
- **Number of fetched tasks.** Number of tasks the Workflow Monitor fetched from the repository during the period specified in the Time window.

You can also view statistics about nodes and sessions.

Viewing Properties

You can view properties for the following items:

- **Tasks.** You can view properties, such as task name, start time, and status.
- **Sessions.** You can view properties about the Session task and session run, such as mapping name and number of rows successfully loaded. You can also view load statistics about the session run. You can also view performance details about the session run.
- **Workflows.** You can view properties such as start time, status, and run type.
- **Links.** When you double-click a link between tasks in Gantt Chart view, you can view tasks that you filtered out.
- **Integration Services.** You can view properties such as Integration Service version and startup time. You can also view the sessions and workflows running on the Integration Service.
- **Grid.** You can view properties such as the name, Integration Service type, and code page of a node in the Integration Service grid. You can view these details in the Integration Service Monitor.
- **Folders.** You can view properties such as the number of workflow runs displayed in the Time window.

To view properties for all objects, right-click the object and select Properties. You can right-click items in the Navigator or the Time window in either Gantt Chart view or Task view.

To view link properties, double-click the link in the Time window of Gantt Chart view. When you view link properties, you can double-click a task in the Link Properties dialog box to view the properties for the filtered task.

Customizing Workflow Monitor Options

You can configure how the Workflow Monitor displays general information, workflows, and tasks. You can configure general tasks such as the maximum number of days or runs that the Workflow Monitor appears. You can also configure options specific to Gantt Chart and Task view.

Click Tools > Options to configure Workflow Monitor options.

You can configure the following options in the Workflow Monitor:

- **General.** Customize general options such as the maximum number of workflow runs to display and whether to receive messages from the Workflow Manager. See "Configuring General Options" on page 206.
- **Gantt Chart view.** Configure Gantt Chart view options such as workspace color, status colors, and time format. See "Configuring Gantt Chart View Options" on page 206.
- **Task view.** Configure which columns to display in Task view. See "Configuring Task View Options" on page 206.

- **Advanced.** Configure advanced options such as the number of workflow runs the Workflow Monitor holds in memory for each Integration Service. See "Configuring Advanced Options" on page 207.

Configuring General Options

You can customize general options such as the maximum number of days to display and which text editor to use for viewing session and workflow logs.

The following table describes the options you can configure on the General tab:

Setting	Description
Maximum Days	Number of tasks the Workflow Monitor displays up to a maximum number of days. Default is 5.
Maximum Workflow Runs per Folder	Maximum number of workflow runs the Workflow Monitor displays for each folder. Default is 200.
Receive Messages from Workflow Manager	Select to receive messages from the Workflow Manager. The Workflow Manager sends messages when you start or schedule a workflow in the Workflow Manager. The Workflow Monitor displays these messages in the Output window.
Receive Notifications from Repository Service	Select to receive notification messages in the Workflow Monitor and view them in the Output window. You must be connected to the repository to receive notifications. Notification messages include information about objects that another user creates, modifies, or delete. You receive notifications about folders and Integration Services. The Repository Service notifies you of the changes so you know objects you are working with may be out of date. You also receive notices posted by the user who manages the Repository Service.

Configuring Gantt Chart View Options

You can configure Gantt Chart view options such as workspace color, status colors, and time format.

The following table describes the options you can configure on the Gantt Chart tab:

Setting	Description
Status Color	Select a status and configure the color for the status. The Workflow Monitor displays tasks with the selected status in the colors you select. You can select two colors to display a gradient.
Recovery Color	Configure the color for the recovery sessions. The Workflow Monitor uses the status color for the body of the status bar, and it uses and the recovery color as a gradient in the status bar.
Workspace Color	Select a color for each workspace component.
Time Format	Select a display format for the time window.

Configuring Task View Options

You can select the columns you want to display in Task view. You can also reorder the columns and specify a default column width.

Configuring Advanced Options

You can configure advanced options such as the number of workflow runs the Workflow Monitor holds in memory for each Integration Service.

The following table describes the options you can configure on the Advanced tab:

Setting	Description
Expand Running Workflows Automatically	Expands running workflows in the Navigator.
Refresh Workflow Tasks When the Connection to the Integration Service is Re-established	Refreshes workflow tasks when you reconnect to the Integration Service.
Expand Workflow Runs When Opening the Latest Runs	Expands workflows when you open the latest run.
Hide Folders/Workflows That Do Not Contain Any Runs When Filtering By Running/ Schedule Runs	Hides folders or workflows under the Workflow Run column in the Time window when you filter running or scheduled tasks.
Highlight the Entire Row When an Item Is Selected	Highlights the entire row in the Time window for selected items. When you disable this option, the Workflow Monitor highlights the item in the Workflow Run column in the Time window.
Open Latest 20 Runs At a Time	You can open the number of workflow runs. Default is 20.
Minimum Number of Workflow Runs (Per Integration Service) the Workflow Monitor Will Accumulate in Memory	Specifies the minimum number of workflow runs for each Integration Service that the Workflow Monitor holds in memory before it starts releasing older runs from memory. When you connect to an Integration Service, the Workflow Monitor fetches the number of workflow runs specified on the General tab for each folder you connect to. When the number of runs is less than the number specified in this option, the Workflow Monitor stores new runs in memory until it reaches this number.

Using Workflow Monitor Toolbars

The Workflow Monitor toolbars allow you to select tools and tasks quickly. You can perform the following toolbar operations:

- Display or hide a toolbar.
- Create a new toolbar.
- Add or remove buttons.

By default, the Workflow Monitor displays the following toolbars:

- **Standard.** Contains buttons to connect to and disconnect from repositories, print, view print previews, search the workspace, show or hide the navigator in task view, and show or hide the output window.
- **Integration Service.** Contains buttons to connect to and disconnect from Integration Services, ping Integration Service, and perform workflows operations.

- **View.** Contains buttons to configure time increments and show properties, workflow logs, or session logs.
- **Filters.** Contains buttons to display most recent runs, and to filter tasks, Integration Services, and folders.

After a toolbar appears, it displays until you exit the Workflow Monitor or hide the toolbar. You can drag each toolbar to resize and reposition each toolbar.

Working with Tasks and Workflows

You can perform the following tasks with objects in the Workflow Monitor:

- Run a task or workflow.
- Resume a suspended workflow.
- Restart a task or workflow without recovery.
- Stop or abort a task or workflow.
- Schedule and unschedule a workflow.
- View session logs and workflow logs.
- View history names.

Opening Previous Workflow Runs

In both the Gantt Chart view and Task View, you can open previous workflow runs.

1. In the Navigator or Workflow Run List, select the workflow with the runs you want to see.
2. Right-click the workflow and select Open Latest 20 Runs.

   The menu option is disabled when the latest 20 workflow runs are already open.

   Up to 20 of the latest runs appear.

Displaying Previous Workflow Runs

In both the Gantt Chart view and Task View, you can display previous workflow runs.

1. Click on the Display Recent Runs icon.
2. Select the number of runs you want to display.

   The runs appear in the Workflow Run List.

Running a Task, Workflow, or Worklet

The Workflow Monitor displays workflows that have run at least once. In the Workflow Monitor, you can run a workflow or any task or worklet in the workflow. To run a workflow or part of a workflow, right-click the workflow or task and select a restart option. When you select restart, the task, workflow, or worklet runs on the Integration Service you specify in the workflow properties.

You can also run part of a workflow. When you run part of a workflow, the Integration Service runs the workflow from the selected task to the end of the workflow.

Restart behavior for real-time sessions depends on the real-time source.

Recovering a Workflow or Worklet

In the workflow properties, you can choose to suspend the workflow or worklet if a session fails. After you fix the errors that caused the session to fail, recover the workflow in the Workflow Monitor. When you recover a workflow, the Integration Service recovers the failed session and continues running the rest of the tasks in the workflow path. Recovery behavior for real-time sessions depends on the real-time source.

The Integration Service appends log events to the existing log events when you recover the workflow. The Integration Service creates another session log when you recover a session.

To recover a workflow or worklet:

1. In the Navigator, select the workflow or worklet you want to recover.
2. Click Tasks > Recover.

   The Workflow Monitor displays Integration Service messages about the recover command in the Output window.

Restarting a Task or Workflow Without Recovery

You can restart a task or workflow without recovery by using a cold start. Cold start is a start mode that the Integration Service uses to restart a task or workflow without recovery. When you restart a failed task or workflow that has recovery enabled, the Integration Service does not process recovery data. The Integration Service clears the state of operation and the recovery file or table before it restarts the task or workflow. You do not want to recover data if you already cleaned up the target system.

To restart a task or workflow without recovery:

1. In the Navigator, select the task or workflow you want to restart.
2. Click Tasks > Cold Start Task or Workflows > Cold Start Workflow.

Stopping or Aborting Tasks and Workflows

You can stop or abort a task, workflow, or worklet in the Workflow Monitor at any time. When you stop a task in the workflow, the Integration Service stops processing the task and all other tasks in its path. The Integration Service continues running concurrent tasks. If the Integration Service cannot stop processing the task, you need to abort the task. When the Integration Service aborts a task, it kills the DTM process and terminates the task.

Behavior for real-time sessions depends on the real-time source.

To stop or abort workflows, tasks, or worklets in the Workflow Monitor:

1. In the Navigator, select the task, workflow, or worklet you want to stop or abort.
2. Click Tasks > Stop.

   -or-

   Click Tasks > Abort.

   The Workflow Monitor displays the status of the stop or abort command in the Output window.

Scheduling Workflows

You can schedule workflows in the Workflow Monitor. You can schedule any workflow that is not configured to run on demand. When you try to schedule a run on demand workflow, the Workflow Monitor displays an error message in the Output window.

When you schedule an unscheduled workflow, the workflow uses its original schedule specified in the workflow properties. If you want to specify a different schedule for the workflow, you must edit the scheduler in the Workflow Manager.

To schedule a workflow in the Workflow Monitor:

1. Right-click the workflow and select Schedule.
2. The Workflow Monitor displays the workflow status as Scheduled, and displays a message in the Output window.

Unscheduling Workflows

You can unschedule workflows in the Workflow Monitor.

1. Right-click the workflow and select Unschedule.
2. The Workflow Monitor displays the workflow status as Unscheduled and displays a message in the Output window.

Session and Workflow Logs in the Workflow Monitor

You can view session and workflow logs from the Workflow Monitor. You can view the most recent log, or you can view past logs.

If you want to view past session or workflow logs, configure the session or workflow to save logs by timestamp. When you configure the workflow to save log files, the workflow creates a text file and the binary file that displays in the Log Events window. You can save log files by timestamp or by workflow or session runs. You can configure how many workflow or session runs to save.

When you open a session or workflow log, the Log Events window sends a request to the Log Agent. The Log Agent retrieves logs from each node that ran the session or workflow. The Log Events window displays the logs by node.

Related Topics:

- “Session and Workflow Logs” on page 229

Viewing Session and Workflow Logs

1. Right-click a session or workflow in the Navigator or Time window.
2. Select Get Session Log

   -or-

   Select Get Workflow Log.

   The log file opens in the Log Events window.

   Tip: When the Workflow Monitor retrieves the session or workflow log, you can press Esc to cancel the process.

Viewing History Names

If you rename a task, workflow, or worklet, the Workflow Monitor can show a history of names. When you start a renamed task, workflow, or worklet, the Workflow Monitor displays the current name. To view a list of historical names, select the task, workflow, or worklet in the Navigator. Right-click and select Show History Names.

Workflow and Task Status

The Workflow Monitor displays the status of workflows and tasks.

The following table describes the different statuses for workflows and tasks:

Status Name	Status for	Description
Aborted	Workflows Tasks	You choose to abort the workflow or task in the Workflow Monitor or through *pmcmd*. The Integration Service kills the DTM process and aborts the task. You can recover an aborted workflow if you enable the workflow for recovery.
Aborting	Workflows Tasks	The Integration Service is in the process of aborting the workflow or task.
Disabled	Workflows Tasks	You select the Disabled option in the workflow or task properties. The Integration Service does not run the disabled workflow or task until you clear the Disabled option.
Failed	Workflows Tasks	The Integration Service fails the workflow or task because it encountered errors. You cannot recover a failed workflow.
Preparing to Run	Workflows	The Integration Service is waiting for an execution lock for the workflow.
Running	Workflows Tasks	The Integration Service is running the workflow or task.
Scheduled	Workflows	You schedule the workflow to run at a future date. The Integration Service runs the workflow for the duration of the schedule.
Stopped	Workflows Tasks	You choose to stop the workflow or task in the Workflow Monitor or through *pmcmd*. The Integration Service stops processing the task and all other tasks in its path. The Integration Service continues running concurrent tasks. You can recover a stopped workflow if you enable the workflow for recovery.
Stopping	Workflows Tasks	The Integration Service is in the process of stopping the workflow or task.
Succeeded	Workflows Tasks	The Integration Service successfully completes the workflow or task.
Suspended	Workflows Worklets	The Integration Service suspends the workflow because a task failed and no other tasks are running in the workflow. This status is available when you select the Suspend on Error option. You can recover a suspended workflow.

Status Name	Status for	Description
Suspending	Workflows Worklets	A task fails in the workflow when other tasks are still running. The Integration Service stops running the failed task and continues running tasks in other paths. This status is available when you select the Suspend on Error option.
Terminated	Workflows Tasks	The Integration Service shuts down unexpectedly when running this workflow or task. You can recover a terminated workflow if you enable the workflow for recovery.
Terminating	Workflows Tasks	The Integration Service is in the process of terminating the workflow or task.
Unknown Status	Workflows Tasks	This status displays in the following situations: - The Integration Service cannot determine the status of the workflow or task. - The Integration Service does not respond to a ping from the Workflow Monitor. - The Workflow Monitor cannot connect to the Integration Service within the resilience timeout period. - The Integration Service fails to authenticate a user within the default timeout of 30 seconds. To increase the timeout, append the following content to the INFA_JAVA_OPTS environment variable: `-DINFA_DEFAULT_CONNECTION_TIMEOUT=<value in seconds>` **Note:** Separate each value for the INFA_JAVA_OPTS variable with a space.
Unscheduled	Workflows	You remove a workflow from the schedule.
Waiting	Workflows Tasks	The Integration Service is waiting for available resources so it can run the workflow or task. For example, you may set the maximum number of running Session and Command tasks allowed for each Integration Service process on the node to 10. If the Integration Service is already running 10 concurrent sessions, all other workflows and tasks have the Waiting status until the Integration Service is free to run more tasks.

To see a list of tasks by status, view the workflow in the Task view and filter by status. Or, click Edit > List Tasks in Gantt Chart view.

Using the Gantt Chart View

You can view chronological details of workflow runs with the Gantt Chart view. The Gantt Chart view displays the following information:

- **Task name.** Name of the task in the workflow.
- **Duration.** The length of time the Integration Service spends running the most recent task or workflow.
- **Status.** The status of the most recent task or workflow.
- **Connection between objects.** The Workflow Monitor shows links between objects in the Time window.

Listing Tasks and Workflows

The Workflow Monitor lists tasks and workflows in all repositories you connect to. You can view tasks and workflows by status, such as failed or succeeded. You can highlight the task in Gantt Chart view by double-clicking the task in the list.

To view a list of tasks and workflows by status:

1. Open the Gantt Chart view and click Edit > List Tasks.
2. In the List What field, select the type of task status you want to list.

   For example, select Failed to view a list of failed tasks and workflows.
3. Click List to view the list.

   Tip: Double-click the task name in the List Tasks dialog box to highlight the task in Gantt Chart view.

Navigating the Time Window in Gantt Chart View

You can scroll through the Time window in Gantt Chart view to monitor the workflow runs. To scroll the Time window, use one of the following methods:

- Use the scroll bars.
- Right-click the task or workflow and click Go To Next Run or Go To Previous Run.
- Click View > Organize to select the date you want to display.

When you click View > Organize, the Go To field appears above the Time window. Click the Go To field to view a calendar and select the date you want to display. When you select a date, the Workflow Monitor displays that date beginning at 12:00 a.m.

Zooming the Gantt Chart View

You can change the zoom settings in Gantt Chart view. By default, the Workflow Monitor shows the Time window in increments of one hour. You can change the time increments to zoom the Time window.

To zoom the Time window in Gantt Chart view, click View > Zoom, and then select the time increment. You can also select the time increment in the Zoom button on the toolbar.

Performing a Search

Use the search tool in the Gantt Chart view to search for tasks, workflows, and worklets in all repositories you connect to. The Workflow Monitor searches for the word you specify in task names, workflow names, and worklet names. You can highlight the task in Gantt Chart view by double-clicking the task after searching.

To perform a search:

1. Open the Gantt Chart view and click Edit > Find.

   The Find Object dialog box appears.
2. In the Find What field, enter the keyword you want to find.

3. Click Find Now.

   The Workflow Monitor displays a list of tasks, workflows, and worklets that match the keyword.

   Tip: Double-click the task name in the Find Object dialog box to highlight the task in Gantt Chart view.

Opening All Folders

You can open all folders that you have permission for in a repository. To open all the folders in the Gantt Chart view, right-click the Integration Service you want to view, and select Open All Folders. The Workflow Monitor displays workflows and tasks in the folders.

Using the Task View

The Task view displays information about workflow runs in a report format. The Task view provides a convenient way to compare and filter details of workflow runs. Task view displays the following information:

- **Workflow run list.** The list of workflow runs. The workflow run list contains folder, workflow, worklet, and task names. The Workflow Monitor displays workflow runs chronologically with the most recent run at the top. It displays folders and Integration Services alphabetically.
- **Status message.** Message from the Integration Service regarding the status of the task or workflow.
- **Run type.** The method you used to start the workflow. You might manually start the workflow or schedule the workflow to start.
- **Node.** Node of the Integration Service that ran the task.
- **Start time.** The time that the Integration Service starts running the task or workflow.
- **Completion time.** The time that the Integration Service finishes executing the task or workflow.
- **Status.** The status of the task or workflow.

You can perform the following tasks in Task view:

- **Filter tasks.** Use the Filter menu to select the tasks you want to display or hide.
- **Hide and view columns.** Hide or view an entire column in Task view.
- **Hide and view the Navigator.** You can hide the Navigator in Task view. Click View > Navigator to hide or view the Navigator.

To view the tasks in Task view, select the Integration Service you want to monitor in the Navigator.

Filtering in Task View

In Task view, you can view all or some workflow tasks. You can filter tasks in the following ways:

- **By task type.** You can filter out tasks you do not want to view. For example, if you want to view only Session tasks, you can filter out all other tasks.
- **By nodes in the Navigator.** You can filter the workflow runs in the Time window by selecting different nodes in the Navigator. For example, when you select a repository name in the Navigator, the Time window displays all workflow runs that ran on the Integration Services registered to that repository.

When you select a folder name in the Navigator, the Time window displays all workflow runs in that folder.

- **By the most recent runs.** To display by the most recent runs, click Filters > Most Recent Runs and select the number of runs you want to display.
- **By Time window columns.** You can click Filters > Auto Filter and filter by properties you specify in the Time window columns.

To filter by Time view columns:

1. Click Filters > Auto Filter.

   The Filter button appears in the some columns of the Time Window in Task view.
2. Click the Filter button in a column in the Time window.
3. Select the properties you want to filter.

   When you click the Filter button in either the Start Time or Completion Time column, you can select a custom time to filter.
4. Select Custom for either Start Time or Completion Time.

   The Filter Start Time or Custom Completion Time dialog box appears.
5. Choose to show tasks before, after, or between the time you specify.
6. Select the date and time. Click OK.

Opening All Folders

You can open all folders that you have permission for in a repository. To open all folders in the Task view, right-click the Integration Service with the folders you want to view, and select Open All Folders. The Workflow Monitor displays workflows and tasks in the folders.

Tips for Monitoring Workflows

Reduce the size of the Time window.

When you reduce the size of the Time window, the Workflow Monitor refreshes the screen faster, reducing flicker.

Use the Repository Manager to truncate the list of workflow logs.

If the Workflow Monitor takes a long time to refresh from the repository or to open folders, truncate the list of workflow logs. When you configure a session or workflow to archive session logs or workflow logs, the Integration Service saves those logs in local directories. The repository also creates an entry for each saved workflow log and session log. If you move or delete a session log or workflow log from the workflow log directory or session log directory, truncate the lists of workflow and session logs to remove the entries from the repository. The repository always retains the most recent workflow log entry for each workflow.

Workflow Monitor Details

Workflow Monitor Details Overview

The Workflow Monitor displays information that you can use to troubleshoot and analyze workflows. You can view details about services, workflows, worklets, and tasks in the Properties window of the Workflow Monitor.

You can view the following details in the Workflow Monitor:

- **Repository Service details.** View information about repositories, such as the number of connected Integration Services.
- **Integration Service properties.** View information about the Integration Service, such as the Integration Service Version. You can also view system resources that running workflows consume, such as the system swap usage at the time of the running workflow.
- **Repository folder details.** View information about a repository folder, such as the folder owner.
- **Workflow run properties.** View information about a workflow, such as the start and end time.
- **Worklet run properties.** View information about a worklet, such as the execution nodes on which the worklet is run.
- **Command task run properties.** View the information about Command tasks in a running workflow, such as the start and end time.
- **Session task run properties.** View information about Session tasks in a running workflow, such as details on session failures.
- **Performance details.** View counters that help you understand the session and mapping efficiency, such as information on the data cache size for an Aggregator transformation.

Repository Service Details

To view details about a repository, right-click on the repository and choose Properties.

The following table describes the attributes that appear in the Repository Details area:

Attribute Name	Description
Repository Name	Name of the repository.
Is Opened	Yes, if you are connected to the repository. Otherwise, value is No.
User Name	Name of the user connected to the repository. Attribute appears if you are connected to the repository.
Number of Connected Integration Services	Number of Integration Services you are connected to in the Workflow Monitor. Attribute appears if you are connected to the repository.
Is Versioning Enabled	Indicates whether repository versioning is enabled.

Integration Service Properties

When you view Integration Service properties, the following areas appear in the Properties window:

- **Integration Service Details.** Displays information about the Integration Service.
- **Integration Service Monitor.** Displays system resource usage information about nodes associated with the Integration Service.

Integration Service Details

To view details about the Integration Service, right-click an Integration Service and choose Properties.

The following table describes the attributes that appear in the Integration Service Details area:

Attribute Name	Description
Integration Service Name	Name of the Integration Service.
Integration Service Version	PowerCenter version and build. Appears if you are connected to the Integration Service in the Workflow Monitor.
Integration Service Mode	Data movement mode of the Integration Service. Appears if you are connected to the Integration Service in the Workflow Monitor.
Integration Service OperatingMode	The operating mode of the Integration Service. Appears if you are connected to the Integration Service in the Workflow Monitor.
Startup Time	Time the Integration Service started. Startup Time appears in the following format: MM/DD/YYYY HH:MM:SS AM|PM. Appears if you are connected to the Integration Service in the Workflow Monitor.
Current Time	Current time of the Integration Service.
Last Updated Time	Time the Integration Service was last updated. Last Updated Time appears in the following format: MM/DD/YYYY HH:MM:SS AM|PM. Appears if you are connected to the Integration Service in the Workflow Monitor.
Grid Assigned	Grid the Integration Service is assigned to. Attribute appears if the Integration Service is assigned to a grid. Appears if you are connected to the Integration Service in the Workflow Monitor.
Node(s)	Names of nodes configured to run Integration Service processes. Appears if you are connected to the Integration Service in the Workflow Monitor.
Is Connected	Appears if you are not connected to the Integration Service.
Is Registered	Displays one of the following values: - Yes if the Integration Service is associated with a repository. - No if the Integration Service is not associated with a repository. Appears if you are not connected to the Integration Service.

Integration Service Monitor

The Integration Service Monitor displays system resource usage information about nodes associated with the Integration Service. This window also displays system resource usage information about tasks running on the node.

To view the Integration Service Monitor, right-click an Integration Service and choose Properties. The Integration Service Monitor area appears if you are connected to an Integration Service. You can view the Integration Service type and code page for each node the Integration Service is running on. To view the tool tip for the Integration Service type and code page, move the pointer over the node name.

The following table describes the attributes that appear in the Integration Service Monitor area:

Attribute Name	Description
Node Name	Name of the node on which the Integration Service is running.
Folder	Folder that contains the running workflow.
Workflow	Name of the running workflow.
Task/Partition	Name of the session and partition that is running. Or, name of Command task that is running.
Status	Status of the task.
Process ID	Process ID of the task.
CPU %	For a node, this is the percent of CPU usage of processes running on the node. For a task, this is the percent of CPU usage by the task process.
Memory Usage	For a node, this is the memory usage of processes running on the node. For a task, this is the memory usage of the task process.
Swap Usage	Amount of swap space usage of processes running on the node.

Repository Folder Details

To view information about a repository folder, right-click the folder and choose Properties.

The following table describes the attributes that appear in the Folder Details area:

Attribute Name	Description
Folder Name	Name of the repository folder.
Is Opened	Indicates if the folder is open.
Number of Workflow Runs Within Time Window	Number of workflows that have run in the time window during which the Workflow Monitor displays workflow statistics.
Number of Fetched Workflow Runs	Number of workflow runs displayed during the time window.

Attribute Name	Description
Workflows Fetched Between	Time period during which the Integration Service fetched the workflows. Appears as, DD/MM/YYYT HH:MM:SS and DD/MM/YYYT HH:MM:SS.
Deleted	Indicates if the folder is deleted.
Owner	Repository folder owner.

Workflow Run Properties

The Workflow Run Properties window displays information about workflows, such as the name of the Integration Service assigned to the workflow and workflow run details.

When you view workflow properties, the following areas appear in the Properties window:

- **Workflow details.** View information about the workflow.
- **Task progress details.** View information about the tasks in the workflow.
- **Session statistics.** View information about the session.

Workflow Details

To view workflow details in the Properties window, right-click on a workflow and choose Get Run Properties. In the Properties window, you can click Get Workflow Log to view the Log Events window for the workflow.

The following table describes the attributes that appear in the Workflow Details area:

Attribute Name	Description
Task Name	Name of the workflow.
Concurrent Type	-
OS Profile	Name of the operating system profile assigned to the workflow. Value is empty if an operating system profile is not assigned to the workflow.
Task Type	Task type is Workflow.
Integration Service Name	Name of the Integration Service assigned to the workflow.
User Name	Name of the user running the workflow.
Start Time	Start time of the workflow.
End Time	End time of the workflow.
Recovery Time(s)	Times of recovery workflows.
Status	Status of the workflow.

Attribute Name	Description
Status Message	Message about the workflow status.
Run Type	Method used to start the workflow.
Deleted	Yes if the workflow is deleted from the repository. Otherwise, value is No.
Version Number	Version number of the workflow.
Execution Node(s)	Nodes on which workflow tasks run.

Task Progress Details

The Task Progress Details area displays the Gantt Chart view of Session and Command tasks in a running workflow.

Session Statistics

The Session Statistics area displays information about sessions, such as the session run time and the number or rows loaded to the targets.

The following table describes the attributes that appear in the Session Statistics area:

Attribute Name	Description
Session	Name of the session.
Source Success Rows	Number of rows the Integration Service successfully read from the source.
Source Failed Rows	Number of rows the Integration Service failed to read from the source.
Target Success Rows	Number of rows the Integration Service wrote to the target.
Target Failed Rows	Number of rows the Integration Service failed to write the target.
Total Transformation Errors	Number of transformation errors in the session.
Start Time	Start time of the session.
End Time	End time of the session.

Worklet Run Properties

The Worklet Details Run Properties window displays information about worklets, such as the name of the Integration Service assigned to the workflow and worklet run details.

When you view worklet properties, the following areas appear in the Properties window:

- **Worklet details.** View information about the worklet.
- **Session statistics.** View information about the session.

Worklet Details

To view worklet details in the Properties window, right-click on a worklet and choose Get Run Properties.

The following table describes the attributes that appear in the Worklet Details area:

Attribute Name	Description
Instance Name	Name of the worklet instance in the workflow.
Task Type	Task type is Worklet.
Integration Service Name	Name of the Integration Service assigned to the workflow associated with the worklet.
Start Time	Start time of the worklet.
End Time	End time of the worklet.
Recovery Time(s)	Time of the recovery worklet run.
Status	Status of the worklet.
Status Message	Message about the worklet status.
Deleted	Indicates if the worklet is deleted from the repository.
Version Number	Version number of the worklet.
Execution Node(s)	Nodes on which worklet tasks run.

Command Task Run Properties

The Task Run Properties window for Command tasks displays information about Command tasks, such as the start time and end time. To view command task details in the Properties window, right-click on a Command task and choose Get Run Properties.

The following table describes the attributes that appear in the Task Details area:

Attribute Name	Description
Instance Name	Command task name.
Task Type	Task type is Command.
Integration Service Name	Name of the Integration Service assigned to the workflow associated with the Command task.
Node(s)	Nodes on which the commands in the Command task run.
Start Time	Start time of the Command task.
End Time	End time of the Command task.

Attribute Name	Description
Recovery Time(s)	Time of the recovery run.
Status	Status of the Command task.
Status Message	Message about the Command task status.
Deleted	Indicates if the Command task is deleted.
Version Number	Version number of the Command task.

Session Task Run Properties

When the Integration Service runs a session, the Workflow Monitor creates session details that provide load statistics for each target in the mapping. You can view session details when the session runs or after the session completes.

When you view session task properties, the following areas display in the Properties window:

- **Failure information.** View information about session failures.
- **Task details.** View information about the session.
- **Source and target statistics.** View information about the number of rows the Integration Service read from the source and wrote to the target.
- **Partition details.** View information about partitions in a session.
- **Performance.** View information about session performance.

To view session details, right-click a session in the Workflow Monitor and choose Get Run Properties.

When you load data to a target with multiple groups, such as an XML target, the Integration Service provides session details for each group.

Failure Information

The Failure Information area displays information about session errors.

The following table describes the attributes that appear in the Failure Information area:

Attribute Name	Description
First Error Code	Error code for the first error.
First Error	First error message.

Session Task Details

The Task Details area displays information about session task.

The following table describes the attributes that appear in the Task Details area:

Attribute Name	Description
Instance Name	Name of the session.
Task Type	Task type is Session.
Integration Service Name	Name of the Integration Service assigned to the workflow associated with the session.
Node(s)	Node on which the session is running.
Start Time	Start time of the session.
End Time	End time of the session.
Recovery Time(s)	Time of the recovery session run.
Status	Status of the session.
Status Message	Message about the session status.
Deleted	Indicates if the session is deleted from the repository.
Version Number	Version number of the session.
Mapping Name	Name of the mapping associated with the session.
Source Success Rows	Number of rows the Integration Service successfully read from the source.
Source Failed Rows	Number of rows the Integration Service failed to read from the source.
Target Success Rows[1]	Number of rows the Integration Service wrote to the target.
Target Failed Rows	Number of rows the Integration Service failed to write the target.
Total Transformation Errors	Number of transformation errors in the session.

[1]. *For a recovery session, this value lists the number of rows the Integration Service processed after recovery. To determine the number of rows processed before recovery, see the session log.*

Source and Target Statistics

The Source/Target Statistics area displays information about the rows the Integration Service read from the sources and the rows it loaded to the target.

The following table describes the attributes that appear in the Source/Target Statistics area:

Attribute Name	Description
Transformation Name	Name of the source qualifier instance or the target instance in the mapping. If you create multiple partitions in the source or target, the Instance Name displays the partition number. If the source or target contains multiple groups, the Instance Name displays the group name.
Node	Node running the transformation.
Applied Rows	For sources, shows the number of rows the Integration Service successfully read from the source. For targets, shows the number of rows the Integration Service successfully applied to the target. For example, you have a target table with one column called SALES_ID and five rows that contain the values 1, 2, 3, 2, and 2. You have a source table with one column called SALES_ID_IN and five rows that contain the values 1, 2, 3, 4, and 5. You mark rows for update where SALES_ID_IN is 2. The Integration Service applies one row, which updates three rows in the target. If you mark rows for update where SALES_ID_IN is 4, the Integration Service applies one row. The Integration Service does not update any rows at the target as the target does not contain rows with SALES_ID as 4. For a recovery session, this value lists the number of rows that the Integration Service affected or applied to the target after recovery. To determine the number of rows processed before recovery, see the session log.
Affected Rows	For sources, shows the number of rows the Integration Service successfully read from the source. For targets, shows the number of rows affected by the specified operation. For example, you have a table with one column called SALES_ID and five rows that contain the values 1, 2, 3, 2, and 2. You mark rows for update where SALES_ID is 2. The Integration Service updates three rows, even though there was one update request. If you mark rows for update where SALES_ID is 4, the Integration Service updates no rows. For a recovery session, this value lists the number of rows that the Integration Service affected or applied to the target after recovery. To determine the number of rows processed before recovery, see the session log.
Rejected Rows	Number of rows the Integration Service dropped when reading from the source, or the number of rows the Integration Service rejected when writing to the target.
Throughput (Rows/Sec)	Rate at which the Integration Service read rows from the source or wrote data into the target per second.
Throughput (Bytes/Sec)	Estimated rate at which the Integration Service read data from the source and wrote data to the target in bytes per second. Throughput (Bytes/Sec) is based on the Throughput (Rows/Sec) and the row size. The row size is based on the number of columns the Integration Service read from the source and wrote to the target, the data movement mode, column metadata, and if you enabled high precision for the session. The calculation is not based on the actual data size in each row.
Bytes	Total bytes processed in the PowerCenter Integration Service memory for the source and target.
Last Error Code	Error message code of the most recent error message written to the session log. If you view details after the session completes, this field displays the last error code.
Last Error Message	Most recent error message written to the session log. If you view details after the session completes, this field displays the last error message.

Attribute Name	Description
Start Time	Time the Integration Service started to read from the source or write to the target. The Workflow Monitor displays time relative to the Integration Service.
End Time	Time the Integration Service finished reading from the source or writing to the target. The Workflow Monitor displays time relative to the Integration Service.

Partition Details

The Partition Details area displays information about partitions in a session. When you create multiple partitions in a session, the Integration Service provides session details for each partition. Use these details to determine if the data is evenly distributed among the partitions. For example, if the Integration Service moves more rows through one target partition than another, or if the throughput is not evenly distributed, you might want to adjust the data range for the partitions.

The following table describes the attributes that appear in the Partition Details area:

Attribute Name	Description
Partition Name	Name of the partition.
Node	Node running the partition.
Transformations	Transformations in the partition pipeline.
Process ID	Process ID of the partition.
CPU %	Percent of the CPU the partition is consuming during the current session run.
CPU Seconds	Amount of process time in seconds the CPU is taking to process the data in the partition during the current session run.
Memory Usage	Amount of memory the partition consumes during the current session run.
Input Rows	Number of input rows for the partition.
Output Rows	Number of output rows for the partition. For a recovery session, this value lists the number of rows the Integration Service processed after recovery. To determine the number of rows processed before recovery, see the session log.

Performance Details

The performance details provide counters that help you understand the session and mapping efficiency. Each source qualifier and target definition appears in the performance details, along with counters that display performance information about each transformation. You can view session performance details in the Workflow Monitor or in the performance details file.

By evaluating the final performance details, you can determine where session performance slows down. The Workflow Monitor also provides session-specific details that can help tune the following memory settings:

- Buffer block size
- Index and data cache size for Aggregator, Rank, Lookup, and Joiner transformations

Viewing Performance Details in the Workflow Monitor

When you configure the session to collect performance details, you can view them in the Workflow Monitor. When you configure the session to store performance details, you can view the details for previous sessions.

To view performance details in the Workflow Monitor:

1. Right-click a session in the Workflow Monitor and choose Get Run Properties.
2. Click the Performance area in the Properties window.

   The following table describes the attributes that appear in the Performance area:

Attribute Name	Description
Performance Counter	Name of the performance counter.
Counter Value	Value of the performance counter.

   When you create multiple partitions, the Performance Area displays a column for each partition. The columns display the counter values for each partition.
3. Click OK.

Viewing Performance Details in the Performance Details File

The Integration Service creates a performance detail file for the session when it completes. Use a text editor to view the performance details file.

1. Locate the performance details file.

   The Integration Service names the file *session_name*.perf, and stores it in the same directory as the session log. If there is no session-specific directory for the session log, the Integration Service saves the file in the default log files directory.
2. Open the file in any text editor.

Understanding Performance Counters

All transformations have some basic counters that indicate the number of input rows, output rows, and error rows.

Source Qualifier, Normalizer, and target transformations have additional counters that indicate the efficiency of data moving into and out of buffers. Use these counters to locate performance bottlenecks.

Some transformations have counters specific to their functionality. For example, each Lookup transformation has a counter that indicates the number of rows stored in the lookup cache.

When you view the performance details file, the first column displays the transformation name as it appears in the mapping, the second column contains the counter name, and the third column holds the resulting number or efficiency percentage. If you use a Joiner transformation, the first column shows two instances of the Joiner transformation:

- ***<Joiner transformation>* [M]**. Displays performance details about the master pipeline of the Joiner transformation.
- ***<Joiner transformation>* [D]**. Displays performance details about the detail pipeline of the Joiner transformation.

When you create multiple partitions, the Integration Service generates one set of counters for each partition. The following performance counters illustrate two partitions for an Expression transformation:

```
Transformation Name          Counter Name                   Counter Value

EXPTRANS [1]                 Expression_input rows          8

EXPTRANS [1]                 Expression_output rows         8

EXPTRANS [2]                 Expression_input rows          16

EXPTRANS [2]                 Expression_output rows         16
```

Note: When you increase the number of partitions, the number of aggregate or rank input rows may be different from the number of output rows from the previous transformation.

The following table describes the Aggegator and Rank Transformation counters/descriptions that may appear in the Session Performance Details area or in the performance details file:

Counters	Description
*Aggregator/Rank*_inputrows	Number of rows passed into the transformation.
*Aggregator/Rank*_outputrows	Number of rows sent out of the transformation.
*Aggregator/Rank*_errorrows	Number of rows in which the Integration Service encountered an error.
*Aggregator/Rank*_readfromcache	Number of times the Integration Service read from the index or data cache.
*Aggregator/Rank*_writetocache	Number of times the Integration Service wrote to the index or data cache.
*Aggregator/Rank*_readfromdisk	Number of times the Integration Service read from the index or data file on the local disk, instead of using cached data.
*Aggregator/Rank*_writetodisk	Number of times the Integration Service wrote to the index or data file on the local disk, instead of using cached data.
*Aggregator/Rank*_newgroupkey	Number of new groups the Integration Service created.
*Aggregator/Rank*_oldgroupkey	Number of times the Integration Service used existing groups.

The following table describes the Lookup Transformation counters/descriptions that may appear in the Session Performance Details area or in the performance details file:

Counters	Description
*Lookup*_inputrows	Number of rows passed into the transformation.
*Lookup*_outputrows	Number of rows sent out of the transformation.
*Lookup*_errorrows	Number of rows in which the Integration Service encountered an error.
*Lookup*_rowsinlookupcache	Number of rows stored in the lookup cache.

The following table describes the Master and Detail Joiner Transformation counters/descriptions that may appear in the Session Performance Details area or in the performance details file:

Counters	Description
*Joiner*_inputMasterRows	Number of rows the master source passed into the transformation.
*Joiner*_inputDetailRows	Number of rows the detail source passed into the transformation.
*Joiner*_outputrows	Number of rows sent out of the transformation.
*Joiner*_errorrows	Number of rows in which the Integration Service encountered an error.
*Joiner*_readfromcache	Number of times the Integration Service read from the index or data cache.
*Joiner*_writetocache	Number of times the Integration Service wrote to the index or data cache.
*Joiner*_readfromdisk	Number of times the Integration Service read from the index or data files on the local disk, instead of using cached data. The Integration Service generates this counter when you use sorted input for the Joiner transformation.
*Joiner*_writetodisk	Number of times the Integration Service wrote to the index or data files on the local disk, instead of using cached data. The Integration Service generates this counter when you use sorted input for the Joiner transformation.
*Joiner*_readBlockFromDisk	Number of times the Integration Service read from the index or data files on the local disk, instead of using cached data. The Integration Service generates this counter when you do not use sorted input for the Joiner transformation.
*Joiner*_writeBlockToDisk	Number of times the Integration Service wrote to the index or data cache. The Integration Service generates this counter when you do not use sorted input for the Joiner transformation.

Counters	Description
*Joiner*_seekToBlockInDisk	Number of times the Integration Service accessed the index or data files on the local disk. The Integration Service generates this counter when you do not use sorted input for the Joiner transformation.
*Joiner*_insertInDetailCache	Number of times the Integration Service wrote to the detail cache. The Integration Service generates this counter if you join data from a single source. The Integration Service generates this counter when you use sorted input for the Joiner transformation.
*Joiner*_duplicaterows	Number of duplicate rows the Integration Service found in the master relation.
*Joiner*_duplicaterowsused	Number of times the Integration Service used the duplicate rows in the master relation.

The following table describes All Other Transformation counters/descriptions that may appear in the Session Performance Details area or in the performance details file:

Counters	Description
*Transformation*_inputrows	Number of rows passed into the transformation.
*Transformation*_outputrows	Number of rows sent out of the transformation.
*Transformation*_errorrows	Number of rows in which the Integration Service encountered an error.

If you have multiple source qualifiers and targets, evaluate them as a whole. For source qualifiers and targets, a high value is considered 80-100 percent. Low is considered 0-20 percent.

Session and Workflow Logs

Session and Workflow Logs Overview

The Service Manager provides accumulated log events from each service in the domain and for sessions and workflows. To perform the logging function, the Service Manager runs a Log Manager and a Log Agent. The Log Manager runs on the master gateway node. The Integration Service generates log events for workflows and sessions. The Log Agent runs on the nodes to collect and process log events for sessions and workflows.

Log events for workflows include information about tasks performed by the Integration Service, workflow processing, and workflow errors. Log events for sessions include information about the tasks performed by the Integration Service, session errors, and load summary and transformation statistics for the session.

You can view log events for workflows with the Log Events window in the Workflow Monitor. The Log Events window displays information about log events including severity level, message code, run time, workflow name, and session name. For session logs, you can set the tracing level to log more information. All log events display severity regardless of tracing level.

The following steps describe how the Log Manager processes session and workflow logs:

1. The Integration Service writes binary log files on the node. It sends information about the sessions and workflows to the Log Manager.
2. The Log Manager stores information about workflow and session logs in the domain configuration database. The domain configuration database stores information such as the path to the log file location, the node that contains the log, and the Integration Service that created the log.
3. When you view a session or workflow in the Log Events window, the Log Manager retrieves the information from the domain configuration database to determine the location of the session or workflow logs.
4. The Log Manager dispatches a Log Agent to retrieve the log events on each node to display in the Log Events window.

To access log events for more than the last workflow run, you can configure sessions and workflows to archive logs by time stamp. You can also configure a workflow to produce text log files. You can archive text log files by run or by time stamp. When you configure the workflow or session to produce text log files, the Integration Service creates the binary log and the text log file.

You can limit the size of session logs for long-running and real-time sessions. You can limit the log size by configuring a maximum time frame or a maximum file size. When a log reaches the maximum size, the Integration Service starts a new log.

Log Events

You can view log events in the Workflow Monitor Log Events window and you can view them as text files. The Log Events window displays log events in a tabular format.

Log Codes

Use log events to determine the cause of workflow or session problems. To resolve problems, locate the relevant log codes and text prefixes in the workflow and session log.

The Integration Service precedes each workflow and session log event with a thread identification, a code, and a number. The code defines a group of messages for a process. The number defines a message. The message can provide general information or it can be an error message.

Some log events are embedded within other log events. For example, a code CMN_1039 might contain informational messages from Microsoft SQL Server.

Message Severity

The Log Events window categorizes workflow and session log events into severity levels. It prioritizes error severity based on the embedded message type. The error severity level appears with log events in the Log Events window in the Workflow Monitor. It also appears with messages in the workflow and session log files.

Note: If you cannot view all the workflow log messages when the error severity level is at warning, change the error severity level of the workflow log. Change the log level from warning to info in the advanced properties of the PowerCenter Integration Service process.

The following table describes message severity levels:

Severity Level	Description
FATAL	Fatal error occurred. Fatal error messages have the highest severity level.
ERROR	Indicates the service failed to perform an operation or respond to a request from a client application. Error messages have the second highest severity level.
WARNING	Indicates the service is performing an operation that may cause an error. This can cause repository inconsistencies. Warning messages have the third highest severity level.
INFO	Indicates the service is performing an operation that does not indicate errors or problems. Information messages have the third lowest severity level.
TRACE	Indicates service operations at a more specific level than Information. Tracing messages are generally record message sizes. Trace messages have the second lowest severity level.
DEBUG	Indicates service operations at the thread level. Debug messages generally record the success or failure of service operations. Debug messages have the lowest severity level.

Writing Logs

The Integration Service writes the workflow and session logs as binary files on the node where the service process runs. It adds a .bin extension to the log file name you configure in the session and workflow properties.

When you run a session on a grid, the Integration Service creates one session log for each DTM process. The log file on the primary node has the configured log file name. The log file on a worker node has a .w<Partition Group Id> extension:

```
<session or workflow name>.w<Partition Group ID>.bin
```

For example, if you run the session s_m_PhoneList on a grid with three nodes, the session log files use the names, s_m_PhoneList.bin, s_m_PhoneList.w1.bin, and s_m_PhoneList.w2.bin.

When you rerun a session or workflow, the Integration Service overwrites the binary log file unless you choose to save workflow logs by time stamp. When you save workflow logs by time stamp, the Integration Service adds a time stamp to the log file name and archives them.

To view log files for more than one run, configure the workflow or session to create log files.

A workflow or session continues to run even if there are errors while writing to the log file after the workflow or session initializes. If the log file is incomplete, the Log Events window cannot display all the log events.

The Integration Service starts a new log file for each workflow and session run. When you recover a workflow or session, the Integration Service appends a recovery.time stamp extension to the file name for the recovery run.

For real-time sessions, the Integration Service overwrites the log file when you restart a session in cold start mode or when you restart a JMS or WebSphere MQ session that does not have recovery data. The

Integration Service appends the log file when you restart a JMS or WebSphere MQ session that has recovery data.

To convert the binary file to a text file, use the *infacmd* convertLog or the *infacmd* GetLog command.

Passing Session Events to an External Library

You can pass session event messages to an external procedure to handle. You write the procedure according to how you want to handle the events and compile it into a shared library. The shared library must implement a set of functions in the Session Log Interface provided by PowerCenter. In the Administrator tool, you configure the Integration Service to use the shared library to handle the session logs.

The Session Log Interface lets you pass session event messages, but not workflow event messages, to an external shared library.

Log Events Window

View log events in the Log Events window. The Log Events window displays the following information for each session and workflow:

- **Severity.** Lists the type of message, such as informational or error.
- **Time stamp.** Date and time the log event reached the Log Agent.
- **Node.** Node on which the Integration Service process is running.
- **Thread.** Thread ID for the workflow or session.
- **Process ID.** Windows or UNIX process identification numbers. Displays in the Output window only.
- **Message Code.** Message code and number.
- **Message.** Message associated with the log event.

By default, the Log Events window displays log events according to the date and time the Integration Service writes the log event on the node. The Log Events window displays logs consisting of multiple log files by node name. When you run a session on a grid, log events for the partition groups are ordered by node name and grouped by log file.

You can perform the following tasks in the Log Events window:

- **Save log events to file.** Click Save As to save log events as a binary, text, or XML file.
- **Copy log event text to a file.** Click Copy to copy one or more log events and paste them into a text file.
- **Sort log events.** Click a column heading to sort log events.
- **Search for log events.** Click Find to search for text in log events.
- **Refresh log events.** Click Refresh to view updated log events during a workflow or session run.

Note: When you view a log larger than 2 GB, the Log Events window displays a warning that the file might be too large for system memory. If you continue, the Log Events window might shut down unexpectedly.

Searching for Log Events

Search for log events based on any information in the Log Events window. For example, you can search for text in a message or search for messages based on the date and time of the log event.

To search for log events:

1. Open the Workflow Monitor.
2. Connect to a repository in the Navigator.
3. Select an Integration Service.
4. Right-click a workflow and select Get Workflow Log.

   The Log Events window displays.
5. In the Log Events window, click Find.

   The Query Area appears.
6. Enter the text you want to find.
7. Optionally, click Match Case if you want the query to be case sensitive.
8. Select Message to search text in the Message field.

   -or-

   Select All Fields to search text in all fields.
9. Click Find Next to search for the next instance of the text in the Log Events. Or, click Find Previous to search for the previous instance of the text in the Log Events.

Keyboard Shortcuts for the Log Events Window

The following table lists shortcuts that you can use to search for Log Events:

To	Press
Open the Query Area.	Ctrl+ F
Find next instance of the text.	F3
Find the previous instance of the text.	Shift+F3

Working with Log Files

Configure a workflow or session to write log events to log files in the workflow or session properties. The Integration Service writes information about the workflow or session run to a text file in addition to writing log events to a binary file. If you configure workflow or session properties to create log files, you can open the text files with any text editor or import the binary files to view logs in the Log Events window.

By default, the Integration Service writes log files based on the Integration Service code page. If you enable the LogInUTF8 option in the Advanced Properties for the Integration Service, the Integration Service writes to the logs using the UTF-8 character set. If you configure the Integration Service to run in ASCII mode, it sorts all character data using a binary sort order even if you select a different sort order in the session properties.

Optimize performance by disabling the option to create text log files.

Writing to Log Files

When you create a workflow or session log, you can configure log options in the workflow or session properties. You can configure the following information for a workflow or session log:

- **Write Backward Compatible Log File.** Select this option to create a text file for workflow or session logs. If you do not select the option, the Integration Service creates the binary log only.
- **Log File Directory.** The directory where you want the log file created. By default, the Integration Service writes the workflow log file in the directory specified in the service process variable, $PMWorkflowLogDir. It writes the session log file in the directory specified in the service process variable, $PMSessionLogDir. If you enter a directory name that the Integration Service cannot access, the workflow or session fails.

  The following table shows the default location for each type of log file and the associated service process variables:

Log File Type	Default Directory (Service Process Variable)	Default Value for Service Process Variable
Workflow logs	$PMWorkflowLogDir	$PMRootDir/WorkflowLogs
Session logs	$PMSessionLogDir	$PMRootDir/SessLogs

- **Name.** The name of the log file. You must configure a name for the log file or the workflow or session is invalid. You can use a service, service process, or user-defined workflow or worklet variable for the log file name.

Note: The Integration Service stores the workflow and session log names in the domain configuration database. If you want to use Unicode characters in the workflow or session log file names, the domain configuration database must be a Unicode database.

Archiving Log Files

By default, when you configure a workflow or session to create log files, the Integration Service creates one log file for the workflow or session. The Integration Service overwrites the log file when you run the workflow again.

To create a log file for more than one workflow or session run, configure the workflow or session to archive logs in the following ways:

- **By run.** Archive text log files by run. Configure a number of text logs to save.
- **By time stamp**. Archive binary logs and text files by time stamp. The Integration Service saves an unlimited number of logs and labels them by time stamp. When you configure the workflow or session to archive by time stamp, the Integration Service always archives binary logs.

Note: When you run concurrent workflows with the same instance name, the Integration Service appends a timestamp to the log file name, even if you configure the workflow to archive logs by run.

Archiving Logs by Run

If you archive log files by run, specify the number of text log files you want the Integration Service to create. The Integration Service creates the number of historical log files you specify, plus the most recent log file. If you specify five runs, the Integration Service creates the most recent workflow log, plus historical logs zero to four, for a total of six logs. You can specify up to 2,147,483,647 historical logs. If you specify zero logs, the Integration Service creates only the most recent workflow log file.

The Integration Service uses the following naming convention to create historical logs:

```
<session or workflow name>.n
```

where n=0 for the first historical log. The variable increments by one for each workflow or session run.

If you run a session on a grid, the worker service processes use the following naming convention for a session:

```
<session name>.n.w<DTM ID>
```

Archiving Log Files by Time Stamp

When you archive logs by time stamp, the Integration Service creates an unlimited number of binary and text file logs. The Integration Service adds a time stamp to the text and binary log file names. It appends the year, month, day, hour, and minute of the workflow or session completion to the log file. The resulting log file name is <session or workflow log name>.*yyyymmddhhmi*, where:

- *yyyy* = year
- *mm* = month, ranging from 01-12
- *dd* = day, ranging from 01-31
- *hh* = hour, ranging from 00-23
- *mi* = minute, ranging from 00-59

Binary logs use the .bin suffix.

To prevent filling the log directory, periodically purge or back up log files when using the time stamp option.

If you run a session on a grid, the worker service processes use the following naming convention for sessions:

```
<session name>.yyyymmddhhmi.w<DTM ID>
<session name>.yyyymmddhhmi.w<DTM ID>.bin
```

When you archive text log files, view the logs by navigating to the workflow or session log folder and viewing the files in a text reader. When you archive binary log files, you can view the logs by navigating to the workflow or session log folder and importing the files in the Log Events window. You can archive binary files when you configure the workflow or session to archive logs by time stamp. You do not have to create text log files to archive binary files. You might need to archive binary files to send to Informatica Global Customer Support for review.

Session Log Rollover

You can limit the size of session log files for real-time sessions. Configure a maximum log file size for the session log. When the session log reaches a maximum size, the Integration Service creates a new log file and writes the session logs to the new log file. When the session log is contained in multiple log files, each file is a partial log.

Configure the session log to roll over to a new file after the log file reaches a maximum size. Or, configure the session log to roll over to a new file after a maximum period of time. The Integration Service saves the previous log files.

You can configure the maximum number of partial log files to save for the session. The Integration Service saves one more log file that the number of files you configure. The Integration Service does not purge the first session log file. The first log file contains details about the session initialization.

The Integration Service names each partial session log file with the following syntax:

```
<session log file>.part.n
```

Configure the following attributes on the Advanced settings of the Config Object tab:

- **Session Log File Max Size**. The maximum number of megabytes for a log file. Configure a maximum size to enable log file rollover by file size. When the log file reaches the maximum size, the Integration Service creates a new log file. Default is zero.
- **Session Log File Max Time Period**. The maximum number of hours that the Integration Service writes to a session log. Configure the maximum time period to enable log file rollover by time. When the period is over, the Integration service creates another log file. Default is zero.
- **Maximum Partial Session Log Files.** Maximum number of session log files to save. The Integration Service overwrites the oldest partial log file if the number of log files has reached the limit. If you configure a maximum of zero, then the number of session log files is unlimited. Default is one.

Note: You can configure a combination of log file maximum size and log file maximum time. You must configure one of the properties to enable session log file rollover. If you configure only maximum partial session log files, log file rollover is not enabled.

Configuring Workflow Log File Information

You can configure workflow log information on the workflow Properties tab.

1. Select the Properties tab of a workflow.
2. Enter the following workflow log options:

Option Name	Description
Write Backward Compatible Workflow Log File	Writes workflow logs to a text log file. Select this option if you want to create a log file in addition to the binary log for the Log Events window.
Workflow Log File Name	Enter a file name or a file name and directory. You can use a service, service process, or user-defined workflow or worklet variable for the workflow log file name. The Integration Service appends this value to that entered in the Workflow Log File Directory field. For example, if you have $PMWorkflowLogDir\ in the Workflow Log File Directory field, enter “logname.txt” in the Workflow Log File Name field, the Integration Service writes logname.txt to the $PMWorkflowLogDir\ directory.

Option Name	Description
Workflow Log File Directory	Location for the workflow log file. By default, the Integration Service writes the log file in the process variable directory, $PMWorkflowLogDir. If you enter a full directory and file name in the Workflow Log File Name field, clear this field.
Save Workflow Log By	You can create workflow logs according to the following options: - By Runs. The Integration Service creates a designated number of workflow logs. Configure the number of workflow logs in the Save Workflow Log for These Runs option. The Integration Service does not archive binary logs. - By time stamp. The Integration Service creates a log for all workflows, appending a time stamp to each log. When you save workflow logs by time stamp, the Integration Service archives binary logs and workflow log files. You can also use the $PMWorkflowLogCount service variable to create the configured number of workflow logs for the Integration Service.
Save Workflow Log for These Runs	Number of historical workflow logs you want the Integration Service to create. The Integration Service creates the number of historical logs you specify, plus the most recent workflow log.

3. Click OK.

Configuring Session Log File Information

You can configure session log information on the session Properties tab and the Config Object tab.

1. Select the Properties tab of a session.
2. Enter the following session log options:

Option Name	Description
Write Backward Compatible Session Log File	Writes session logs to a log file. Select this option if you want to create a log file in addition to the binary log for the Log Events window.
Session Log File Name	By default, the Integration Service uses the session name for the log file name: *s_mapping name*.log. For a debug session, it uses DebugSession_*mapping name*.log. Enter a file name, a file name and directory, or use the $PMSessionLogFile session parameter. The Integration Service appends information in this field to that entered in the Session Log File Directory field. For example, if you have "C:\session_logs\" in the Session Log File Directory File field and then enter "logname.txt" in the Session Log File field, the Integration Service writes the logname.txt to the C:\session_logs\ directory. You can also use the $PMSessionLogFile session parameter to represent the name of the session log or the name and location of the session log.
Session Log File Directory	Location for the session log file. By default, the Integration Service writes the log file in the process variable directory, $PMSessionLogDir. If you enter a full directory and file name in the Session Log File Name field, clear this field.

3. Click the Config Object tab.

4. Enter the following session log options:

Option Name	Description
Save Session Log By	You can create session logs according to the following options: - Session Runs. The Integration Service creates a designated number of session log files. Configure the number of session logs in the Save Session Log for These Runs option. The Integration Service does not archive binary logs. - Session Time Stamp. The Integration Service creates a log for all sessions, appending a time stamp to each log. When you save a session log by time stamp, the Integration Service archives the binary logs and text log files. You can also use the $PMSessionLogCount service variable to create the configured number of session logs for the Integration Service.
Save Session Log for These Runs	Number of historical session logs you want the Integration Service to create. The Integration Service creates the number of historical logs you specify, plus the most recent session log.

5. Click OK.

Workflow Logs

Workflow logs contain information about the workflow runs. You can view workflow log events in the Log Events window of the Workflow Monitor. You can also create an XML, text, or binary log file for workflow log events.

A workflow log contains the following information:

- Workflow name
- Workflow status
- Status of tasks and worklets in the workflow
- Start and end times for tasks and worklets
- Results of link conditions
- Errors encountered during the workflow and general information
- Some session messages and errors

Workflow Log Events Window

Use the Log Events window in the Workflow Monitor to view log events for a workflow. The Log Events window displays all log events for a workflow. Select a log event to view more information about the log event.

Workflow Log Sample

A workflow log file provides the same information as the Log Events window for a workflow. You can view a workflow log file in a text editor.

The following sample shows a section of a workflow log file:

```
INFO : LM_36435 [Mon Apr 03 15:10:20 2006] : (3060|3184) Starting execution of
workflow [Wk_Java] in folder [EmployeeData] last saved by user [ellen].
```

```
INFO : LM_36330 [Mon Apr 03 15:10:20 2006] : (3060|3184) Start task instance
[Start]: Execution started.
INFO : LM_36318 [Mon Apr 03 15:10:20 2006] : (3060|3184) Start task instance
[Start]: Execution succeeded.
INFO : LM_36505 : (3060|3184) Link [Start --> s_m_jtx_hier_useCase]: empty
expression string, evaluated to TRUE.
INFO : LM_36388 [Mon Apr 03 15:10:20 2006] : (3060|3184) Session task instance
[s_m_jtx_hier_useCase] is waiting to be started.
INFO : LM_36682 [Mon Apr 03 15:10:20 2006] : (3060|3184) Session task instance
[s_m_jtx_hier_useCase]: started a process with pid [148] on node [garnet].
INFO : LM_36330 [Mon Apr 03 15:10:20 2006] : (3060|3184) Session task instance
[s_m_jtx_hier_useCase]: Execution started.
INFO : LM_36488 [Mon Apr 03 15:10:22 2006] : (3060|3180) Session task instance
[s_m_jtx_hier_useCase] : [TM_6793 Fetching initialization properties from the
Integration Service. : (Mon Apr 03 15:10:21 2006)]
INFO : LM_36488 [Mon Apr 03 15:10:22 2006] : (3060|3180) Session task instance
[s_m_jtx_hier_useCase] : [DISP_20305 The [Preparer] DTM with process id [148] is
running on node [garnet].
 : (Mon Apr 03 15:10:21 2006)]
INFO : LM_36488 [Mon Apr 03 15:10:22 2006] : (3060|3180) Session task instance
[s_m_jtx_hier_useCase] : [PETL_24036 Beginning the prepare phase for the session.]
INFO : LM_36488 [Mon Apr 03 15:10:22 2006] : (3060|3180) Session task instance
[s_m_jtx_hier_useCase] : [TM_6721 Started [Connect to Repository].]
```

Session Logs

Session logs contain information about the tasks that the Integration Service performs during a session, plus load summary and transformation statistics. By default, the Integration Service creates one session log for each session it runs. If a workflow contains multiple sessions, the Integration Service creates a separate session log for each session in the workflow. When you run a session on a grid, the Integration Service creates one session log for each DTM process.

In general, a session log contains the following information:

- Allocation of heap memory
- Execution of pre-session commands
- Creation of SQL commands for reader and writer threads
- Start and end times for target loading
- Errors encountered during the session and general information
- Execution of post-session commands
- Load summary of reader, writer, and DTM statistics
- Integration Service version and build number

Related Topics:

- “Log Options Settings” on page 45

Log Events Window

Use the Log Events window in the Workflow Monitor to view log events for a session. The Log Events window displays all log events for a session. Select a log event to view more information about the log event.

Session Log File Sample

A session log file provides most of the same information as the Log Events window for a session. The session log file does not include severity or DTM prepare messages.

The following sample shows a section of a session log file:

```
DIRECTOR> PETL_24044 The Master DTM will now connect and fetch the prepared session
from the Preparer DTM.
DIRECTOR> PETL_24047 The Master DTM has successfully fetched the prepared session
from the Preparer DTM.
DIRECTOR> DISP_20305 The [Master] DTM with process id [2968] is running on node
[sapphire].
 : (Mon Apr 03 16:19:47 2006)
DIRECTOR> TM_6721 Started [Connect to Repository].
DIRECTOR> TM_6722 Finished [Connect to Repository].  It took [0.656233] seconds.
DIRECTOR> TM_6794 Connected to repository [HR_80] in domain [StonesDomain] user
[ellen]
DIRECTOR> TM_6014 Initializing session [s_PromoItems] at [Mon Apr 03 16:19:48 2006]
DIRECTOR> TM_6683 Repository Name: [HR_80]
DIRECTOR> TM_6684 Server Name: [Copper]
DIRECTOR> TM_6686 Folder: [Snaps]
DIRECTOR> TM_6685 Workflow: [wf_PromoItems]
DIRECTOR> TM_6101 Mapping name: m_PromoItems [version 1]
DIRECTOR> SDK_1805 Recovery cache will be deleted when running in normal mode.
DIRECTOR> SDK_1802 Session recovery cache initialization is complete.
```

The session log file includes the Integration Service version and build number.

```
DIRECTOR> TM_6703 Session [s_PromoItems] is run by 32-bit Integration Service
[sapphire], version [8.1.0], build [0329].
```

Tracing Levels

The amount of detail that logs contain depends on the tracing level that you set. You can configure tracing levels for each transformation or for the entire session. By default, the Integration Service uses tracing levels configured in the mapping.

Setting a tracing level for the session overrides the tracing levels configured for each transformation in the mapping. If you select a normal tracing level or higher, the Integration Service writes row errors into the session log, including the transformation in which the error occurred and complete row data. If you configure the session for row error logging, the Integration Service writes row errors to the error log instead of the session log. If you want the Integration Service to write dropped rows to the session log also, configure the session for verbose data tracing.

Set the tracing level on the Config Object tab in the session properties.

The following table describes the session log tracing levels:

Tracing Level	Description
None	Integration Service uses the tracing level set in the mapping.
Terse	Integration Service logs initialization information, error messages, and notification of rejected data.

Tracing Level	Description
Normal	Integration Service logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes session results, but not at the level of individual rows.
Verbose Initialization	In addition to normal tracing, the Integration Service logs additional initialization details, names of index and data files used, and detailed transformation statistics.
Verbose Data	In addition to verbose initialization tracing, the Integration Service logs each row that passes into the mapping. Also notes where the Integration Service truncates string data to fit the precision of a column and provides detailed transformation statistics. When you configure the tracing level to verbose data, the Integration Service writes row data for all rows in a block when it processes a transformation.

You can also enter tracing levels for individual transformations in the mapping. When you enter a tracing level in the session properties, you override tracing levels configured for transformations in the mapping.

Viewing Log Events

The Integration Service generates log events when you run a session or workflow. You can view log events in the following types of log files:

- Most recent session or workflow log
- Archived binary log files
- Archived text log files

Viewing the Log Events Window

You can view the session or workflow log in the Log Events window for the last run workflow.

1. In the Workflow Monitor, right-click the workflow or session.
2. Select Get Session Log or Get Workflow Log.

Viewing an Archived Binary Log File

You can view archived binary log files in the Log Events window.

1. If you do not know the session or workflow log file name and location, check the Log File Name and Log File Directory attributes on the Session or Workflow Properties tab.

   If you are running the Integration Service on UNIX and the binary log file is not accessible on the Windows machine where the PowerCenter client is running, you can transfer the binary log file to the Windows machine using FTP.
2. In the Workflow Monitor, click Tools > Import Log.
3. Navigate to the session or workflow log file directory.
4. Select the binary log file you want to view.
5. Click Open.

Viewing a Text Log File

You can view text log files in any text editor.

1. If you do not know the session or workflow log file name and location, check the Log File Name and Log File Directory attributes on the Session or Workflow Properties tab.
2. Navigate to the session or workflow log file directory.

   The session and workflow log file directory contains the text log files and the binary log files. If you archive log files, check the file date to find the latest log file for the session.
3. Open the log file in any text editor.

Session Properties Reference

General Tab

The following table describes settings on the General tab:

General Tab Options	Description
Rename	You can enter a new name for the session task with the Rename button.
Description	You can enter a description for the session task in the Description field.
Mapping name	Name of the mapping associated with the session task.
Resources	You can associate an object with an available resource.
Fail Parent if This Task Fails	Fails the parent worklet or workflow if this task fails. Appears only in the Workflow Designer.
Fail Parent if This Task Does Not Run	Fails the parent worklet or workflow if this task does not run. Appears only in the Workflow Designer.
Disable This Task	Disables the task. Appears only in the Workflow Designer.
Treat the Input Links as AND or OR	Runs the task when all or one of the input link conditions evaluate to True. Appears only in the Workflow Designer.

Properties Tab

On the Properties tab, you can configure the following settings:

- **General Options.** General Options settings allow you to configure session log file name, session log file directory, parameter file name and other general session settings.
- **Performance.** The Performance settings allow you to increase memory size, collect performance details, and set configuration parameters.

General Options Settings

The following table describes the General Options settings on the Properties tab:

General Options Settings	Description
Write Backward Compatible Session Log File	Writes the session log to a file.
Session Log File Name	Enter a file name, a file name and directory, or use the $PMSessionLogFile session parameter. The Integration Service appends information in this field to that entered in the Session Log File Directory field. For example, if you have "C:\session_logs\" in the Session Log File Directory File field and then enter "logname.txt" in the Session Log File field, the Integration Service writes the logname.txt to the C:\session_logs\ directory.
Session Log File Directory	Location for the session log file. By default, the Integration Service writes the log file in the service process variable directory, $PMSessionLogDir. If you enter a full directory and file name in the Session Log File Name field, clear this field.
Parameter File Name	The name and directory for the parameter file. Use the parameter file to define session parameters and override values of mapping parameters and variables. You can enter a workflow or worklet variable as the session parameter file name if you configure a workflow to run concurrently, and you want to use different parameter files for the sessions in each workflow run instance.
Enable Test Load	You can configure the Integration Service to perform a test load. With a test load, the Integration Service reads and transforms data without writing to targets. The Integration Service generates all session files and performs all pre- and post-session functions, as if running the full session. Enter the number of source rows you want to test in the Number of Rows to Test field.
Number of Rows to Test	Enter the number of source rows you want the Integration Service to test load.
$Source Connection Value	The database connection you want the Integration Service to use for the $Source connection variable. You can select a relational or application connection object, or you can use the $DBConnection*Name* or $AppConnection*Name* session parameter if you want to define the connection value in a parameter file.
$Target Connection Value	The database connection you want the Integration Service to use for the $Target connection variable. You can select a relational or application connection object, or you can use the $DBConnection*Name* or $AppConnection*Name* session parameter if you want to define the connection value in a parameter file.
Treat Source Rows As	Indicates how the Integration Service treats all source rows. If the mapping for the session contains an Update Strategy transformation or a Custom transformation configured to set the update strategy, the default option is Data Driven. When you select Data Driven and you load to either a Microsoft SQL Server or Oracle database, you must use a normal load. If you bulk load, the Integration Service fails the session.

General Options Settings	Description
Commit Type	Determines if the Integration Service uses a source- or target-based, or user-defined commit. You can choose source- or target-based commit if the mapping has no Transaction Control transformation or only ineffective Transaction Control transformations. By default, the Integration Service performs a target-based commit. A user-defined commit is enabled by default if the mapping has effective Transaction Control transformations.
Commit Interval	In conjunction with the selected commit interval type, indicates the number of rows. By default, the Integration Service uses a commit interval of 10,000 rows. This option is not available for user-defined commit.
Commit On End of File	By default, this option is enabled and the Integration Service performs a commit at the end of the file. Clear this option if you want to roll back open transactions. This option is enabled by default for a target-based commit. You cannot disable it.
Rollback Transactions on Errors	The Integration Service rolls back the transaction at the next commit point when it encounters a non-fatal writer error.
Recovery Strategy	Choose one of the following recovery strategies: - Resume from the last checkpoint. The Integration Service saves the session state of operation and maintains target recovery tables. - Restart. The Integration Service runs the session again when it recovers the workflow. - Fail session and continue the workflow. The Integration Service cannot recover the session, but it continues the workflow. This is the default session recovery strategy.
Java Classpath	If you enter a Java Classpath in this field, the Java Classpath is added to the beginning of the system classpath when the Integration Service runs the session. Use this option if you use third-party Java packages, built-in Java packages, or custom Java packages in a Java transformation. You can use service process variables to define the classpath. For example, you can use $PMRootDir to define a classpath within the $PMRootDir folder.

Performance Settings

The following table describes the Performance settings on the Properties tab:

Performance Settings	Description
DTM Buffer Size	Amount of memory allocated to the session from the DTM process. By default, the PowerCenter Integration Service determines the DTM buffer size at run time. The Workflow Manager allocates a minimum of 12 MB for DTM buffer memory. You can specify auto or a numeric value. If you enter 2000, the PowerCenter Integration Service interprets the number as 2000 bytes. Append KB, MB, or GB to the value to specify other units. For example, you can specify 512MB. Increase the DTM buffer size in the following circumstances: - A session contains large amounts of character data and you configure it to run in Unicode mode. Increase the DTM buffer size to 24MB. - A session contains *n* partitions. Increase the DTM buffer size to at least *n* times the value for the session with one partition. - A source contains a large binary object with a precision larger than the allocated DTM buffer size. Increase the DTM buffer size so that the session does not fail.
Collect Performance Data	Collects performance details when the session runs. Use the Workflow Monitor to view performance details while the session runs.
Write Performance Data to Repository	Writes performance details for the session to the PowerCenter repository. Write performance details to the repository to view performance details for previous session runs. Use the Workflow Monitor to view performance details for previous session runs.
Incremental Aggregation	The PowerCenter Integration Service performs incremental aggregation.
Reinitialize Aggregate Cache	Overwrites existing aggregate files for an incremental aggregation session.
Enable High Precision	Processes the Decimal datatype to a precision of 28.
Session Retry On Deadlock	The PowerCenter Integration Service retries target writes on deadlock for normal load. You can configure the PowerCenter Integration Service to set the number of deadlock retries and the deadlock sleep time period.
Pushdown Optimization	The PowerCenter Integration Service analyzes the transformation logic, mapping, and session configuration to determine the transformation logic it can push to the database. Select one of the following pushdown optimization values: - None. The PowerCenter Integration Service does not push any transformation logic to the database. - To Source. The PowerCenter Integration Service pushes as much transformation logic as possible to the source database. - To Target. The PowerCenter Integration Service pushes as much transformation logic as possible to the target database. - Full. The PowerCenter Integration Service pushes as much transformation logic as possible to both the source database and target database. - $$PushdownConfig. The $$PushdownConfig mapping parameter allows you to run the same session with different pushdown optimization configurations at different times. Default is None.

Performance Settings	Description
Allow Temporary View for Pushdown	Allows the PowerCenter Integration Service to create temporary views in the database when it pushes the session to the database. The PowerCenter Integration Service must create a view in the database if the session contains an SQL override, a filtered lookup, or an unconnected lookup.
Allow Temporary Sequence for Pushdown	Allows the PowerCenter Integration Service to create temporary sequence objects in the database. The PowerCenter Integration Service must create a sequence object in the database if the session contains a Sequence Generator transformation.
Session Sort Order	Sort order for the session. The session properties display the options that you can select based on the client locale settings. You can select one of the following values for the sort order: - 0. BINARY - 2. SPANISH - 3. TRADITIONAL_SPANISH - 4. DANISH - 5. SWEDISH - 6. FINNISH When the PowerCenter Integration Service runs in Unicode mode, it sorts character data in the session using the selected sort order. When the PowerCenter Integration Service runs in ASCII mode, it ignores this setting and uses a binary sort order to sort character data.

Mapping Tab (Transformations View)

The Transformations view of the Mapping tab contains the following nodes:

- **Start Page.** Describes the nodes on the Mapping tab.
- **Pushdown Optimization.** Displays the Pushdown Optimization Viewer where you can view and configure pushdown groups.
- **Connections.** Displays the source, target, lookup, stored procedure, FTP, external loader, and queue connections. You can choose connection types and connection values. You can also edit connection object values.
- **Memory Properties.** Displays memory attributes that you configured on other tabs in the session properties. Configure memory attributes such as DTM buffer size, cache sizes, and default buffer block size.
- **Files, Directories, and Commands.** Displays file names and directories for the session. This includes session logs reject file, and target file names and directories.
- **Sources.** Displays the mapping sources and settings that you can configure in the session.
- **Targets.** Displays the mapping target and settings that you can configure in the session.
- **Transformations.** Displays the mapping transformations and settings that you can configure in the session.

Sources Node

The Sources node lists the mapping sources and displays the settings. If you want to view and configure the settings of a specific source, select the source from the list. You can configure the following settings:

- **Readers.** Displays the reader that the Integration Service uses with each source instance. The Workflow Manager specifies the necessary reader for each source instance.
- **Connections.** Displays the source connections. You can choose connection types and connection values. You can also edit connection object values.
- **Properties.** Displays source and source qualifier properties. For relational sources, you can override properties that you configured in the Mapping Designer.

  For file sources, you can override properties that you configured in the Source Analyzer. You can also configure the following session properties for file sources:

File Source Options	Description
Source File Directory	Enter the directory name in this field. By default, the Integration Service looks in the service process variable directory, $PMSourceFileDir, for file sources. If you specify both the directory and file name in the Source Filename field, clear this field. The Integration Service concatenates this field with the Source Filename field when it runs the session. You can also use the $InputFile*Name* session parameter to specify the file directory.
Source Filename	Enter the file name, or file name and path. Optionally use the $InputFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Source File Directory field when it runs the session. For example, if you have “C:\data\” in the Source File Directory field, then enter “filename.dat” in the Source Filename field. When the Integration Service begins the session, it looks for “C:\data\filename.dat”. By default, the Workflow Manager enters the file name configured in the source definition.
Source Filetype	You can configure multiple file sources using a file list. Indicates whether the source file contains the source data, or a list of files with the same file properties. Select Direct if the source file contains the source data. Select Indirect if the source file contains a list of files. When you select Indirect, the Integration Service finds the file list then reads each listed file when it executes the session.

When you configure a session to extract data from a PowerExchange nonrelational source in batch mode, you can configure the following session properties for the source:

Attribute Name	Description
Schema Name Override	Overrides the schema name in the source PowerExchange data map.
Map Name Override	Overrides the data map name of the source PowerExchange data map.

Attribute Name	Description
File Name	For the ADABAS Unload source type, specifies the file name of the unloaded Adabas database. Required for the ADABAS Unload source type.
ADABAS Password	For the ADABAS source type, the database password. If the ADABAS FDT for the source file is password-protected, enter the ADABAS FDT password. **Note:** PowerCenter encrypts the password and displays the encrypted password in the XML file that it generates for the workflow.
Database Id Override	For the ADABAS and ADABAS Unload source types, overrides the ADABAS Database ID in the PowerExchange data map.
File Id Override	For the ADABAS and ADABAS Unload source types, overrides the Adabas file ID in the PowerExchange data map.
DB2 Sub System Id	For the DB2 Datamaps source type, overrides the DB2 subsystem ID in the PowerExchange data map.
DB2 Table name	For the DB2 Datamaps source type, overrides the DB2 table name in the PowerExchange data map.
Unload File Name	For the DB2 Unload Datasets source type, overrides the DB2 unload file name in the PowerExchange data map.

Attribute Name	Description

Filter Overrides	Filters the source data that PowerExchange reads based on specific conditions that you define.
	PWXPC adds the filter conditions in a WHERE clause on a SELECT SQL statement and then passes the SQL statement to PowerExchange for processing. You can use any filter condition syntax that PowerExchange supports for NRDB SQL.
	For a single-record source, use the following syntax:
	`filter_condition`
	For example, the following filter condition selects records where a column called TYPE has a value of A or D:
	`TYPE=‘A’ or TYPE=‘D’`
	For a multiple-record source, use one of the following syntax alternatives:
	`filter_condition`
	`group_name1=filter;    group_name2=filter;...`
	The *group_name* syntax limits the SQL query condition to a specific record in a multi-record source definition. If you do not use the *group_name* syntax, the SQL query condition applies to all records in the multi-record source definition.
	For example, to select only records that contain an ID column value of "DBA" for a multi-record source that has USER1 and USER2 records, specify one of the following SQL query conditions:
	`USER1=ID=’DBA’;USER2=ID=’DBA’`
	`ID=’DBA’`
	Note: If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example:
	`SELECT * from schema.table WHERE Filter_Overrides_conditions AND SQL_Query_Override_conditions`

IMS Unload File Name	For the IMS source type, an IMS database unload file name. Required if you want to read source data from the backup file instead of from the IMS database. For a multiple-record write to an IMS unload file, required for both the source and target.
IMS AM Override	For the IMS source type, overrides the IMS access method in the imported data map for the source with the other available access method. The session then uses the override access method at run time. - If you imported a source data map that specifies the DL/1 BATCH access method, enter **O** to override it with the IMS ODBA access method. For ODBA access, you must also specify the **IMS PSBNAME Override** and **IMS PCBNAME Override** attributes. - If you imported a source data map that specifies the IMS ODBA access method, enter **D** to override it with the DL/1 BATCH access method, which provides DL/I or BMP access. You must also specify the **IMS PCBNUMBER Override** attribute. **Important:** Before you run the session with an access method override, ensure that you complete the PowerExchange configuration tasks for the new access method. For example, if the override is DL/1 BATCH, you must configure LISTENER and NETPORT statements in the DBMOVER member and configure the netport JCL. If the override is IMS ODBA, you must perform other configuration tasks. For more information, see "IMS Data Maps" in the *PowerExchange Navigator User Guide*.

Attribute Name	Description
IMS SSID Override	For the IMS source type, if you imported an IMS ODBA data map for the source and did not override the access method, use this attribute to override the IMS subsystem ID (SSID) from the data map for the session. If you specified ODBA access as an override in the **IMS AM Override** session attribute, you must enter this value. An SSID is required for ODBA access. If the session has an IMS unload file source, you can use this override to point to another IMSID statement in the DBMOVER member for the purpose of changing from one DBD library to another DBD library. By using the override, you can switch DBD libraries without editing or adding any IMSID statement and restarting the PowerExchange Listener. For example, use this override to test changes that you made to a DBD library against an unload file. If you use a netport job with BMP access to IMS, you can use this override with the %IMSID substitution variable in the netport JCL to specify an IMS SSID to use for the session. This override replaces the substitution variable. By using the override with the substitution variable, you can use the same netport JCL to access multiple IMS environments, such as development, test, and production environments. **Note:** An IMS SSID is not required for DL/I batch access to IMS data or for access to an IMS unload file.
IMS PSBNAME Override	For the IMS source type, if you imported an IMS ODBA data map for the source and did not override the access method, this value overrides the PSB name from the data map. If you specified ODBA access as an override in the **IMS AM Override** attribute, you must enter this value. A PSB name is required for ODBA access. If you use DL/I batch or BMP access and specify this override, you must also specify the PSB=%PSBNAME substitution variable in the netport JCL. The override value then replaces the substitution variable in the JCL. If you specify the PSB=%1 substitution variable instead of PSB=%PSBNAME in the netport JCL, the session uses the PSB name from the NETPORT statement, if specified. In this case, you need a separate NETPORT statement for each PSB. To avoid exceeding the limit of ten NETPORT statements in the DBMOVER member, use this override with %PSBNAME substitution variable instead. **Note:** A PSB name is not used for access to an IMS source unload file.
IMS PCBNAME Override	For the IMS source type, if you imported an IMS ODBA data map for the source and did not override the access method, this value overrides the PCB name from the data map. If you specified ODBA access as an override in the **IMS AM Override** attribute, you must enter this value. A PCB name is required for ODBA access. A PCB name is not used for DL/I batch or BMP access or for access to an IMS unload file.
IMS PCBNUMBER Override	For the IMS source type, if you imported a DL/1 BATCH data map for the source and did not override the access method, this value overrides the PCB number from the data map. If you specified DL/I access as an override in the **IMS AM Override** attribute, you must enter this value. A PCB number is required for DL/I or BMP access. A PCB number is not used for IMS ODBA access or for access to an IMS unload file.
File Name Override	For the VSAM Files and Sequential Files source types, overrides the data set or file name in the PowerExchange data map. Enter the complete data set or file name. For the i5/OS, the format is: *library_name*/*file_name*. If you select the **Filelist File** check box, enter the name of a filelist file in this attribute. A filelist file is a list of files.

Attribute Name	Description
Filelist File	For the VSAM Files and Sequential Files source types, identifies the file that contains a list of files. Select this attribute only if you entered a filelist file in the **File Name Override** field.

SQL Query Override	Overrides the SQL query sent to PowerExchange.
	PWXPC replaces the default SQL query with the SQL statement that you enter and passes the SQL statement to PowerExchange for processing. You can enter any SQL statement that PowerExchange supports for NRDB SQL.
	For example, you can select records from table USER where a column called TYPE has a value of A or D by specifying the following SQL query override:
	`Select ID, NAME from USER where TYPE=‘A’ or TYPE=‘D’;`
	For a multiple-record source, use the following syntax:
	`group_name1=sql_query_override1;`
	`group_name2=sql_query_override2;...`
	For example, you can select only records with ID column values that contain DBA for a multi-record source with two records called USER1 and USER2 by specifying the following SQL query override:
	`USER1=Select ID, NAME from USER1 where ID='DBA'; USER2=Select ID, NAME from USER2 where ID='DBA';`
	Note: If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example:
	`SELECT * from schema.table WHERE Filter_Overrides_conditions AND SQL_Query_Override_conditions`

Attribute Name	Description
PWX Partition Strategy	For offloaded DB2 Unload, VSAM Files, and Sequential Files source types, specifies one of the following partitioning strategies: - **Single Connection**. PowerExchange creates a single connection to the data source. Any overrides specified for the first partition are used for all partitions. With this option, if you specify any overrides for other partitions that differ from the overrides for the first partition, the session fails with an error message. - **Overrides Driven**. If the specified overrides are the same for all partitions, PowerExchange creates a single connection to the data source. If the overrides are not identical for all partitions, PowerExchange creates multiple connections.
Flush After N Blocks	For multiple-record sources, specifies the maximum number of block flushes that can occur without any one block being flushed. For bulk multiple-record sources, by default, PWXPC flushes blocks of data only when the buffers are completely full or at end-of-file. If some record types do not have as much data as others, flushing might not occur often. In this case, the record types might not have data on the target for a long time, thereby blocking flushes on the writer side. To ensure that buffers for all record types are flushed at a regular interval, define this **Flush After N Blocks** session property. This property specifies the maximum number of block flushes that can occur across all record types without any one block being flushed. A value of zero disables this feature and causes flushing to occur only when blocks are full. Valid values for the property are -1 to 100000. The default value of -1 works in the following manner: - For all multiple-record sources that do not use sequence fields, process the same as Flush After N Blocks = 0, which disables this feature and flushes only when blocks are full . - For all multiple-record sources that use sequence fields, use Flush After N Blocks = 7 * (number of record types in the source).

When you configure a session to extract data from a PowerExchange relational source in batch mode, you can configure the following session properties for the source:

Attribute Name	Description
Schema Name Override	Overrides the source schema name.
Map Name Override	Overrides the source table name.
DB2 Sub System Id	Overrides the DB2 instance name in the PowerExchange data map.
Image Copy Dataset	For DB2 image copy sources, provides the image copy data set name. If not specified and the table is in a non-partitioned table space, the most current image copy data set with TYPE=FULL and SHRLEVEL=REFERENECE is used. If the table is in a partitioned table space, you must specify the **Image Copy Dataset** attribute.

Attribute Name	Description
Disable Consistency Checking	If cleared for a DB2 image copy source, PowerExchange reads the catalog to verify that the DSN of the specified image copy data set is defined with SHRLEVEL=REFERENCE and TYPE=FULL and is an image copy of the specified table. If the DSN is not defined with these properties, the session fails. If selected, PowerExchange reads the Image Copy Dataset regardless of the values of SHRLEVEL and TYPE and without verifying that the object ID in the image copy matches the object ID in the DB2 catalog.
Filter Overrides	Filters the source data that PowerExchange reads based on specified conditions. PWXPC adds filter conditions specified to the WHERE clause on the SELECT SQL statement and passes the SQL statement to PowerExchange for processing. You can use any filter condition syntax that PowerExchange supports for NRDB SQL. For more information, see the *PowerExchange Reference Manual*. For example, you can select records where a column called TYPE has a value of A or D by specifying the following filter condition: `TYPE='A' or TYPE='D'` **Note:** If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example: `SELECT * from` *`schema.table`* `WHERE` *`Filter_Overrides_conditions`* `AND` *`SQL_Query_Override_conditions`*
SQL Query Override	Overrides the SQL query sent to PowerExchange. **Caution:** For DB2 for z/OS data sources, PowerExchange automatically appends FOR FETCH ONLY to SQL SELECT statements. If you include FOR FETCH ONLY in the **Sql Query Override** attribute in the **Properties** area, the expression is included twice in the SELECT statement. In this case, PowerExchange issues an error. **Note:** If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example: `SELECT * from` *`schema.table`* `WHERE` *`Filter_Overrides_conditions`* `AND` *`SQL_Query_Override_conditions`*

When you create a source definition for a CDC source by using an extraction map and then configure a session to extract data from the source, you can configure the following session properties for the source:

Attribute Name	Description
Schema Name Override	Overrides the schema name in the PowerExchange extraction map.
Map Name Override	Overrides the PowerExchange extraction map name.

Attribute Name	Description
ADABAS Password	For the Adabas source type, an Adabas password for the source file. If the Adabas FDT for the source file is password-protected, enter the Adabas FDT password. **Note:** PowerCenter encrypts the password and displays the encrypted password in the XML file that it generates for the workflow.
Database Id Override	For the Adabas source type, overrides the Adabas database ID in the PowerExchange data map.
File Id Override	For the Adabas source type, overrides the Adabas file ID in the PowerExchange data map.
Library/File Override	For the DB2i5OS Real Time source type, overrides the library and file names in the extraction map. Specify the full library name and file name in the format: `library/file` Alternatively, specify an asterisk (*) wildcard for the library name to retrieve changes for all files of the same file name across multiple libraries. This attribute overrides the **Library/File Override** attribute on the application connection.
Source Schema Override	For the Oracle source type, overrides the source schema name.

Attribute Name	Description

Filter Overrides	Filters the source data that PowerExchange reads based on specified conditions.
	PWXPC adds filter conditions specified to the WHERE clause on the SELECT SQL statement and passes the SQL statement to PowerExchange for processing. You can use any filter condition syntax that PowerExchange supports for NRDB SQL. For more information, see the *PowerExchange Reference Manual*.
	For example, you can select records where a column called TYPE has a value of A or D by specifying the following filter condition:
	`TYPE=‘A’ or TYPE=‘D’`
	To select change records where columns ID and ACCOUNT have changed, you can use the DTL__CI columns by specifying the following filter condition:
	`DTL__CI_ID=‘Y’ and DTL__CI_ACCOUNT=’Y’`
	Note: If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example:
	`SELECT * from schema.table WHERE Filter_Overrides_conditions AND SQL_Query_Override_conditions`

SQL Query Override	Overrides the SQL query sent to PowerExchange. **Note:** If you specify both the Filter Overrides attribute and a SQL Query Override attribute that contains a filtering WHERE clause, the resulting SELECT statement contains a WHERE clause that uses the AND operator to associate the Filter Overrides filter conditions with the SQL Query Override conditions. For example: `SELECT * from schema.table WHERE Filter_Overrides_conditions AND SQL_Query_Override_conditions`

When you create a source definition for a CDC source by importing metadata from a relational database and then configure a session to extract data from the source, you can configure the following session properties for the source:

Attribute Name	Description
Extraction Map Name	Required. The PowerExchange extraction map name for the CDC source. You must specify the extraction map name for the relational source.
Library/File Override	Optional. For the DB2i5OS Real Time source type, overrides the library and file names in the extraction map. Specify the full library name and file name in the format: *library/file* Alternatively, specify an asterisk (*) wildcard for the library name to retrieve changes for all files of the same file name across multiple libraries. This attribute overrides the **Library/File Override** value on the application connection.
Source Schema Override	Optional. For the Oracle Change and Real Time source types, overrides the source schema name.

Targets Node

The Targets node lists the mapping targets and displays the settings. To view and configure the settings of a specific target, select the target from the list. You can configure the following settings:

- **Writers.** Displays the writer that the Integration Service uses with each target instance. For relational targets, you can choose a relational writer or a file writer. Choose a file writer to use an external loader. After you override a relational target to use a file writer, define the file properties for the target. Click Set File Properties and choose the target to define.
- **Connections.** Displays the target connections. You can choose connection types and connection values. You can also edit connection object values.
- **Properties.** Displays different properties for different target types. For relational targets, you can override properties that you configured in the Mapping Designer. You can also configure the following session properties for relational targets:

Relational Target Property	Description
Target Load Type	You can choose Normal or Bulk. If you select Normal, the Integration Service loads targets normally. You can choose Bulk when you load to DB2, Sybase, Oracle, or Microsoft SQL Server. If you specify Bulk for other database types, the Integration Service reverts to a normal load. Loading in bulk mode can improve session performance, but limits the ability to recover because no database logging occurs. Choose Normal mode if the mapping contains an Update Strategy transformation. If you choose Normal and the Microsoft SQL Server target name includes spaces, configure the following connection environment SQL in the connection object: `SET QUOTED_IDENTIFIER ON`
Insert	The Integration Service inserts all rows flagged for insert.
Update (as Update)	The Integration Service updates all rows flagged for update.
Update (as Insert)	The Integration Service inserts all rows flagged for update.
Update (else Insert)	The Integration Service updates rows flagged for update if they exist in the target, and inserts remaining rows marked for insert.
Delete	The Integration Service deletes all rows flagged for delete.
Truncate Table	The Integration Service truncates the target before loading.

Relational Target Property	Description
Reject File Directory	Reject-file directory name. By default, the Integration Service writes all reject files to the service process variable directory, $PMBadFileDir. If you specify both the directory and file name in the Reject Filename field, clear this field. The Integration Service concatenates this field with the Reject Filename field when it runs the session. You can also use the $BadFile*Name* session parameter to specify the file directory.
Reject Filename	File name or file name and path for the reject file. By default, the Integration Service names the reject file after the target instance name: *target_name*.bad. Optionally, use the $BadFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Reject File Directory field when it runs the session. For example, if you have "C:\reject_file\" in the Reject File Directory field, and enter "filename.bad" in the Reject Filename field, the Integration Service writes rejected rows to C:\reject_file\filename.bad.

For file targets, you can override properties that you configured in the Target Designer. You can also configure the following session properties for file targets:

File Target Property	Description
Merge Partitioned Files	When selected, the Integration Service merges the partitioned target files into one file when the session completes, and then deletes the individual output files. If the Integration Service fails to create the merged file, it does not delete the individual output files. You cannot merge files if the session uses FTP, an external loader, or a message queue.
Merge File Directory	Enter the directory name in this field. By default, the Integration Service writes the merged file in the service process variable directory, $PMTargetFileDir. If you enter a full directory and file name in the Merge File Name field, clear this field.
Merge File Name	Name of the merge file. Default is *target_name*.out. This property is required if you select Merge Partitioned Files.
Create Directory if Not Exists	Creates the target directory if it does not exist.
Output File Directory	Enter the directory name in this field. By default, the Integration Service writes output files in the service process variable directory, $PMTargetFileDir. If you specify both the directory and file name in the Output Filename field, clear this field. The Integration Service concatenates this field with the Output Filename field when it runs the session. You can also use the $OutputFile*Name* session parameter to specify the file directory.

File Target Property	Description
Output Filename	Enter the file name, or file name and path. By default, the Workflow Manager names the target file based on the target definition used in the mapping: *target_name*.out. If the target definition contains a slash character, the Workflow Manager replaces the slash character with an underscore. When you use an external loader to load to an Oracle database, you must specify a file extension. If you do not specify a file extension, the Oracle loader cannot find the flat file and the Integration Service fails the session. Enter the file name, or file name and path. Optionally use the $OutputFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Output File Directory field when it runs the session. **Note:** If you specify an absolute path file name when using FTP, the Integration Service ignores the Default Remote Directory specified in the FTP connection. When you specify an absolute path file name, do not use single or double quotes.
Reject File Directory	Enter the directory name in this field. By default, the Integration Service writes all reject files to the service process variable directory, $PMBadFileDir. If you specify both the directory and file name in the Reject Filename field, clear this field. The Integration Service concatenates this field with the Reject Filename field when it runs the session. You can also use the $BadFile*Name* session parameter to specify the file directory.
Reject Filename	Enter the file name, or file name and path. By default, the Integration Service names the reject file after the target instance name: *target_name*.bad. Optionally use the $BadFile*Name* session parameter for the file name. The Integration Service concatenates this field with the Reject File Directory field when it runs the session. For example, if you have "C:\reject_file\" in the Reject File Directory field, and enter "filename.bad" in the Reject Filename field, the Integration Service writes rejected rows to C:\reject_file\filename.bad.

You can configure the following session properties for PowerExchange nonrelational targets:

Attribute Name	Description
ADABAS Password	For the ADABAS target type, the Adabas file password. If the ADABAS FDT for the target file is password protected, enter the ADABAS FDT password. **Note:** PowerCenter encrypts the password and displays the encrypted password in the XML file that it generates for the workflow.
BLKSIZE	For the SEQ target type on z/OS, the z/OS data set block size. Default is 0, which means use the best possible block size. If you select **VB** for the **RECFM** value, the actual block size might be up to four bytes greater than the value you specify for **BLKSIZE**.
DATACLAS	For the SEQ target type on z/OS, the z/OS SMS data class name.
Delete SQL Override	For the ADABAS and VSAM target types, overrides the default Delete SQL that is sent to PowerExchange.

Attribute Name	Description
Disp	For the SEQ target type on z/OS, the z/OS data set disposition. Valid values: - OLD - SHR - NEW - MOD Default is MOD if the data set exists, and NEW if it does not.
File Name Override	For the SEQ and VSAM target types, overrides the data set or file name in the PowerExchange data map. Enter the complete data set or file name. For i5/OS, use the following format: *library_name/file_name*.
IMS AM Override	For the IMS target type, overrides the IMS access method in the imported data map for the target with the other allowable access method. The session then uses the override access method at run time. - If you imported a target data map that specifies the DL/1 BATCH access method, enter **O** to override it with the IMS ODBA access method. For ODBA access, you must also specify the **IMS PSBNAME Override** and **IMS PCBNAME Override** attributes. - If you imported a target data map that specifies the IMS ODBA access method, enter **D** to override it with the DL/1 BATCH access method, which provides DL/I or BMP access. You must also specify the **IMS PCBNUMBER Override** attribute. **Important:** Before you run the session with an access method override, ensure that you complete the PowerExchange configuration tasks for the new access method. For example, if the override is DL/1 BATCH, you must configure LISTENER and NETPORT statements in the DBMOVER member and configure the netport JCL. If the override is IMS ODBA, you must perform other configuration tasks. For more information, see "IMS Data Maps" in the *PowerExchange Navigator User Guide*.
IMS PCBNAME Override	For the IMS target type, if you imported an IMS ODBA data map for the target and did not override the access method, this value overrides the PCB name from the data map. If you specified ODBA access as an override in the **IMS AM Override** attribute, you must enter this value. A PCB name is required for ODBA access. A PCB name is not used for DL/I or BMP access.
IMS PCBNUMBER Override	For the IMS target type, if you imported a DL/1 BATCH data map for the target and did not override the access method, this value overrides the PCB number from the data map. If you specified DL/I or BMP access as an override in the **IMS AM Override** attribute, you must enter this value. A PCB number is required for DL/I or BMP access. A PCB number is not used for IMS ODBA access.
IMS PSBNAME Override	If you imported an IMS ODBA data map for the target and did not override the access method, this value overrides the PSB name from the data map. If you specified ODBA access as an override in the **IMS AM Override** attribute, you must enter this value. A PSB name is required for ODBA access. If you use DL/I batch or BMP access and specify this override, you must also specify the PSB=%PSBNAME substitution variable in the netport JCL. The override value then replaces the substitution variable in the JCL. If you specify the PSB=%1 substitution variable instead of PSB=%PSBNAME in the netport JCL, the session uses the PSB name in the NETPORT statement, if specified. In this case, you need a separate NETPORT statement for each PSB. To avoid exceeding the limit of ten NETPORT statements, use this override with %PSBNAME substitution variable instead.

Attribute Name	Description
IMS SSID Override	For the IMS target type, if you imported an IMS ODBA data map for the target and did not override the access method, use this value to override the IMS subsystem ID (SSID). If you specified ODBA access as an override in the **IMS AM Override** attribute, you must enter this value. An SSID is required for ODBA access. If you use the IMS DL/1 BATCH access method and a BMP netport job, you can use this override with the %IMSID substitution variable in the netport JCL. This override replaces the substitution variable to specify the IMS SSID to use for the session. By using the substitution variable and override together, you can use the same netport JCL to access multiple IMS environments, such as development, testing, and production environments. **Note:** An IMS SSID is not required for DL/I batch access to IMS data or for access to an IMS unload file.
Initialize Target	For the VSAM target type, select this option to have PowerExchange allow both inserts and updates into empty VSAM data sets. If this option is not selected, PowerExchange only allows inserts into empty VSAM data sets.
Insert Only	For the ADABAS and VSAM target types, processes updates and deletes as inserts. **Note:** You must select this option when the target has no keys.
Insert SQL Override	For all nonrelational target types, overrides the default Insert SQL sent to PowerExchange.
LRECL	For the SEQ target type on z/OS, the data set logical record length. This value is ignored if Disp is not MOD or NEW. Default is 256. If you select **VB** for the **RECFM** value, specify the maximum number of data bytes in a logical record for LRECL. PowerExchange adds 4 to this value for the record descriptor word (RDW).
Map Name Override	For all nonrelational target types, overrides the target PowerExchange data map name. **Note:** PWXPC sends the file name that is specified for the source in the mapping unless this name is overridden in the **File Name Override** attribute.
MGMTCLAS	For the SEQ target type on z/OS, the SMS management class name. This value is ignored if Disp is not MOD or NEW.
MODELDCB	for the SEQ target type on z/OS, the Model DCB for non-SMS-managed GDG data sets. This value is ignored if Disp is not MOD or NEW.
Post SQL	For all nonrelational target types, one or more SQL statements that are executed after the session runs with the target database connection.
Pre SQL	For all nonrelational target types, one or more SQL statements that are executed before the session runs with the target database connection. **Note:** In certain cases, you must specify the **Pre SQL run once per Connection** attribute along with the **Pre SQL** attribute.

Attribute Name	Description
Pre SQL run once per Connection	For all nonrelational target types, runs the SQL that you specify in the **Pre SQL** attribute only once for a connection. Select this attribute in either of the following cases: - In the **Pre SQL** attribute for a session that uses writer partitioning, you specify a SQL statement such as CREATEFILE that can run only once for the session. If you do not select **Pre SQL run once per Connection**, the session tries to run the statement once for each partition. - In the **Pre SQL** attribute for a session that performs a multiple-record write, you specify a CREATEFILE statement that creates a new generation of a GDG or creates an empty file. If you do not select **Pre SQL run once per Connection**, the session creates a generation or tries to create a new empty file for each record that the session writes.
Primary Space	For the SEQ target type on z/OS, the primary space allocation, in the units specified in the **Space** attribute. This value is ignored if Disp is not MOD or NEW. Default is 1.
RECFM	For the SEQ target type on z/OS, the z/OS record format. Valid values are F, V, FU, FB, VU, VB, FBA, and VBA. This value is ignored if DISP is not MOD or NEW.
Schema Name Override	For all nonrelational target types, overrides the schema name in the target PowerExchange data map. **Note:** PWXPC sends the file name for the source in the mapping unless this name is overridden in **File Name Override** attribute.
Secondary Space	For the SEQ target type on z/OS, the secondary space allocation, in the units specified in the **Space** attribute. This value is ignored if Disp is not MOD or NEW. Default is 1.
Space	For the SEQ target type on z/OS, the type of units for expressing primary or secondary space for z/OS data sets. Valid values are: - CYLINDER - TRACK This value is ignored if Disp is not MOD or NEW. Default is TRACK.
STORCLAS	For the SEQ target type on z/OS, the SMS storage class name. This value is ignored if Disp is not MOD or NEW.
Truncate target option	For the VSAM target type, truncates, or deletes, table contents before loading new data. **Note:** VSAM data sets must be defined with the REUSE option for this truncate option to function correctly.
UNIT	For the SEQ target type on z/OS, the z/OS unit type. This value is ignored if Disp is not MOD or NEW. Default is SYSDA.
Update SQL Override	For the ADABAS and VSAM target type, overrides the default Update SQL that is sent to PowerExchange.

Attribute Name	Description
Upsert	For the ADABAS and VSAM target type, processes failed inserts as updates and updates as inserts.
VOLSER	For the SEQ target type on z/OS, the volume serial number. This value is ignored if Disp is not MOD or NEW.

Transformations Node

On the Transformations node, you can override transformation properties that you configure in the Designer. The attributes you can configure depends on the type of transformation you select.

Mapping Tab (Partitions View)

In the Partitions view of the Mapping tab, you can configure partitions. You can configure partitions for non-reusable sessions in the Workflow Designer and for reusable sessions in the Task Developer.

The following nodes are available in the Partitions view:

- **Partition Properties.** Configure partitions with the Partition Properties node.
- **KeyRange.** Configure the partition range for key-range partitioning. Select Edit Keys to edit the partition key.
- **HashKeys.** Configure hash key partitioning. Select Edit Keys to edit the partition key.
- **Partition Points.** Select a partition point to configure attributes. You can add and delete partitions and partition points, configure the partition type, and add keys and key ranges.
- **Non-Partition Points.** The Non-Partition Points node displays mapping objects as icons. The Partition Points node lists the non-partition points in the tree. You can select a non-partition point and add partitions.

Components Tab

In the Components tab, you can configure pre-session shell commands, post-session commands, email messages if the session succeeds or fails, and variable assignments.

The following table describes the Components tab options:

Components Tab Option	Description
Task	Configure pre- or post-session shell commands, success or failure email messages, and variable assignments.
Type	Select None if you do not want to configure commands and emails in the Components tab. For pre- and post-session commands, select Reusable to call an existing reusable Command task as the pre- or post-session shell command. Select Non-Reusable to create pre- or post-session shell commands for this session task. For success or failure emails, select Reusable to call an existing Email task as the success or failure email. Select Non-Reusable to create email messages for this session task.
Value	Use to configure commands, emails, or variable assignments.

The following table describes the tasks available in the Components tab:

Components Tab Tasks	Description
Pre-Session Command	Shell commands that the Integration Service performs at the beginning of a session.
Post-Session Success Command	Shell commands that the Integration Service performs after the session completes successfully.
Post-Session Failure Command	Shell commands that the Integration Service performs if the session fails.
On Success Email	Integration Service sends On Success email message if the session completes successfully.
On Failure Email	Integration Service sends On Failure email message if the session fails.
Pre-session variable assignment	Assign values to mapping parameters, mapping variables, and session parameters before a session runs. Read-only for reusable sessions.
Post-session on success variable assignment	Assign values to parent workflow and worklet variables after a session completes successfully. Read-only for reusable sessions.
Post-session on failure variable assignment	Assign values to parent workflow and worklet variables after a session fails. Read-only for reusable sessions.

Metadata Extensions Tab

The following table describes the configuration options for the Metadata Extensions tab:

Metadata Extensions Tab Options	Description
Extension Name	Name of the metadata extension. Metadata extension names must be unique in a domain.
Datatype	Datatype: numeric (integer), string, boolean, or XML.
Value	Value of the metadata extension. For a numeric metadata extension, the value must be an integer. For a boolean metadata extension, choose true or false. For a string or XML metadata extension, click the button in the Value field to enter a value of more than one line. The Workflow Manager does not validate XML syntax.
Precision	Maximum length for string or XML metadata extensions.
Reusable	Select to make the metadata extension apply to all objects of this type (reusable). Clear to make the metadata extension apply to this object only (non-reusable).
Description	Description of the metadata extension.

Workflow Properties Reference

General Tab

You can change the workflow name and enter a comment for the workflow on the General tab. By default, the General tab appears when you open the workflow properties.

The following table describes the settings on the General tab:

General Tab Options	Description
Name	Name of the workflow.
Comments	Comment that describes the workflow.
Integration Service	Integration Service that runs the workflow by default. You can also assign an Integration Service when you run the workflow.
Suspension Email	Email message that the Integration Service sends when a task fails and the Integration Service suspends the workflow.
Disabled	Disables the workflow from the schedule. The Integration Service stops running the workflow until you clear the Disabled option.
Suspend on Error	The Integration Service suspends the workflow when a task in the workflow fails.
Web Services	Creates a service workflow. Click Config Service to configure service information.

General Tab Options	Description
Configure Concurrent Execution	Enables the Integration Service to run more than one instance of the workflow at a time. You can run multiple instances of the same workflow name, or you can configure a different name and parameter file for each instance. Click Configure Concurrent Execution to configure instance names.
Service Level	Determines the order in which the Load Balancer dispatches tasks from the dispatch queue when multiple tasks are waiting to be dispatched. Default is “Default.” You create service levels in the Administrator tool.

Properties Tab

Configure parameter file name and workflow log options on the Properties tab.

The following table describes the settings on the Properties tab:

Properties Tab Options	Description
Parameter File Name	Designates the name and directory for the parameter file. Use the parameter file to define workflow variables.
Write Backward Compatible Workflow Log File	Select to write workflow log to a file.
Workflow Log File Name	Enter a file name, or a file name and directory. Required. The Integration Service appends information in this field to that entered in the Workflow Log File Directory field. For example, if you have “C:\workflow_logs\” in the Workflow Log File Directory field, then enter “logname.txt” in the Workflow Log File Name field, the Integration Service writes logname.txt to the C:\workflow_logs\ directory.
Workflow Log File Directory	Designates a location for the workflow log file. By default, the Integration Service writes the log file in the service variable directory, $PMWorkflowLogDir. If you enter a full directory and file name in the Workflow Log File Name field, clear this field.
Save Workflow Log By	If you select Save Workflow Log by Timestamp, the Integration Service saves all workflow logs, appending a timestamp to each log. If you select Save Workflow Log by Runs, the Integration Service saves a designated number of workflow logs. Configure the number of workflow logs in the Save Workflow Log for These Runs option. You can also use the $PMWorkflowLogCount service variable to save the configured number of workflow logs for the Integration Service.
Save Workflow Log For These Runs	Number of historical workflow logs you want the Integration Service to save. The Integration Service saves the number of historical logs you specify, plus the most recent workflow log. Therefore, if you specify 5 runs, the Integration Service saves the most recent workflow log, plus historical logs 0–4, for a total of 6 logs. You can specify up to 2,147,483,647 historical logs. If you specify 0 logs, the Integration Service saves only the most recent workflow log.
Enable HA Recovery	Enable workflow recovery. Not available for web service workflows.

Properties Tab Options	Description
Automatically recover terminated tasks	Recover terminated tasks without user intervention. You must have high availability and the workflow must still be running. Not available for web service workflows.
Maximum automatic recovery attempts	When you automatically recover terminated tasks you can choose the number of times the Integration Service attempts to recover the task. Default is 5.

Scheduler Tab

The Scheduler Tab lets you schedule a workflow to run continuously, run at a given interval, or manually start a workflow.

You can configure the following types of scheduler settings:

- **Non-Reusable.** Create a non-reusable scheduler for the workflow.
- **Reusable.** Choose a reusable scheduler for the workflow.

The following table describes the settings on the Scheduler Tab:

Scheduler Tab Options	Description
Non-Reusable/Reusable	Indicates the scheduler type. If you select Non Reusable, the scheduler can only be used by the current workflow. If you select Reusable, choose a reusable scheduler. You can create reusable schedulers by selecting Schedulers.
Scheduler	Select a set of scheduler settings for the workflow.
Description	Enter a description for the scheduler.
Summary	Read-only summary of the selected scheduler settings.

Edit Scheduler Settings

Click the Edit Scheduler Settings button to configure the scheduler. The Edit Scheduler dialog box appears.

The following table describes the settings on the Edit Scheduler dialog box:

Scheduler Options	Description
Run Options: Run On Integration Service Initialization/Run On Demand/Run Continuously	Indicates the workflow schedule type. If you select Run On Integration Service Initialization, the Integration Service runs the workflow as soon as the Integration Service is initialized. If you select Run On Demand, the Integration Service only runs the workflow when you start the workflow. If you select Run Continuously, the Integration Service starts the next run of the workflow as soon as it finishes the first run.
Schedule Options: Run Once/Run Every/Customized Repeat	Required if you select Run On Integration Service Initialization in Run Options. Also required if you do not choose any setting in Run Options. If you select Run Once, the Integration Service runs the workflow once, as scheduled in the scheduler. If you select Run Every, the Integration Service runs the workflow at regular intervals, as configured. If you select Customized Repeat, the Integration Service runs the workflow on the dates and times specified in the Repeat dialog box.
Edit	Required if you select Customized Repeat in Schedule Options. Opens the Repeat dialog box, allowing you to schedule specific dates and times for the workflow to run. The selected scheduler appears at the bottom of the page.
Start Date	Required if you select Run On Integration Service Initialization in Run Options. Also required if you do not choose any setting in Run Options. Indicates the date on which the Integration Service begins scheduling the workflow.
Start Time	Required if you select Run On Integration Service Initialization in Run Options. Also required if you do not choose any setting in Run Options. Indicates the time at which the Integration Service begins scheduling the workflow.
End Options: End On/End After/ Forever	Required if the workflow schedule is Run Every or Customized Repeat. If you select End On, the Integration Service stops scheduling the workflow in the selected date. If you select End After, the Integration Service stops scheduling the workflow after the set number of workflow runs. If you select Forever, the Integration Service schedules the workflow as long as the workflow does not fail.

Customizing Repeat Option

You can schedule the workflow to run once, run at an interval, or customize the repeat option. Click the Edit button on the Edit Scheduler dialog box to configure Customized Repeat options.

The following table describes options in the Customized Repeat dialog box:

Repeat Option	Description
Repeat Every	Enter the numeric interval you want to schedule the workflow, then select Days, Weeks, or Months, as appropriate. If you select Days, select the appropriate Daily Frequency settings. If you select Weeks, select the appropriate Weekly and Daily Frequency settings. If you select Months, select the appropriate Monthly and Daily Frequency settings.
Weekly	Required to enter a weekly schedule. Select the day or days of the week on which you want to schedule the workflow.
Monthly	Required to enter a monthly schedule. If you select Run On Day, select the dates on which you want the workflow scheduled on a monthly basis. The Integration Service schedules the workflow on the selected dates. If you select a numeric date exceeding the number of days within a particular month, the Integration Service schedules the workflow for the last day of the month, including leap years. For example, if you schedule the workflow to run on the 31st of every month, the Integration Service schedules the session on the 30th of the following months: April, June, September, and November. If you select Run On The, select the week or weeks of the month, then the day of the week on which you want the workflow to run. For example, if you select Second and Last, then select Wednesday, the Integration Service schedules the workflow on the second and last Wednesday of every month.
Daily	Enter the number of times you would like the Integration Service to run the workflow on any day the session is scheduled. If you select Run Once, the Integration Service schedules the workflow once on the selected day, at the time entered on the Start Time setting on the Time tab. If you select Run Every, enter Hours and Minutes to define the interval at which the Integration Service runs the workflow. The Integration Service then schedules the workflow at regular intervals on the selected day. The Integration Service uses the Start Time setting for the first scheduled workflow of the day. If you choose an interval that is bigger than the start time, the workflow runs one time each day. The Integration Service then schedules the workflow at regular intervals on the selected day.

Variables Tab

Before using workflow variables, you must declare them on the Variables tab.

The following table describes the settings on the Variables tab:

Variable Options	Description
Name	Name of the workflow variable.
Datatype	Datatype of the workflow variable.
Persistent	Indicates whether the Integration Service maintains the value of the variable from the previous workflow run.
Is Null	Indicates whether the workflow variable is null.

Variable Options	Description
Default	Default value of the workflow variable.
Description	Optional details about the workflow variable.

Events Tab

Before using the Event-Raise task, declare a user-defined event on the Events tab.

The following table describes the settings on the Events tab:

Events Tab Options	Description
Events	Name of the event you declare.
Description	Details to describe the event.