Deployment Manager 4.1 User’s Guide

SPSS Inc.233 South Wacker Drive, 11th Floor
Chicago, IL 60606-6412
Tel: (312) 651-3000
Fax: (312) 651-3668

Printed in the United States of America.

Preface

Deployment Manager is a client application of PASW Collaboration and Deployment Services. This manual documents the tasks associated with everyday use of the facilities of PASW Collaboration and Deployment Services through Deployment Manager, such as managing repository content, scheduling jobs, and updating models or scores.

Technical Support

The services of SPSS Inc. Technical Support are available to registered customers of SPSS Inc.. Customers may contact Technical Support for assistance in using SPSS Inc. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the SPSS Inc. Web site at *http://www.spss.com*, or contact your local office, listed on the SPSS Inc. Web site at *http://www.spss.com/worldwide*. Be prepared to identify yourself, your organization, and the serial number of your system.

Tell Us Your Thoughts

Your comments are important. Please let us know about your experiences with SPSS Inc. products. Please send e-mail to *suggest@us.ibm.com*, or write to SPSS Inc., Attn: Director of Product Planning, 233 South Wacker Drive, 11th Floor, Chicago IL 60606-6412.

Contents

1 Overview 1

PASW Collaboration and Deployment Services 1

Collaboration 1

Deployment 2

System Architecture 2

Repository 3

Deployment Manager 4

Deployment Portal 4

Browser-based Deployment Manager 5

Enterprise View 5

Execution Servers 5

PASW BIRT Report Designer 6

Products with Collaboration 6

2 What's New? 8

New in Release 4.1 8

Deprecated Features 9

3 Getting Started 10

Launching Deployment Manager 10

Navigating Through the System 10

Using the Mouse vs. Pressing Enter 11

Selecting Items from the File Menu vs. Using the Right-click Method 11

Dragging and Dropping Items in the User Interface 11

Accessing Help 12

Entry Field Content Assist 12

Field Conventions 13

Naming Conventions Within the System 14

Exiting the System 14

4 The Content Explorer 15

Content Explorer Overview . . . 15
- What Is a Content Object? . . . 15
- Organization of the Content Explorer . . . 15

Working with Servers . . . 16
- Creating New Content Server Connections . . . 16
- Logging in to a Server . . . 17
- Logging off from a Server. . . . 18
- Changing Server Passwords . . . 19

Working with Files . . . 20
- Opening External Files . . . 20
- Permissions Inheritance . . . 20

Working with the Repository. . . . 20
- Adding Files to the Repository . . . 20
- Downloading Files from the Repository . . . 21
- Deleting Files from the Repository . . . 21

Searching. . . . 22
- Accessing the Search Dialogs . . . 22
- Simple Search . . . 23
- Advanced Search . . . 23
- Viewing Search Results. . . . 29
- Terms Not Included in a Search. . . . 30

Object Locking . . . 31
- Locking Objects. . . . 31
- Viewing the Locked Objects Table . . . 32
- Unlocking Objects . . . 33
- Exporting and Importing Locked Objects . . . 33

Obtaining Updates . . . 33

5 Properties 34

Working With Object Properties . . . 34
- Viewing Object Properties . . . 34
- Editing Object Properties . . . 35
- Working with Expiration Dates and Expired Files . . . 50

Working With Server Properties and User Preferences. . . . 54
- Server Properties . . . 55
- User Preferences . . . 58

Version Properties . . . 61

Working with Custom Properties. 64
Verifying Access Privileges to Create Custom Properties 65
Accessing the Custom Properties Dialog Box. 65
Creating Custom Properties 66
Editing Custom Properties 69
Searching for Custom Properties 71
Deleting Custom Properties 71
Working with Topics 72
Process Overview 73
Verifying Access Privileges to Create Topic Definitions 73
Working with Topic Definitions 73
Accessing the Topic Definitions Dialog Box 74
Creating New Topic Definitions 74
Renaming Topic Definitions 75
Moving Topic Definitions 75
Deleting Topic Definitions 76
Searching for Topics. 77
Bulk Property Updates 77
Updating General Properties in Bulk 78
Updating Version Properties in Bulk. 80
Updating Permissions in Bulk. 81

6 Resource Definitions 82

Adding New Credentials. 82
Credential Destination 83
Specifying a User and Password for Credentials 84
Creating a New Domain. 84
Working with Data Source Definitions. 85
Selecting a Data Source Definition Type 85
Specifying a DSN for an ODBC Data Source. 86
Specifying a JDBC Name and URL. 87
Specifying a JNDI Name for an Application Server Data Source 91
Specifying Properties for Data Service Data Sources. 91
Modifying Data Source Definitions. 93
Message Domains 94
Creating a New Message Domain 94
Modifying Message Domain Definitions. 96
Server Definitions. 96
Adding New Server Definitions 97
Modifying Server Definitions 102

Server Clusters 103
Creating a New Server Cluster 103
Modifying a Server Cluster 106
Importing Resource Definitions 106

7 Exporting and Importing Folders 108

Export and Import Overview 108
Job Components That Are Migrated During Export and Import 108
Exporting External References 108
Export and Import Restrictions 109
Recommended Import Order 109
Security Permissions When Importing Folders 109
Exporting Folders 110
Importing Folders 111
Resolving Import Conflicts 114

8 Enterprise View 123

Enterprise View Overview 123
Enterprise View Builder Interface 123
Credential Definitions 124
Data Source Definitions 124
Enterprise View 124
Enterprise View Editor 125
Application View 146
Creating an Application View 147
Application View Editor 147
Merging View Versions 153
Resolve Conflicts 154
Data Provider Definition 155
Creating a Data Provider Definition 155
Data Provider Definition Editor 158
Data Provider Definition - Real Time 162
Creating a Data Provider Definition - Real Time 163
Data Provider Definition - Real Time Editor 165
Expression Editor 172
Expression Panel 173

Functions Panel . . . 174
Importing and Exporting Enterprise View Objects . . . 206
Enterprise View Data Types . . . 207

9 Scoring 212

Supported Scoring Functions and Models. . . . 212
Scoring Configurations . . . 214
New Scoring Model Configuration . . . 214
Model Specific Settings. . . . 215
Data Provider Settings . . . 216
Input Data Order . . . 217
Input Data Returned Settings . . . 218
Output Data Returned Settings. . . . 219
Logging Settings . . . 220
Advanced Settings . . . 223
Scoring View . . . 224
Filtering the Scoring View . . . 224
Editing a Scoring Configuration . . . 225
Suspending and Resuming Scoring Configurations . . . 225
Deleting a Scoring Configuration . . . 226
Scoring Graph View . . . 226

10 Jobs 228

What Is a Job? . . . 228
Versioning and Labeling Jobs . . . 228
Components of a Job . . . 229
Prerequisite to Running Jobs . . . 229
External File Dependencies . . . 229
Job Process Overview . . . 230
Working with Jobs in the Content Explorer . . . 230
Creating New Jobs . . . 230
Opening an Existing Job. . . . 232
Viewing Job Properties . . . 232
Working with the Job Editor . . . 232
General Information Tab . . . 234
Job Variables. . . . 234
Adding Steps to a Job. . . . 236
Executing Job Steps Concurrently . . . 237

Specifying Relationships in a Job 237
Sequential Connector 238
Pass Connector 238
Fail Connector 238
Conditional Connector 238
Viewing General Relationship Properties and Modifying Relationships. 240
Deleting Relationships in a Job 242
Saving Jobs 243
Job Step Results. 245
Output File Location. 245
Output File Permissions 246
Output File Metadata 247

11 Running Jobs 250

On-demand Job Execution 250
Specifying On-demand Execution Options 251
Scheduled Job Execution 251
Creating Schedules 252
Editing Schedules 258
Deleting Schedules 258
Message-Based Processing Example 258

12 Monitoring Status 260

Accessing Status Views 260
Selecting a Server in Status Views 261
Opening a Job in the Job Editor 261
Refreshing Status Views. 261
Reordering Items in Status Views 262
Deleting a Job from Status Views 262
Job Schedule View. 262
Job History View. 263
Working With the Job History Table 263
Model Management Views. 265
Model Evaluation View 266
Champion Challenger View 267
Predictors View 269

Filters 270
Filters Common to All Status Views 271
Job Schedule Filters 271
Job History Filters 273
Model Management Filters 275
Server Status View 275
Setting the Refresh Rate 277

13 Notifications and Subscriptions 278

Notifications 278
Job Success and Failure Notifications 278
Job Step Notifications 279
Content Notifications 280
Notifications Based On Model Evaluation Return Codes 281
Notification Settings 281
Subscriptions 290
Subscribing to Files 290
Modifying File Subscriptions and Unsubscribing from Files 291
Subscription E-mail Address 291
Managing Subscriptions 292
Delivery Failures 292

14 Reporting Job Steps 294

General Properties for Reporting Steps 294
Data Source for Reporting Steps 295
Type for Reporting Steps 296
Parameters for Reporting Steps 298
Results for Reporting Steps 298
Clean-up for Reporting Steps 300
Notifications for Reporting Steps 301

15 Visualization Report Job Steps 302

General Properties for Visualization Report Steps 302
Data Source for Visualization Report Steps 303

Type for Visualization Report Steps . . . 304
Parameters for Visualization Report Steps . . . 305
Results for Visualization Report Steps . . . 306
Clean-up for Visualization Report Steps . . . 307
Notifications for Visualization Report Steps . . . 308

16 SAS® Job Steps 309

General Properties for SAS Steps . . . 309
- Additional Arguments for SAS Steps . . . 310

Results for SAS Steps . . . 311
- SAS Result Formats . . . 312

Example SAS Step . . . 313

17 General Job Steps 317

General Properties for General Job Steps . . . 317
Input Files for General Job Steps . . . 318
- Input File Source Location . . . 319
- Input File Target Location . . . 319

Output Files for General Job Steps . . . 320
- Output File Intermediate Location . . . 321

General Job Step Examples . . . 321
- Storing SPSS Inc. PMML Files . . . 321
- Batch Scoring using PASW Statistics PMML Files . . . 325
- Event-based Scheduling . . . 328

18 Message-Based Job Steps 333

General Properties of Message-Based Job Steps . . . 333

19 Notification Job Steps 335

Adding a Notification Job Step to a Job . . . 335
General Information . . . 336

Notification. 337
Updating Notifications 338
Selecting a New Template 340

20 Submitted Jobs 341

Restrictions within the Submitted Jobs Folder. 341
Submitted Jobs and Expiration Dates 342

21 Accessibility 343

Types of Accessibility Support 343
JAWS and Java 343
Keyboard Accessibility 343
General Shortcuts 343
Content Explorer Shortcuts 344
Table Shortcuts 344
Job History and Job Schedule Shortcuts 344
Job Editor Shortcuts 345
Help System Shortcuts 345

22 Support Information 346

Support Resource Overview 346
Documentation Resources. 346
Technical Support 346

Index 347

Chapter

1

Overview

PASW Collaboration and Deployment Services

PASW Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics. PASW Collaboration and Deployment Services provides centralized, secure, and auditable storage of analytical assets and advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms for delivering the results of analytical processing to the end users. The benefits of PASW Collaboration and Deployment Services include:

- safeguarding the value of analytical assets
- ensuring compliance with regulatory requirements
- improving the productivity of analysts
- minimizing the IT costs of managing analytics

PASW Collaboration and Deployment Services allows you to securely manage diverse analytical assets and fosters greater collaboration among those developing and using them. Furthermore, the deployment facilities ensure that the right people get the information they need to take timely, appropriate action.

Collaboration

Collaboration refers to the ability to share and reuse analytic assets efficiently, and is the key to developing and implementing analytics across an enterprise. Analysts need a location in which to place files that should be made available to other analysts or business users. That location needs a version control implementation for the files to manage the evolution of the analysis. Security is required to control access to and modification of the files. Finally, a backup and restore mechanism is needed to protect the business from losing these crucial assets.

To address these needs, PASW Collaboration and Deployment Services provides a repository for storing assets using a folder hierarchy similar to most file systems for organization. Files stored in the repository are available to users throughout the enterprise, provided those users have the appropriate permissions for access. To assist users in finding assets, the repository offers a search facility.

Analysts can work with files in the repository from client applications that leverage the service interface of PASW Collaboration and Deployment Services. Products such as PASW Statistics and PASW Modeler allow direct interaction with files in the repository. An analyst can store a version of a file in development, retrieve that version at a later time, and continue to modify it until it is finalized and ready to be moved into a production process. These files can include custom interfaces that run analytical processes allowing business users to take advantage of an analyst's work.

The use of the repository protects the business by providing a central location for analytical assets that can be easily backed-up and restored. In addition, permissions at the user, file, and version label levels control access to individual assets. Version control and object version labels ensure the right versions of assets are being used in production processes. Finally, logging features provide the ability to track file and system modifications.

Deployment

To realize the full benefit of predictive analytics, the analytic assets need to provide input for business decisions. Deployment bridges the gap between analytics and action by delivering results to people and processes on a schedule or in real time.

In PASW Collaboration and Deployment Services, individual files stored in the repository can be included in processing **jobs** that define an execution sequence for the files. The execution results can be stored in the repository, on a file system, or delivered to specified recipients. Results stored in the repository can be accessed by any user with sufficient permissions using the Deployment Portal interface. The jobs themselves can be triggered according to a defined schedule or in response to system events.

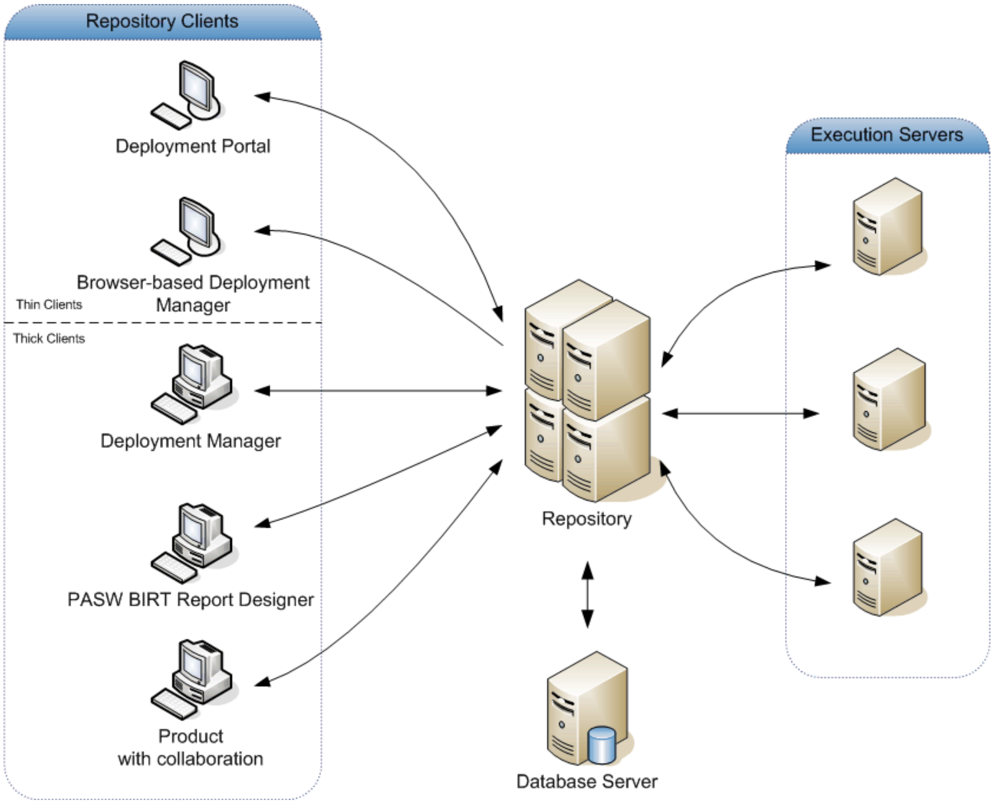
In addition, the scoring service of PASW Collaboration and Deployment Services allows analytical results from deployed models to be delivered in real time when interacting with a customer. An analytical model configured for scoring can combine data collected from a current customer interaction with historical data to produce a score that determines the course of the interaction. The service itself can be leveraged by any client application, allowing the creation of custom interfaces for defining the process.

The deployment facilities of PASW Collaboration and Deployment Services are designed to easily integrate with your enterprise infrastructure. Single sign-on reduces the need to manually provide credentials at various stages of the process. Moreover, the system can be configured to be compliant with Federal Information Processing Standard Publication 140-2.

System Architecture

In general, PASW Collaboration and Deployment Services consists of a single, centralized repository that serves a variety of clients, using execution servers to process analytical assets.

Figure 1-1
PASW Collaboration and Deployment Services Architecture



PASW Collaboration and Deployment Services consists of the following components:

- repository for analytical artifacts
- Product with Collaboration
- Deployment Manager
- Deployment Portal
- browser-based Deployment Manager
- Enterprise View
- PASW BIRT Report Designer

Repository

The repository provides a centralized location for storing analytical assets, such as models and data. The repository includes facilities for:

- Security
- Version control
- Searching
- Auditing

The repository requires an installation of a relational database, such as Oracle, IBM DB2 UDB, or Microsoft SQL Server.

Configuration options for the repository are defined using the Deployment Manager or the browser-based Deployment Manager. The contents of the repository are managed with the Deployment Manager and accessed with Deployment Portal.

Deployment Manager

Deployment Manager is a client application that allows users to schedule, automate, and execute analytical tasks, such as updating models or scores, using the repository. Deployment Manager allows a user to:

- View any existing files within the system, including reports, SAS syntax files, and data files.
- Import files into the repository.
- Schedule jobs to be executed repeatedly using a specified recurrence pattern, such as quarterly or hourly.
- Modify existing job properties in a user-friendly interface.
- Determine the status of a job.
- Specify e-mail notification of job status.

In addition, Deployment Manager allows users to perform administrative tasks for PASW Collaboration and Deployment Services, including:

- user management
- security provider configuration
- role and action assignment

Deployment Portal

Deployment Portal is a thin-client interface for accessing the repository. Unlike the browser-based Deployment Manager, which is intended for PASW Collaboration and Deployment Services administrators, Deployment Portal is a web portal serving a variety of users. Deployment Portal includes the following functionality:

- Browsing the repository content by folder
- Opening published content
- Running jobs and reports
- Generating scores using models stored in the repository
- Searching repository content.
- Viewing content properties.
- Accessing individual user preferences, such as e-mail address and password, general options, subscriptions, and options for output file formats.

Browser-based Deployment Manager

The browser-based Deployment Manager is a thin-client interface for performing setup and system management tasks, including:

- Configuring the system.
- Configuring security providers.
- Managing MIME types.

Non-administrative users can perform any of these tasks provided they have the appropriate actions associated with their login credentials. The actions are assigned by an administrator.

Enterprise View

Enterprise View provides a single, consistent view of enterprise data. Enterprise View allows users to define and maintain a common view of warehoused and transaction data needed to perform analytics, optimization, deployment, and reporting. Underlying data may come from a variety of sources, including a data warehouse, an operational data store, and an online transaction database. Enterprise View ensures a consistent use of enterprise data and hides the complexities of stored data structures from the end user. Enterprise View is the data backbone for the predictive enterprise.

Data discovery requires a major investment of resources from the organizations deploying predictive analytics. The process is labor intensive—it can involve representatives from departments across the organization and often entails resolving differences in data structure and semantics across organizational boundaries. Enterprise View provides a mechanism for recording the outcomes of the data discovery process, versioning and securing the resulting schema, and tracking changes over time.

Enterprise View includes the Enterprise View Driver component designed to provide other applications access to Enterprise View objects stored in the repository. The driver operates similarly to ODBC drivers with the exception that it does not directly query a physical data source but rather references Enterprise View data provider definitions and application views. Note that while Enterprise View is installed as part of Deployment Manager, Enterprise View Driver must be installed separately. For more information, see the installation instructions.

Execution Servers

Execution servers provide the ability to execute resources stored within the repository. When a resource is included in a job for execution, the job step definition includes the specification of the execution server used for processing the step. The execution server type depends on the resource.

Execution servers currently supported by PASW Collaboration and Deployment Services include:

- **SAS**. The SAS execution server is the SAS executable file *sas.exe*, included with Base SAS® Software. Use this execution server to process SAS syntax files.
- **Remote Process**. A remote process execution server allows processes to be initiated and monitored on remote servers. When the process completes, it returns a success or failure message. Any machine acting as a remote process server must have the necessary infrastructure installed for communicating with the repository.

For more information, see the topic Server Definitions in Chapter 6 on p. 96.

Execution servers that process other specific types of resources can be added to the system by installing the appropriate adapters. For information, consult the documentation for those resource types.

During job creation, assign an execution server to each step included in the job. When the job executes, the repository uses the specified execution servers to perform the corresponding analyses.

PASW BIRT Report Designer

The reporting functionality of PASW Collaboration and Deployment Services is enabled by BIRT (Business Intelligence and Reporting Tools), an open-source package distributed by Eclipse Foundation under the Eclipse Public License. BIRT provides core reporting features, such as report layout, data access, and scripting. For more information about BIRT, see the BIRT project page (*http://www.eclipse.org/birt*).

The PASW Collaboration and Deployment Services installation includes the BIRT reporting engine server components, which enable the execution of BIRT report syntax files as part of the PASW Collaboration and Deployment Services reporting job steps. PASW BIRT Report Designer is a standalone application that can be used in conjunction with PASW Collaboration and Deployment Services. It provides a rich user interface with a number of advanced features for creating reports and must be installed separately.

If a PASW BIRT Report Designer report requires a JDBC-based database connection, a corresponding JDBC driver must be installed with the repository. For application server-specific information on the location of the JDBC drivers, see the corresponding section of the repository installation instructions.

To start PASW BIRT Report Designer, execute the file *BIRT.exe* in the installation directory. For information on using PASW BIRT Report Designer, see the documentation installed with the application.

Products with Collaboration

A product with collaboration allows interaction with the repository from within the native interface. Files can be stored and retrieved directly from the collaborating product.

In addition, some files stored in the repository can be executed as steps within jobs. A job can contain any number of steps, with each step corresponding to a separate file. Relationships defined between the steps determine the processing flow. The job can be scheduled to execute at a specific time, according to a recurrence pattern, or in response to a defined event. Moreover, notifications can be sent to specified recipients to report on individual step and overall job execution status.

Collaboration between PASW Collaboration and Deployment Services and other products is enabled through the use of adapters. These adapters are installed into the PASW Collaboration and Deployment Services environment to add the product-specific features. For more information, consult the collaborating product documentation.

Chapter
2

What’s New?

New in Release 4.1

The following enhancements have been added to the application:

Suspending Scoring Configurations

At times, it may be necessary to temporarily suspend a scoring configuration from processing requests. Suspending a configuration frees associated system resources and allows various maintenance tasks to be performed without affecting incoming score requests. Resuming a suspended configuration allows the configuration to begin processing scoring requests again.
For more information, see the topic Suspending and Resuming Scoring Configurations in Chapter 9 on p. 225.

Additional Log Settings for Scoring

Additional values can be included in scoring logs, including:

- input fields used for scoring.
- name and IP address of the machine initiating the scoring request.
- user name associated with the scoring request

For more information, see the topic Logging Settings in Chapter 9 on p. 220.

Control Over Scoring Outputs Returned

Scoring requests can produce a variety of scoring outputs, with the type of output being dependent on the model used for scoring. The scoring configuration allows the specification of a subset of outputs to be returned in response to a scoring request. For models that offer many scoring outputs, restricting the returned values to a small set can help to optimize scoring performance.
For more information, see the topic Output Data Returned Settings in Chapter 9 on p. 219.

Multiple Model Refresh

A scenario created in PASW Modeler 14 may define refresh relationships between specific modeling and model nodes. When including the scenario in a job, you can select any of the defined relationships for refresh when the job executes. If the stream in the scenario has no specified refresh relationships, the refresh settings defined when deploying the scenario will be used.

Enterprise View Settings for Scenario and Champion-Challenger Job Steps

The Enterprise View tab specifies the Enterprise View settings for accessing data. For model refresh, this tab lists all Enterprise View nodes. For the score and evaluation scenario types, the list contains only the node associated with the default branch of the stream in the scenario. The data provider for any of the Enterprise View nodes can be changed to any data provider definition in the system that is compatible with the node.

Deprecated Features

The following feature has been deprecated:

- **Windows command work.** In previous releases of the application, the Windows command work option offered the ability to execute an operation via the command line. Windows command work and the General job step were similar in that you could execute operations via the command line level. However, because the Windows command work step essentially provided a subset of the functionality provided by the General job step, the Windows command work step was deprecated.

Existing jobs that contain Windows command work steps should still function. However, new jobs should use the General job step. For more information, see the topic General Job Steps in Chapter 17 on p. 317.

Chapter
3

Getting Started

Launching Deployment Manager

To launch the client:

▶ From the Start menu, choose:

All Programs
  SPSS Inc.
    Deployment Manager

The Deployment Manager interface appears.

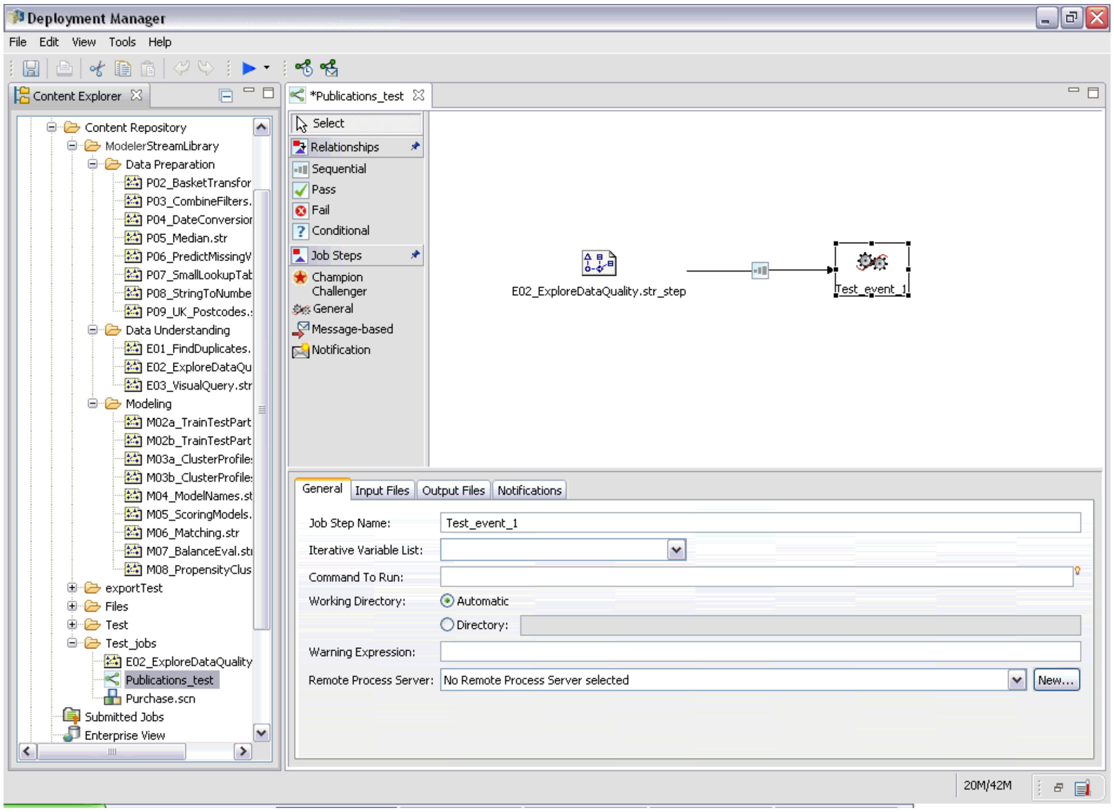
Navigating Through the System

Deployment Manager relies primarily on tab-based navigation. The interface is divided into the following major sections:

Table 3-1
Sections in the Deployment Manager

Section	Description	Location
Content Explorer	The Content Explorer is a tree that displays the contents of your repository. For more information, see the topic Content Explorer Overview in Chapter 4 on p. 15.	Left-hand-pane
Properties window	The Properties window displays the properties of the file that you select in the Content Explorer. Note that this Properties window is different from the individual properties that may appear as part of the Status tabs.	Lower-left-hand pane
Job Editor	The Job Editor allows you to drag-and-drop elements to create jobs.	Upper-right-hand pane

Figure 3-1
Client tool user interface layout



Job Step Name: Test_event_1

Iterative Variable List:

Command To Run:

Working Directory: ● Automatic ○ Directory:

Warning Expression:

Remote Process Server: No Remote Process Server selected

Using the Mouse vs. Pressing Enter

The system is mouse-driven. It is not recommended that you use the Enter key to complete actions. Typically, pressing Enter will not submit your request.

Selecting Items from the File Menu vs. Using the Right-click Method

The majority of the options available on the File menu can also be accessed by right-clicking on an object in the system. In this documentation, the instructions will describe the access via the File menu.

Dragging and Dropping Items in the User Interface

You can drag and drop items in the user interface. For example, you can reorganize items within the Content Explorer, or you can drag and drop files from the Content Explorer into the Job Editor.

The system adheres to the following guidelines for drag-and-drop behavior.

- The Content Repository root cannot be moved.
- You can move items from the Content Explorer into the Job Editor. However, you cannot drag and drop items from the Job Editor to the Content Repository. You must work with items within the Job Editor. For more information, see the topic What Is a Job? in Chapter 10 on p. 228.

Accessing Help

Help is available via an online help system. You can access help using any of the following methods:

Help menu. From the Help menu, choose Deployment Manager Help.

Dialog-level help. To view online help in dialog boxes, click the Help button.

F1 help. In certain sections of the system context-sensitive help is available. To access context-sensitive help, press F1.

Entry Field Content Assist

The content assist feature provides job variables and predefined system property variables that can be used to insert values into entry fields. Variables available through content assist vary from field to field and may include timestamps, object paths, object URLs, job/step start and end times, completion codes, variables passed to a job step by another job step, and variables defined at the job level. Entry fields that allow content assist are marked with a light bulb icon.

To insert variable values into a field, type **$**. The list of available variables appears in a drop-down list. Click the variable name to display its description. Double-click the variable name to select it.

Date and timestamp variables allow the user to select the display format. To select a format after you have specified a variable, type . (period) after the variable name. The list of available formats appears in a drop-down list. Click the format name to display its description. Double-click the format name to select it.

Variables can be used to define file paths. However, a single backslash is an escape character for entry fields. Consequently, use a double backslash or a forward slash when specifying paths. For example, a path of

```
${JobVariable.Var1}\${JobVariable.Var2}
```

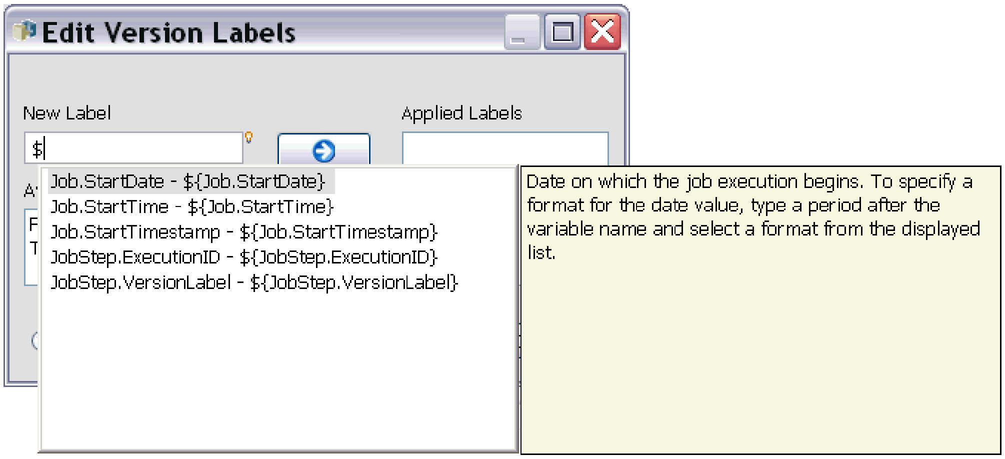
should be specified as

```
${JobVariable.Var1}\\${JobVariable.Var2}
```

or

```
${JobVariable.Var1}/${JobVariable.Var2}
```

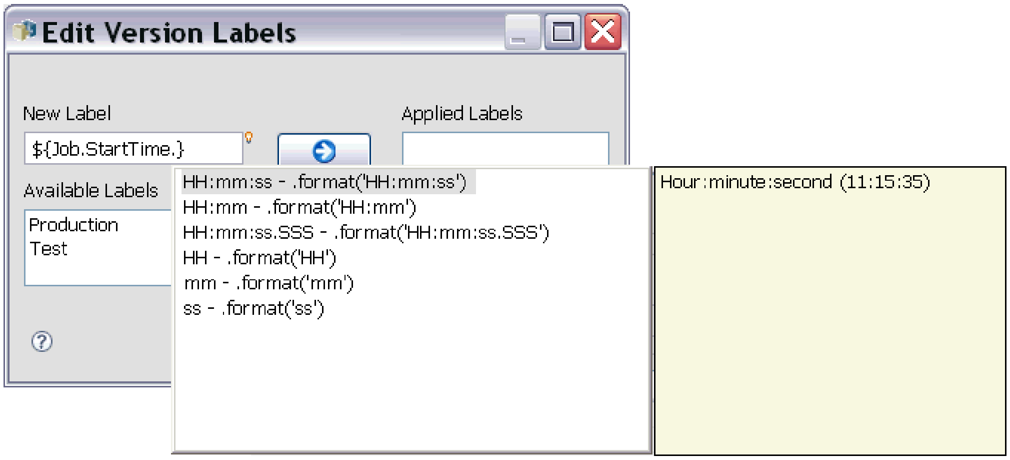
Figure 3-2
Edit Version Labels dialog when selecting a JobStartDate variable using content assist


Edit Version Labels
New Label
Applied Labels
Job.StartDate - ${Job.StartDate}
Job.StartTime - ${Job.StartTime}
Job.StartTimestamp - ${Job.StartTimestamp}
JobStep.ExecutionID - ${JobStep.ExecutionID}
JobStep.VersionLabel - ${JobStep.VersionLabel}

New Label: $

Date on which the job execution begins. To specify a format for the date value, type a period after the variable name and select a format from the displayed list.

Figure 3-3
Edit Version Labels dialog when selecting a format for the JobStartTime variable using content assist



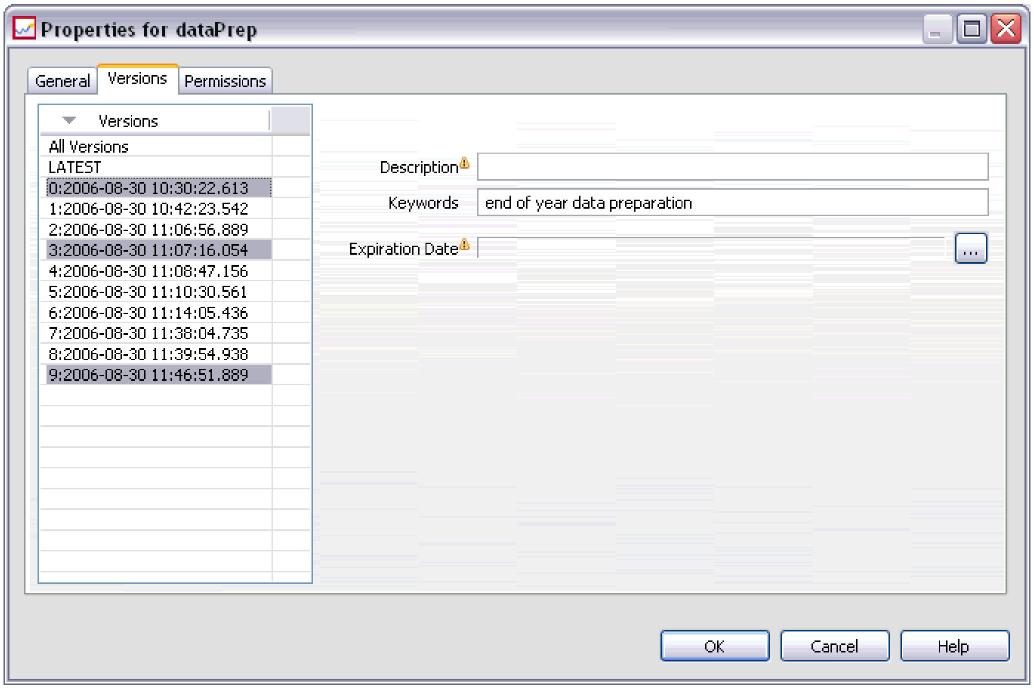
New Label: ${Job.StartTime.}

Hour:minute:second (11:15:35)

Field Conventions

In Deployment Manager, when property values of multiple objects or object versions are compared, property fields that have varying values are empty and marked by alert sign icons.

Figure 3-4
The Properties dialog box with multiple versions selected



Versions

All Versions
LATEST
0:2006-08-30 10:30:22.613
1:2006-08-30 10:42:23.542
2:2006-08-30 11:06:56.889
3:2006-08-30 11:07:16.054
4:2006-08-30 11:08:47.156
5:2006-08-30 11:10:30.561
6:2006-08-30 11:14:05.436
7:2006-08-30 11:38:04.735
8:2006-08-30 11:39:54.938
9:2006-08-30 11:46:51.889

Description

Keywords: end of year data preparation

Expiration Date

Naming Conventions Within the System

At various points in the system, you are prompted to name items. For example, you will be asked to specify names for folders and jobs. All names within the system must be unique.

The following character is not allowed in any name:

- /

Exiting the System

You can exit the Deployment Manager using either of the following methods:

▶ From the File menu, choose Exit.

or

▶ Click the close button (X) in the upper-right hand corner of the user interface.

Chapter

4

The Content Explorer

Content Explorer Overview

The Content Explorer is your entry point into the repository. The Content Explorer, which is located in the left-hand pane, contains a tree hierarchy of objects in the repository. The objects that appear in the Content Explorer depend upon your permissions. For example, you can view only folders for which you have permissions. In the Content Explorer, you can perform the following tasks:

- Log in to and log off from a server.
- Establish server and credential definitions.
- View object properties.
- Access and work with files in the Content Repository.

What Is a Content Object?

A **content object** is any item that resides in the repository. Objects in the repository are stored in a relational database, such as Microsoft SQL Server or Oracle, in binary form. You can add almost any type of file to the Content Repository. Examples of content objects include:

- PASW Modeler streams
- PASW Statistics syntax files
- SAS syntax files
- Jobs
- PASW BIRT Report Designer reports

Typically, you perform an action on a content object. For example, you may add a PASW Modeler stream to a job, or you may run a job. At a minimum, you can move, copy, and paste content objects within the Content Repository.

Organization of the Content Explorer

The server that hosts your repository is represented by a folder in the Content Explorer. Within every server folder, the following folders appear:

Content Repository. This folder contains all of your content objects. You can create subfolders within this folder.

Resource Definitions. This folder contains server, credential, and data source definitions.

These two folders persist permanently at the root. You cannot move, copy, or delete these folders.

Working with Servers

When working with servers, you can perform the following tasks in the Content Explorer:

- Create new server connections.
- Log in to a server.
- Log off from a server.
- Delete a server.

Creating New Content Server Connections

Before you begin working, you must establish a connection to the server that houses your repository. You need to create the server connection only once. After you have created the connection, the server folder appears in the Content Explorer, and you can simply log in to the server. For more information, see the topic Logging in to a Server on p. 17.

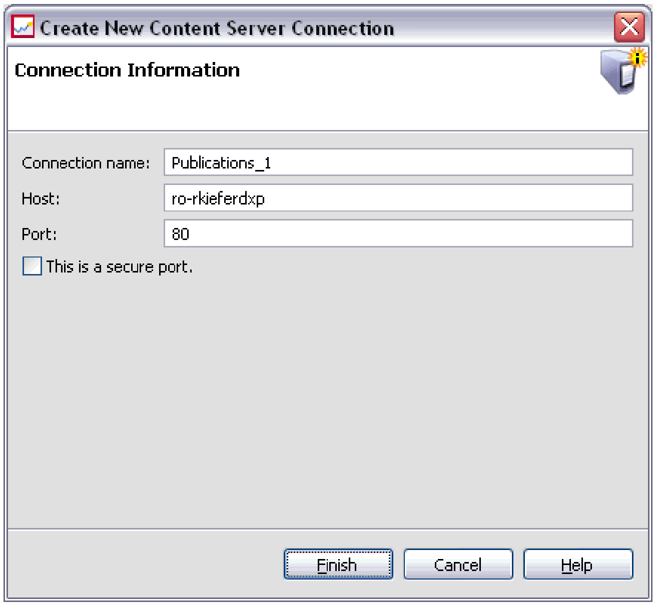
To create a new server connection:

1. Launch Deployment Manager.
2. From the File menu, choose:

   New
     Content Server Connection

   The Create New Content Server Connection dialog box opens.

   Figure 4-1
   Creating a new server connection



Create New Content Server Connection

Connection Information

Connection name: Publications_1

Host: ro-rkieferdxp

Port: 80

□ This is a secure port.

3. In the Connection Name field, enter the name of your repository. This name will appear in the Content Explorer at the root level.

   Note: Alphanumeric characters are recommended. The following symbols are prohibited:

   - quotation marks (single and double)
   - ampersands (&)
   - less-than (<) and greater-than (>) symbols
   - periods
   - commas
   - semicolons

4. In the Host field, enter the name of the machine that houses the repository. You do not need to prefix the hostname with http://.

5. In the Port field, enter the port number for the repository. Note that this port number must match the port number for the host machine that you specified in the previous step. Otherwise, when you try to connect to the server, an error will appear. The default is 80.

6. To use the Secure Socket Layer (SSL) protocol when logging in to this server, select the SSL check box. It's important to note the following when using the SSL protocol:

   - The SSL option is not a global option. It is applied on a per server basis. Thus, when the SSL check box is selected, the SSL protocol is applied only to the current server connection.
   - The SSL protocol must be set up in advance by a system administrator before it can be applied to a server connection.

7. Click Finish.

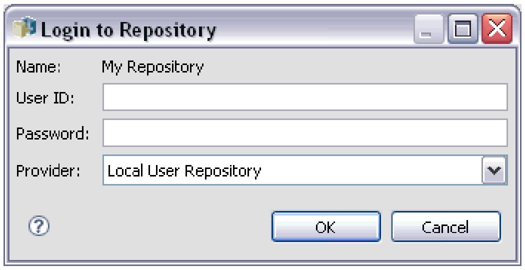
Logging in to a Server

Before you begin working with the Content Explorer, you must log in to a server. The version of the content repository server must be equivalent to or greater than the Deployment Manager version.

At a minimum, you must have one server connection defined in the Content Explorer. For more information, see the topic Server Definition Type in Chapter 6 on p. 97. To log in to a server:

1. Double-click on the server name. (Alternatively, you can expand the server folder by clicking on the + sign.) The Login to repository dialog box opens.

Figure 4-2
Logging in to the server



Name: My Repository

User ID:

Password:

Provider: Local User Repository

2. In the User ID field, enter a valid username for the server. Note that the username that you use determines what you see in the Content Explorer because different users may have different permission levels. For more information, see the topicModifying Permissions in Chapter 5 on p. 41.

3. In the Password field, enter the password that corresponds to the username.

4. If multiple security providers have been configured for the server, use the Provider field to select the security provider against which to validate the user/password combination.

5. Click OK. The system opens the server folder and displays the expanded contents in the server directory.

   Note: If single sign-on has been configured by the PASW Collaboration and Deployment Services administrator, the login page is bypassed and the repository is accessed without the user providing credentials. In that case, the user's Windows credentials are authenticated against an external directory service, for example, Windows Active Directory, which acts as a security provider for the system.

Logging off from a Server

You can access multiple servers simultaneously; however, before logging off from any of them, be sure to save changes to files or jobs on each server because the system does not prompt you to do so.

To log off from a server:

1. In the Content Explorer, right-click on the server from which you want to log off.

2. Select Logoff. You are then logged off from the server, and the contents of the server folder are collapsed.

3. To access stored information, you must log back in to the server. For more information, see the topicLogging in to a Server on p. 17.

Changing Server Passwords

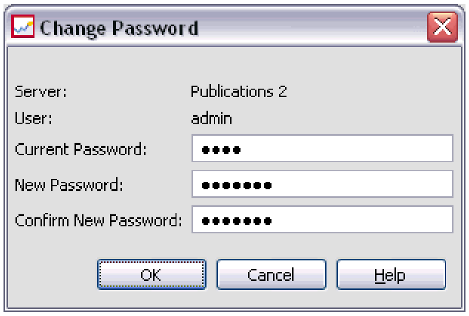
Username and password information is required to log in to a server in the Content Explorer. The password can be changed at any time. However, availability of the change password option depends on the security provider associated with the credentials. For example, passwords can be changed when using PASW Collaboration and Deployment Services native security or IBM i providers, but not when using Active Directory.

The password modification takes effect immediately. Logging off and logging back in to the system is not required.

To change a server password:

1. Right-click on the server name and select Change Password. The Change Password dialog box opens. This dialog box displays the name of the server to which you are currently logged in and your username. Neither the server name nor the username can be modified in this dialog box.

   Figure 4-3
   Change Password dialog box

   Change Password

   Server: Publications 2
   User: admin
   Current Password: ••••
   New Password: •••••••
   Confirm New Password: •••••••

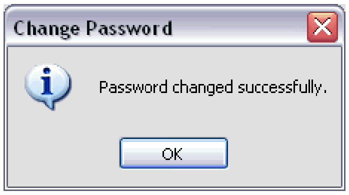
2. In the Current Password field, enter your current password.

3. In the New Password field, enter your new password.

4. In the Confirm New Password field, enter your new password again. The OK button is disabled until all of the fields in the dialog box contain data. In addition, the button is disabled until the information in the New Password and Confirm New Password fields is identical.

5. Click OK. The Change Password dialog box opens showing the message Password changed successfully.

   Figure 4-4
   Password changed successfully

   Change Password

   Password changed successfully.

6. Click OK.

Working with Files

In the Content Explorer, you can perform the following tasks:

- Open external files.
- Add files to the repository.
- Download files from the repository.

Opening External Files

From the Content Explorer, you can open and view files. The method used to open and display the file depends on the file type. For example, if you open a text file, the text is displayed in the Job Editor. However, if you double-click on a PASW Modeler stream, the system launches the PASW Modeler application.

Permissions Inheritance

It is important to understand the relationship between repository resources, particularly when you copy or move resources. The following guidelines apply:

- **Creating resources.** When a resource is added to the repository (for example, when a new job is created), it inherits the permissions of its parent folder; a user with *Write* permissions to a folder he will by default have the same permissions to any resources created in it.
- **Copying resources.** When you copy a resource to a new folder, the resource retains the permissions of the original folder. However, if the user copying the resource is not the owner of the original resource, then the system changes the ownership of the resource to the new user.
- **Moving resources.** When you move a resource from one folder to another, the resource retains the permissions of the original folder. When you cut and paste a resource, this is considered a move. Thus, the resource retains the permissions of the original folder.

Working with the Repository

Files can be added to the repository or downloaded from the repository. Typically, individual files are added or downloaded. It is important to note that adding or downloading files is different from importing or exporting files. For more information, see the topic Export and Import Overview in Chapter 7 on p. 108.

Adding Files to the Repository

You can add almost any type of file to the repository.

Using the File menu. To add a file to the repository:

1. In the Content Explorer, select the folder to which you want to add the file.

2. From the File menu, select Add File to Repository. The Add Files dialog box opens. If the Add File to Repository option is disabled, then you have clicked on an object in the Content Explorer and not on a folder.

3. Navigate to the file that you want to add to the repository.

4. Click Open.

Dragging and dropping files. Alternatively, you can drag and drop files into the repository.

Downloading Files from the Repository

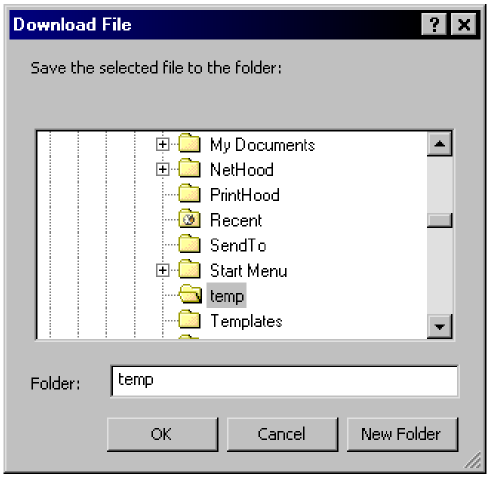
Using the Content Explorer, you can download files from the repository to another computer.

To download files:

1. In the Content Explorer, select the file that you want to download.

2. From the File menu, choose Download File. If your file has multiple versions, the Select File Version dialog box opens. For more information, see the topicSelecting a Version in Chapter 5 on p. 49. Otherwise, the Download Files dialog box opens.

Figure 4-5
Downloading a file



Download File

Save the selected file to the folder:

Folder: temp

3. Navigate to the folder in which you want to place the file.

4. Click OK. A copy of the file is saved in the folder that you specified.

Deleting Files from the Repository

Objects can be deleted from the repository individually or in bulk, provided the appropriate permissions are specified. However, although multiple Enterprise View objects can be selected and deleted simultaneously, Enterprise View objects cannot be deleted in conjunction with non-Enterprise View objects. If an Enterprise View object is selected along with a non-Enterprise

View object, the delete process will fail. For more information, see the topic Enterprise View Overview in Chapter 8 on p. 123.

Searching

Contents of the repository are searchable. The results appear on the Search Results tab. The following types of searches are available:

- Simple search.
- Advanced search.

It is important to note that the search function will search for full text matches only. Currently, partial text and wildcard searches are not supported. Finally, searches cannot be saved and do not persist.

Guidelines for Searching

When searching for objects in the system, the following guidelines apply:

- In order to find an object, the name provided must match the repository object's name exactly.
- The search feature looks inside of PASW Modeler streams to find nodes that contain the search string.
- Although the following objects are returned if they match a search string exactly, the search feature does not look *within* the following types of objects: scenario files, submitted jobs, and resource definitions.
- The following characters are not allowed in search strings: single quotation marks ('), double quotation marks (") and parentheses, (or).
- Certain terms are excluded from the search. For more information, see the topic Terms Not Included in a Search on p. 30.

Accessing the Search Dialogs

To access the search dialogs:

1. Ensure that you are logged into the server that contains the repository that you want to search.

2. In the Content Explorer, select any server folder. The search must be conducted at the server instance level. The search option is disabled at the subfolder level.

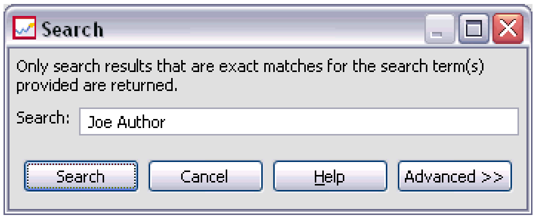
3. From the Edit menu, choose Search. By default, the simple search dialog appears. If the *Search* option is not enabled, then the appropriate folder level was not selected. The search option is enabled only at the server level. All subfolders are automatically searched.

4. To perform an advanced search, click Advanced. For more information, see the topicAdvanced Search on p. 23.

Simple Search

A simple search looks through the repository for objects that match the specified string and returns all matching results. The simple search does not distinguish among property types (e.g., author or title) or narrow results by time frame.

Figure 4-6
Search dialog box



Only search results that are exact matches for the search term(s) provided are returned.

Search: Joe Author

To search for a string:

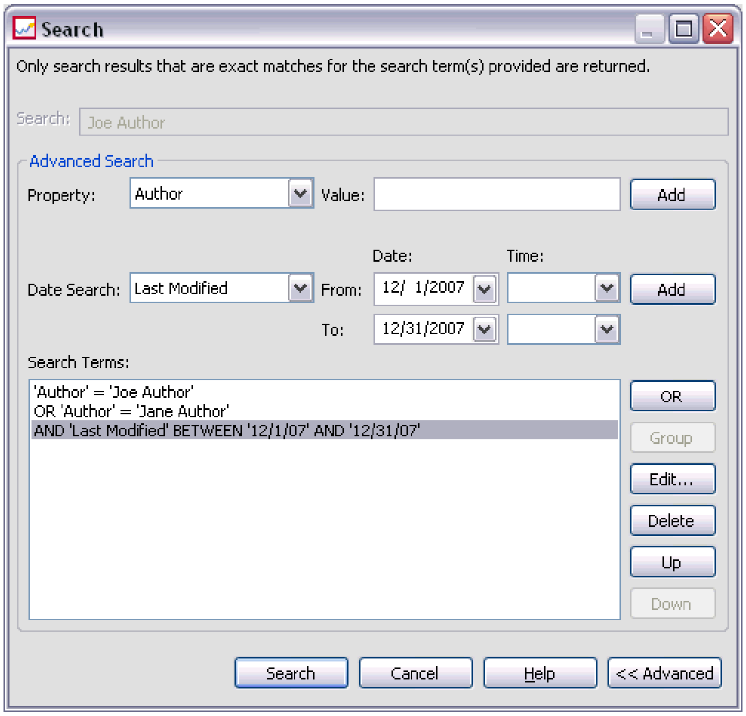
1. In the Search field, enter a text string. Quotation marks are not necessary.

2. Click Search. The Search Results tab is populated with all of the objects that meet the search criteria. For more information, see the topicViewing Search Results on p. 29.

Advanced Search

The advanced search allows greater refinement of the search. For example, in addition to text strings, the advanced search option provides the ability to search by date range.

Figure 4-7
Advanced search dialog



Only search results that are exact matches for the search term(s) provided are returned.

Search: Joe Author

Property: Author
Value:

Date Search: Last Modified
From: 12/ 1/2007
To: 12/31/2007

Search Terms:

'Author' = 'Joe Author'
OR 'Author' = 'Jane Author'
AND 'Last Modified' BETWEEN '12/1/07' AND '12/31/07'

To perform an advanced search, the following options are available:

Property

The list of searchable properties depends on the types of files contained within the repository and may include:

- Author
- Description
- Keyword
- Label
- MIME type
- Title
- Version created by

For any property selected, an exact value needs to be specified. A list of options will not appear for any of these property types. For URI properties, the URI must use the id and version marker specification. Paths and version labels cannot be used when searching by URI.

To search by property:

1. From the Property drop-down list, select a property type.

2. In the Value field, type a value that corresponds to the property type selected. Quotation marks are not necessary.

3. Click Add. The property type and the value appear in the Search Terms box. For the initial entry in the Search Terms box, no AND/OR grouping is specified. By default, any subsequent search terms are combined with the AND operator. The AND operator can be changed to OR at any time. For more information, see the topicSwitching Between AND and OR on p. 25.

Date Search

A date range can also be specified in an advanced search. Valid parameters for a search by date range include:

Date search. Valid values include: *Expiration Date*, *Last Modified*, and *Version Created On*.

Date range. A date range must be provided. The date in the To field must be equal to or greater than the date in the From field.

Time range. The time range is optional. However, to search by a time range a date range must be specified first.

Refining Search Terms

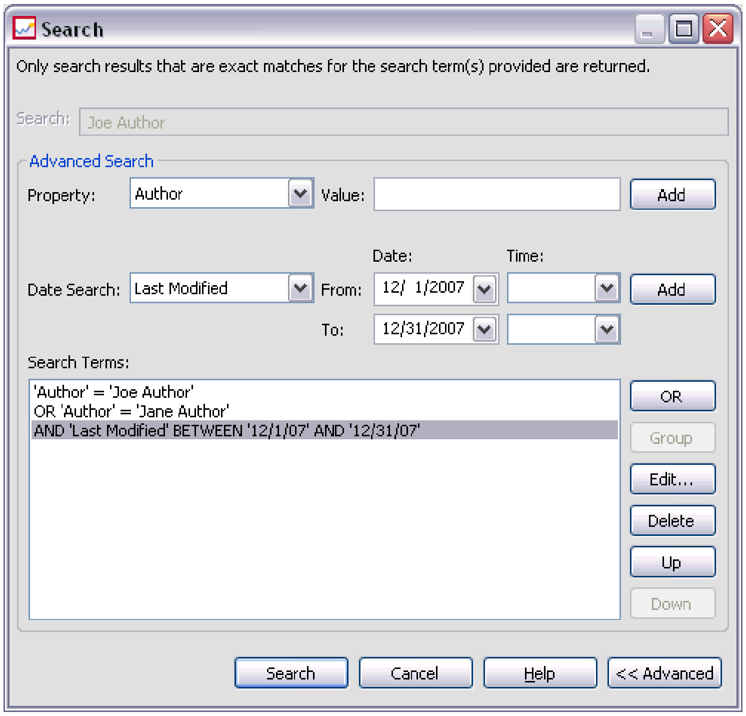
After parameters are specified in the property or date fields and added to the Search Terms list, the selections can be further refined. Specifically, the following enhancements can be made:

- Switching between *AND* or *OR*.
- Grouping search terms
- Editing search term values.
- Reordering search terms.
- Deleting search terms.

Switching Between AND and OR

By default, search terms are joined by the Boolean operation *AND*. The Boolean operator can be changed from *AND* to *OR* and vice versa. The button that appears on the right depends upon the search term selected. If the search term is connected by *AND*, then the OR button appears. If the search term is connected by *OR*, then the AND button appears.

Figure 4-8
Advanced search dialog



Only search results that are exact matches for the search term(s) provided are returned.

Search: Joe Author

Property: Author

Date Search: Last Modified

From: 12/ 1/2007

To: 12/31/2007

Search Terms:

```
'Author' = 'Joe Author'
OR 'Author' = 'Jane Author'
AND 'Last Modified' BETWEEN '12/1/07' AND '12/31/07'
```

To change the connector:

1. Select the search term whose connector you want to change.

2. Click AND or OR.

Grouping and Ungrouping Search Terms

Search terms can be grouped to refine a search further. The ability to perform AND/OR searches allows searches to be organized across attributes. Grouping allows AND/OR searching to occur within the same attribute. Nested groups are supported.

For example, suppose that you want to search for an object modified in December 2007 that could have been created by one of two authors (Joe Author or Jane Author) and carries one of two labels (Test 1 or Test 2). By default, all search terms are joined by *AND*. A search for objects authored by Joe Author and Jane Author AND labeled Test 1 and Test 2 might return limited results. By changing the *AND* between authors and labels to *OR* and using grouping, the search could be refined so that the search results contain a list of objects authored by either Joe Author or Jane Author and labeled either Test 1 or Test 2 and last modified in December 2007.

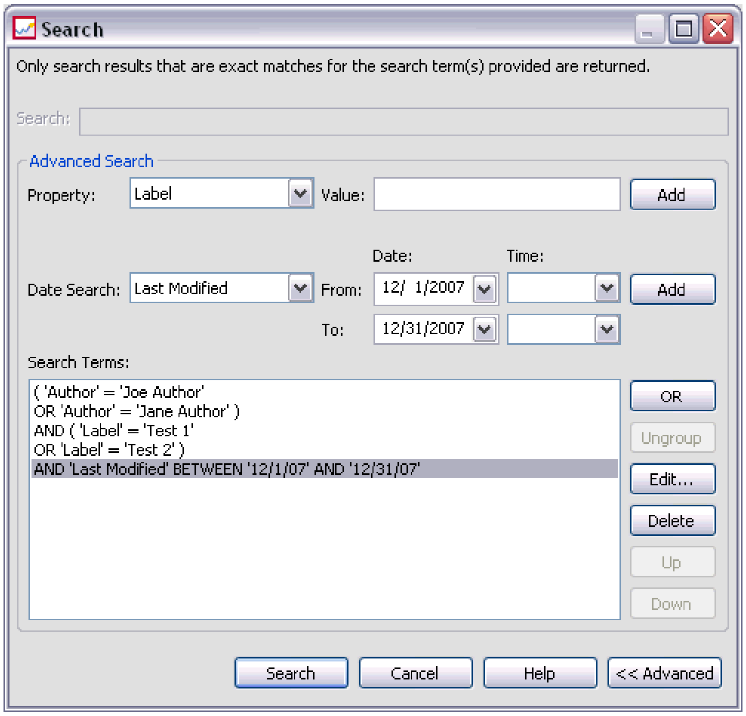
Suppose the following property values were set and appeared in the Search Terms list:

```
'Author' = 'Joe Author'
OR 'Author' = 'Jane Author'
AND 'Label' = 'Test 1'
OR 'Label' = 'Test 2'
AND LastModified BETWEEN '12/1/07' AND '12/31/07'
```

When grouped the properties would be organized in the Search Terms list as follows:

```
('Author' = 'Joe Author'
OR 'Author' = 'Jane Author')
AND ('Label' = 'Test 1'
OR 'Label' = 'Test 2')
AND LastModified BETWEEN '12/1/07' AND '12/31/07'
```

Figure 4-9
Grouping search terms example



Property: Label
Date Search: Last Modified
From: 12/ 1/2007
To: 12/31/2007

```
( 'Author' = 'Joe Author'
OR 'Author' = 'Jane Author' )
AND ( 'Label' = 'Test 1'
OR 'Label' = 'Test 2' )
AND 'Last Modified' BETWEEN '12/1/07' AND '12/31/07'
```

To group search terms:

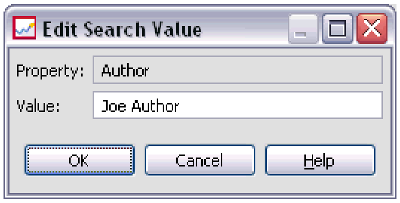
1. Select the search terms that should be evaluated as a group. To group selected items, the items must be adjacent to each other on the list of search terms. To select multiple terms, press Ctrl while selecting rows from the list.

2. Click Group or Ungroup. The *Ungroup* option only appears if the search terms selected were previously grouped together.

Editing Search Terms

The values for search terms can be edited. However, the property type cannot be edited. To add a new property type to the Search Terms list, a new property must be selected from the Property drop-down list and added to the Search Terms list.

Figure 4-10
Editing the search value example



Property: Author
Value: Joe Author

To edit a previously specified property value:

1. Select the search term to edit.

2. Click Edit. The Edit Search Value dialog appears. The contents of the dialog may vary depending upon the search term selected. For example, the dialog for the *Author* property type contains a text field for the author's name; whereas, the dialog for the *Last Modified* property type contains a date range.

3. Specify a new value for the property type.

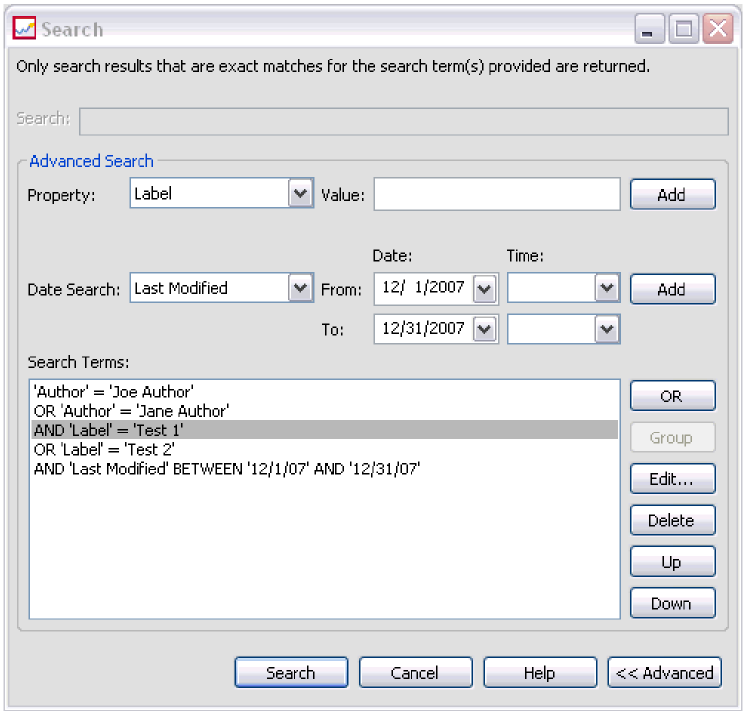
4. Click OK.

Reordering Search Terms

The order in which search terms appear on the list can be reordered. The order of search terms is only important when grouping terms together. Search terms must be adjacent to each other on the search terms list in order for grouping to be possible.

However, the order in which search terms (grouped or ungrouped) appear on the search terms list does not matter. The search feature will return results in order of relevancy. Thus, the search term that matches the largest number of repository objects will be listed first in the search results. The search term with the second largest number of hits will be listed second, and so forth.

Figure 4-11
Reordering search terms



Only search results that are exact matches for the search term(s) provided are returned.

Search:

Property: Label

Date Search: Last Modified

From: 12/ 1/2007

To: 12/31/2007

Search Terms:

'Author' = 'Joe Author'
OR 'Author' = 'Jane Author'
AND 'Label' = 'Test 1'
OR 'Label' = 'Test 2'
AND 'Last Modified' BETWEEN '12/1/07' AND '12/31/07'

To reorder items in the search terms list:

1. From the Search terms list, select the search term to move.

2. Click Up or Down until the search term appears in the desired location. It is important to note that after rows are grouped, the movement of rows is limited. Movement that would potentially affect a grouping is not allowed.

Deleting Search Terms

Search terms can be deleted from the search terms list. After a search term is deleted from the list, it cannot be recovered. The search term would have to be recreated. To delete a search term:

1. Select the search term to delete.

2. Click Delete. The search term is removed from the list.

Viewing Search Results

By default, the Search Results table is hidden. When a search is executed, the Search Results table appears. To open the Search Results table independently:

1. From the View menu, choose Search Results.

The search results table contains the following information.

Table 4-1
Search results

Column	Description
Name	The name of the file that contains the search string.
Folder	The folder that contains the file.
Author	The user who created the file.
Type	The file type. External applications are prefaced with application/x-vnd. For example, the type for a PASW Modeler stream is application/x-vnd.spss-clementine-stream.
Last Modified	Date and time of the most recent modification to the returned item.
Version Count	The number of versions of the search object.

Terms Not Included in a Search

Stopwords are excluded from searches. The words in the following list are not indexed. As a result, the system ignores them when searching for items. Currently, you cannot modify the terms in this list.

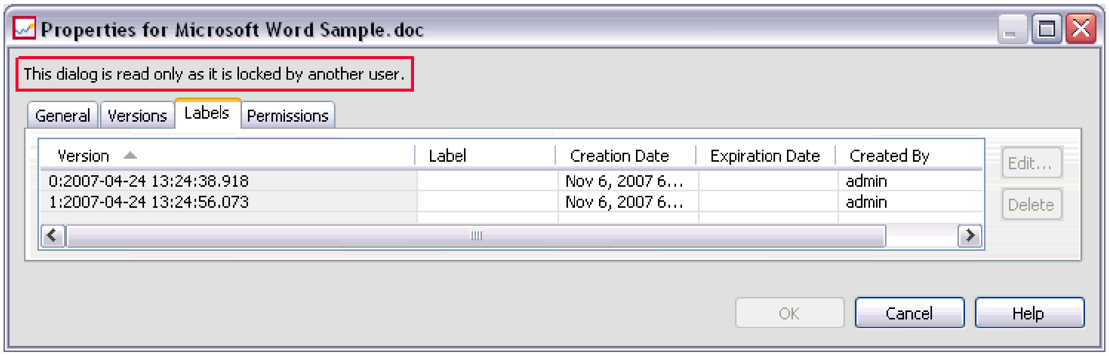
Table 4-2
Stopwords: Terms not included in a search

a	all	am	an
and	any	are	as
at	be	but	by
can	could	did	do
does	etc	for	from
goes	got	had	has
have	he	her	him
his	how	if	in
is	it	let	me
more	much	must	my
nor	not	now	of
off	on	or	our
own	see	set	shall
she	should	so	some
than	that	the	them
then	there	these	this
those	though	to	too
us	was	way	we
what	when	where	which
who	why	will	would
yes	yet	you	

Object Locking

In the Content Explorer, objects can be locked to prevent other users from making changes. For example, suppose that a PASW Modeler stream within a job needs to be modified. Locking the stream would prevent others users from making changes that could potentially conflict with the current user's changes. If another user opens a locked object, the new user receives a message indicating that the file is locked and can only be opened in read-only mode.

Figure 4-12
Locked object opened by another user



This dialog is read only as it is locked by another user.

Version	Label	Creation Date	Expiration Date	Created By
0:2007-04-24 13:24:38.918		Nov 6, 2007 6...		admin
1:2007-04-24 13:24:56.073		Nov 6, 2007 6...		admin

The following guidelines apply to locked objects:

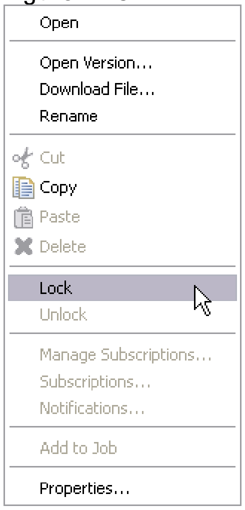
- Any object in the repository—including resource definitions—can be locked. However, folders cannot be locked.
- When an object is locked, all versions of that object are locked. Thus, properties of that object cannot be edited. However, properties will be visible in read-only mode.
- Locks, which are denoted by the lock icon, persist across sessions.
- The locked object cannot be renamed, moved, or deleted. However, the locked object can be copied. The lock is not copied along with the object.
- Objects remain locked until the lock is explicitly removed. Expiration dates cannot be associated with a lock.
- Any user can lock an object. Users can also unlock their own objects. However, to unlock additional objects, users must have the action, *Manage Locks*.

Locking Objects

To lock an object:

▶ In the Content Explorer, right-click on the object and select Lock. The lock icon appears to the left of the object.

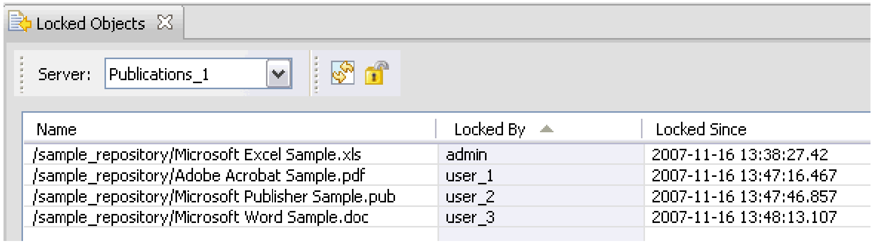
Figure 4-13



Viewing the Locked Objects Table

The Locked Objects table contains a list of locked objects and corresponding details about those objects. Specifically, the Locked Objects table contains the following information:

Figure 4-14
Locked objects table



Name	Locked By	Locked Since
/sample_repository/Microsoft Excel Sample.xls	admin	2007-11-16 13:38:27.42
/sample_repository/Adobe Acrobat Sample.pdf	user_1	2007-11-16 13:47:16.467
/sample_repository/Microsoft Publisher Sample.pub	user_2	2007-11-16 13:47:46.857
/sample_repository/Microsoft Word Sample.doc	user_3	2007-11-16 13:48:13.107

Name. The name and path of the object that is locked.

Locked By. The user who locked the object.

Locked Since. The date and time at which the object was initially locked.

To access the Locked Objects table:

▶ From the View menu, select:

Show View
  Locked Objects

Unlocking Objects

Objects can be unlocked using either of the following methods:

Content Explorer. To unlock an object, right-click on the object and select Unlock. The object is unlocked, and the lock icon is removed.

Locked objects table. To unlock an object:

1. Navigate to the Locked Objects table.
2. Select the object(s) to unlock. To select multiple rows, press the Ctrl key and click simultaneously.

   Note: Only objects for which the user has appropriate permissions can be unlocked.

3. Click the Unlock icon.

Figure 4-15

The selected objects are unlocked and removed from the Locked Objects table.

Exporting and Importing Locked Objects

By default, locks do not persist from one repository instance to the next. If a locked object is exported, the lock is not exported along with the object. The only exception is a lock that the owner exports. If the owner of a locked object exports that object and subsequently imports that object, the lock will be preserved in the imported object. The lock does not need to be reapplied.

Obtaining Updates

Periodically, updates may be available on the server. Each time Deployment Manager is launched, the system checks the server for updates. If updates are available, the system automatically applies the updates and sends a message indicating that the updates have been applied. After updates have been applied, the Deployment Manager needs to be restarted for the changes to take effect.

Chapter

5

Properties

Working With Object Properties

Properties are metadata associated with a content object, which describe the overall content object. Author and modification date are characteristics associated with a content object. Typically, more properties are available as you delve deeper into the content tree. For example, for a folder at the root of the content tree, *title* is the only property available. However, when you view properties for a PASW Modeler stream, you see additional properties, such as author, description, version, and so on.

When working with properties, you can:

- View properties
- Edit properties

Viewing Object Properties

When you select an object in the Content Explorer, properties appear in the Properties pane, which is located below the Content Explorer. The type and number of properties that are displayed depend upon the object and your location within the content tree. For example, if you click on a server folder at the root level, you will see a title in the Properties pane.

Although you view object properties in the Properties pane, you edit them in the Properties dialog box.

Table 5-1
Object properties

Property	**Description**	**Can you view this in the Properties pane?**	**Can you modify this in the Properties dialog box?**
Author	Author refers to the creator of the job. Typically, the author corresponds to a user ID in the system and not an individual person. For example, the author of the job may be the administrator.	Yes	No
Description	The description displayed here corresponds to the description that you specified when you created the object.	Yes	Yes
Modification date	The modification date and time of the most recent modification to the object is displayed here.	Yes	No

Property	**Description**	**Can you view this in the Properties pane?**	**Can you modify this in the Properties dialog box?**
Title	The title is the name of the object.	Yes	Yes. Title corresponds to name in the Properties dialog box.
Version	The version cell contains the following information: <version number>:<modification date><modification time>	Yes	Yes
Name	The name of your object.	Yes. Name corresponds to title in the Properties pane.	Yes
File type	The MIME type associated with the job.	No	Yes
Last modified by	The last user who made changes to the job.	No	No
Custom properties	Properties defined in the Custom Properties dialog box. These properties will vary as they are specific to each server instance.	No	Yes

Editing Object Properties

You may want to modify properties. For example, you may want to add keywords, which are used by the system to search for objects. Properties are grouped into the following categories:

- General
- Labels
- Permissions
- Versions

To access the Properties dialog box:

1. In the Content Explorer, right-click on the object whose properties you want to view.

2. Choose Properties. The Properties dialog box opens.

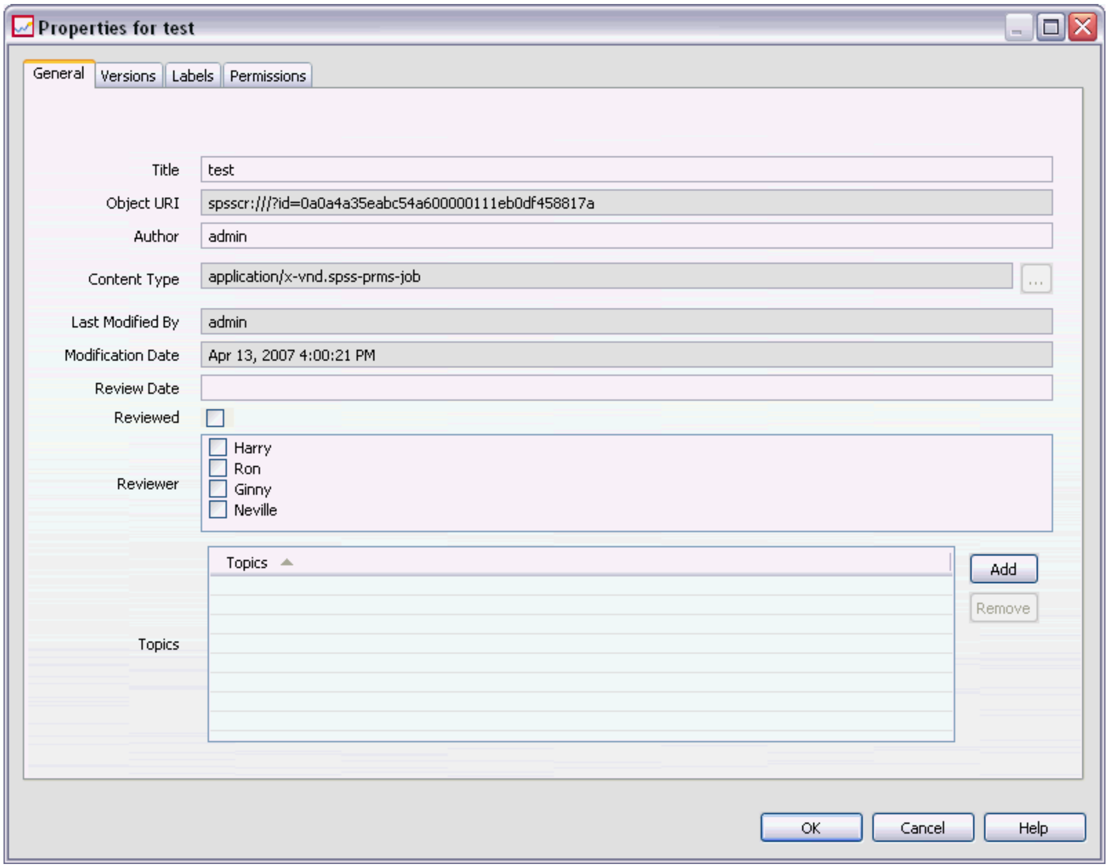
Editing General Properties

To edit general object properties:

1. In the Content Explorer, right-click on the object whose properties you want to view.

2. Choose Properties. The Properties dialog box opens. By default, the General properties dialog box opens.

Figure 5-1
Viewing general job properties



Title: test

Object URI: spsscr:///?id=0a0a4a35eabc54a600000111eb0df458817a

Author: admin

Content Type: application/x-vnd.spss-prms-job

Last Modified By: admin

Modification Date: Apr 13, 2007 4:00:21 PM

Review Date:

Reviewed: □

Reviewer:
□ Harry
□ Ron
□ Ginny
□ Neville

Topics: Topics ▲

The General properties dialog contains information about the object that you selected. Specifically, this dialog describes the following information, which is modifiable, unless otherwise noted.

Title. The name of the object.

Object URI. The uniform resource identifier (URI) for the object. This value is generated by the system and cannot be modified.

Author. The user ID under which the object was created.

Content type. The object type. A new object type can be specified for most objects in the repository. However, a new content type cannot be applied to the following objects:

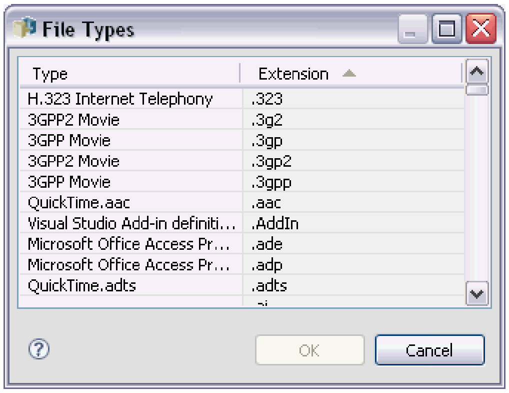
- A job.
- Any object in the *Resource Definitions* folder.
- Any object in the *Submitted Jobs* folder.

Note: Properties can be modified for an object in the *Submitted Jobs* folder once the object has been moved out of that folder into the Content Repository. For more information, see the topic Submitted Jobs in Chapter 20 on p. 341.

To change the file type that is associated with the object:

1. Click the ellipsis button. The File Types dialog box opens.

Figure 5-2
Modifying the file type



File Types

Type	Extension
H.323 Internet Telephony	.323
3GPP2 Movie	.3g2
3GPP Movie	.3gp
3GPP2 Movie	.3gp2
3GPP Movie	.3gpp
QuickTime.aac	.aac
Visual Studio Add-in definiti...	.AddIn
Microsoft Office Access Pr...	.ade
Microsoft Office Access Pr...	.adp
QuickTime.adts	.adts

2. From the Type list, select the new file type.

3. Click OK. The new file type appears in the General properties dialog box.

Last modified by. The user ID under which the last modification was made. This value is generated by the system and cannot be modified.

Modification date. The date and time at which the last modification occurred. This value is generated by the system and cannot be modified.

Custom properties. If custom properties were defined for this server instance, custom property values can be assigned in this dialog.

Topics. If topics were defined for this server instance, topic values can be assigned in this dialog.

Setting Values for Custom Properties

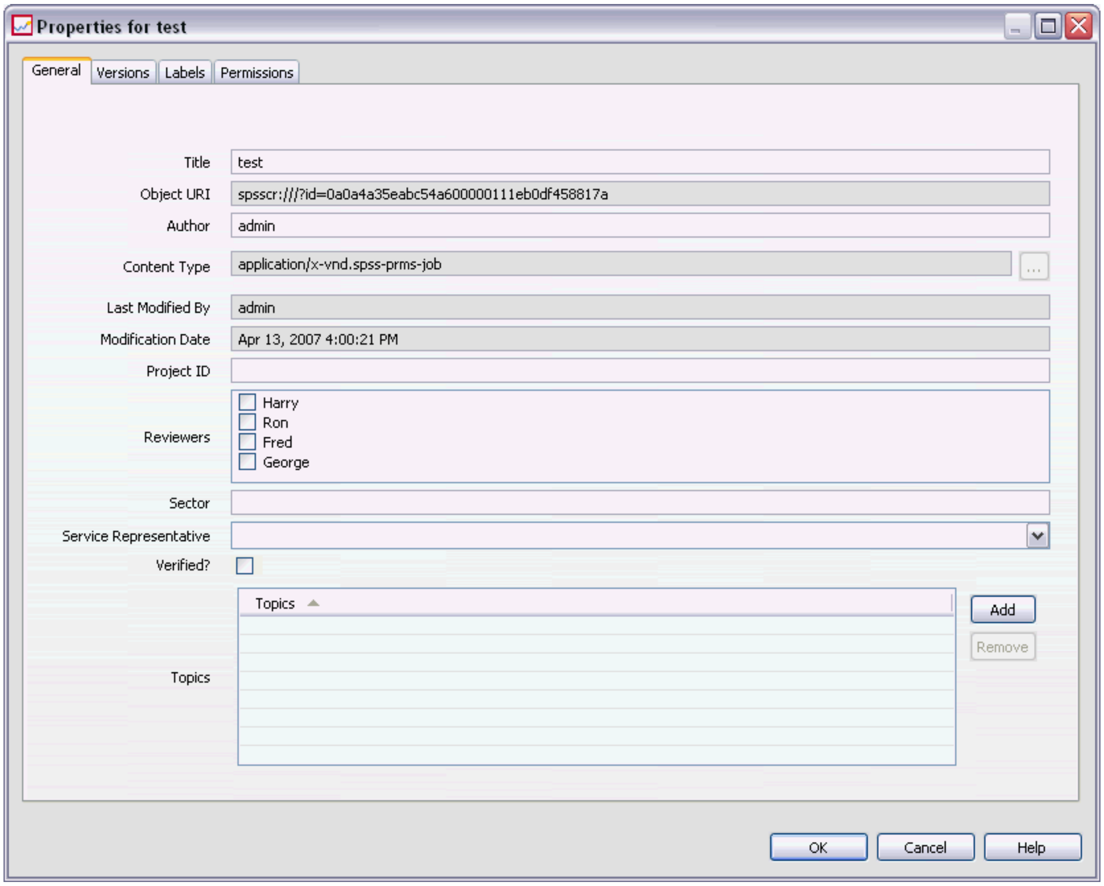
After a custom property has been created, you can assign values to the custom properties fields. Custom properties appear in the General properties dialog box. Specifically, custom properties appear at the end of the dialog box, below the default general properties, after Modification date.

It is important to note that you may not see custom properties for all objects on a server. Custom properties can be restricted to certain objects (for example, jobs) or certain file types.

To set values for an existing custom property:

1. Right-click on any object within the server and choose Properties. The General properties dialog box opens.

Figure 5-3
Custom properties in the general properties dialog box



Title: test

Object URI: spsscr:///?id=0a0a4a35eabc54a600000111eb0df458817a

Author: admin

Content Type: application/x-vnd.spss-prms-job

Last Modified By: admin

Modification Date: Apr 13, 2007 4:00:21 PM

Project ID:

Reviewers: □ Harry □ Ron □ Fred □ George

Sector:

Service Representative:

Verified? □

Topics:

2. Specify the values for the custom properties that appear. The options that appear depend upon the types of custom properties that were created. For more information, see the topicCreating Custom Properties on p. 66.

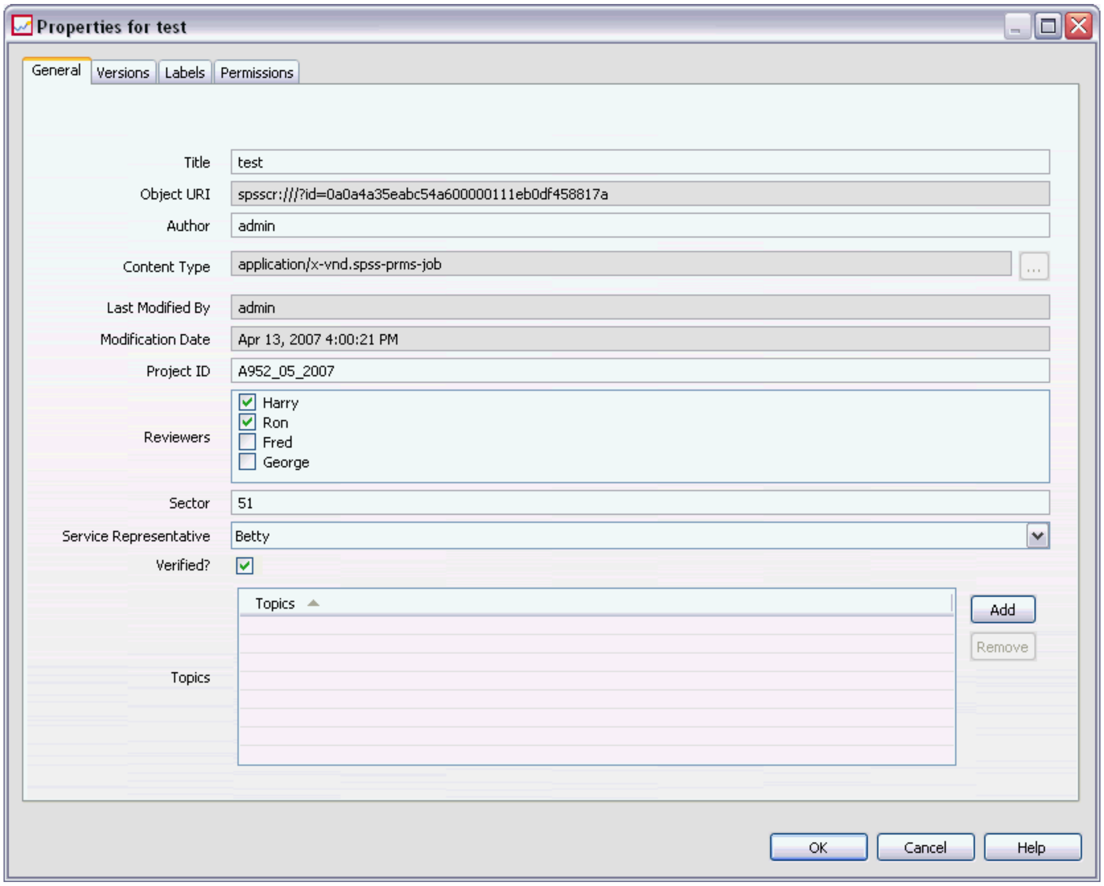
   This example contains the following custom properties:

   Table 5-2
   Custom properties—example

Custom property	**Type**	**Valid values**
Project ID	String	Any string.
Reviewers	Multiple select	One or more selections from the list.
Sector	Number	Any numeric value.
Service representative	Single select	A single selection from the drop-down list.
Verified?	Yes/No	If the check box is selected, this indicates `yes`.

   If you specify values for these properties, the completed dialog box might look like the below example.

Figure 5-4
Custom properties in the General properties dialog box—example



Title: test
Object URI: spsscr:///?id=0a0a4a35eabc54a600000111eb0df458817a
Author: admin
Content Type: application/x-vnd.spss-prms-job
Last Modified By: admin
Modification Date: Apr 13, 2007 4:00:21 PM
Project ID: A952_05_2007
Reviewers: ☑ Harry ☑ Ron □ Fred □ George
Sector: 51
Service Representative: Betty
Verified? ☑
Topics: Topics ▲

3. To save your changes, click OK.

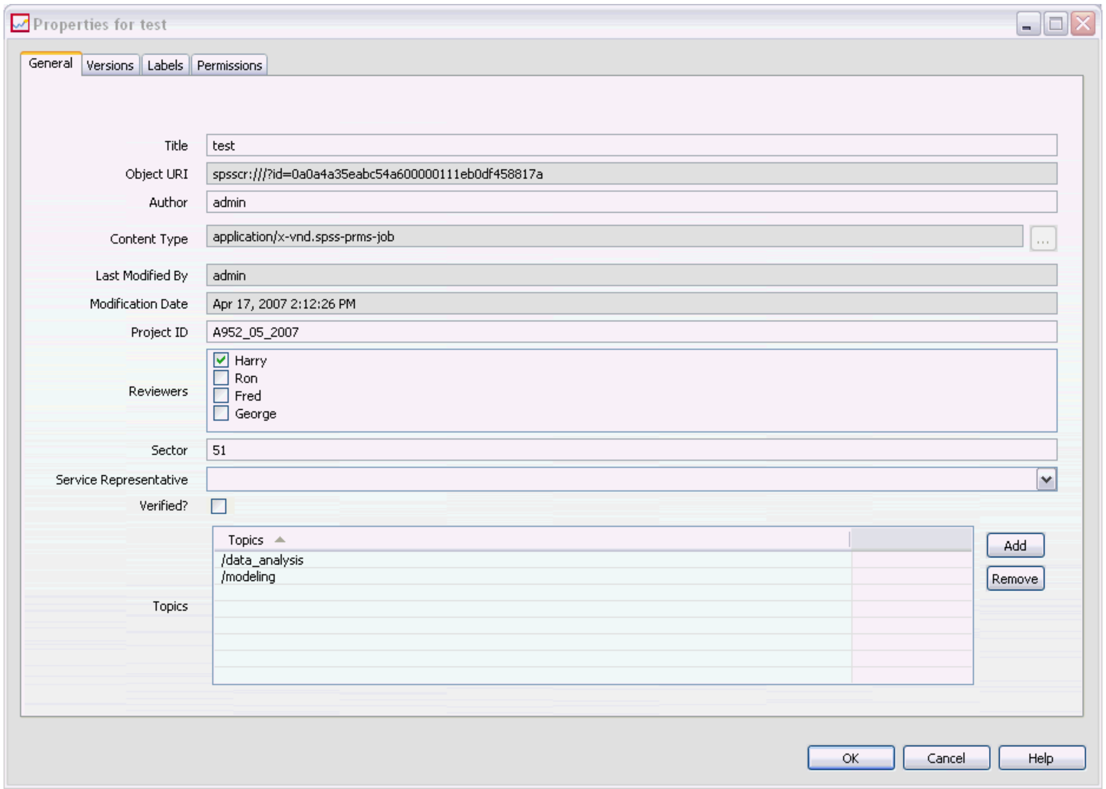
Assigning Topics

After you have created and saved a topic definition, you can assign the topic to a content object.

To assign a topic:

1. Navigate to the General Properties dialog box. The Topics table lists any previously assigned topics.

Figure 5-5
Assigning topics

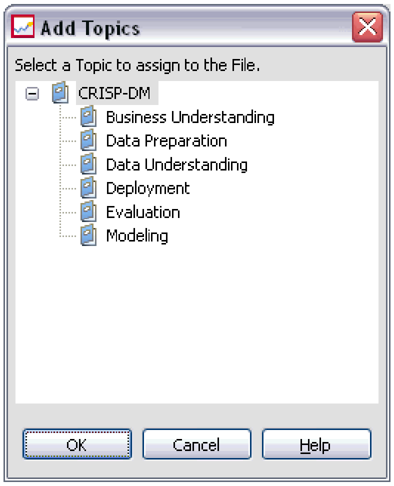


Title: test
Object URI: spsscr:///?id=0a0a4a35eabc54a600000111eb0df458817a
Author: admin
Content Type: application/x-vnd.spss-prms-job
Last Modified By: admin
Modification Date: Apr 17, 2007 2:12:26 PM
Project ID: A952_05_2007
Reviewers: ☑ Harry ☐ Ron ☐ Fred ☐ George
Sector: 51
Service Representative:
Verified? ☐
Topics:

Topics
/data_analysis
/modeling

2. To assign a new topic, click Add. The Add Topics dialog box opens. This dialog box displays the topic hierarchy that was established in the Topics Definition dialog box.

Figure 5-6
Selecting a topic to assign



Select a Topic to assign to the File.

- CRISP-DM
  - Business Understanding
  - Data Preparation
  - Data Understanding
  - Deployment
  - Evaluation
  - Modeling

3. Select a topic from the hierarchy.

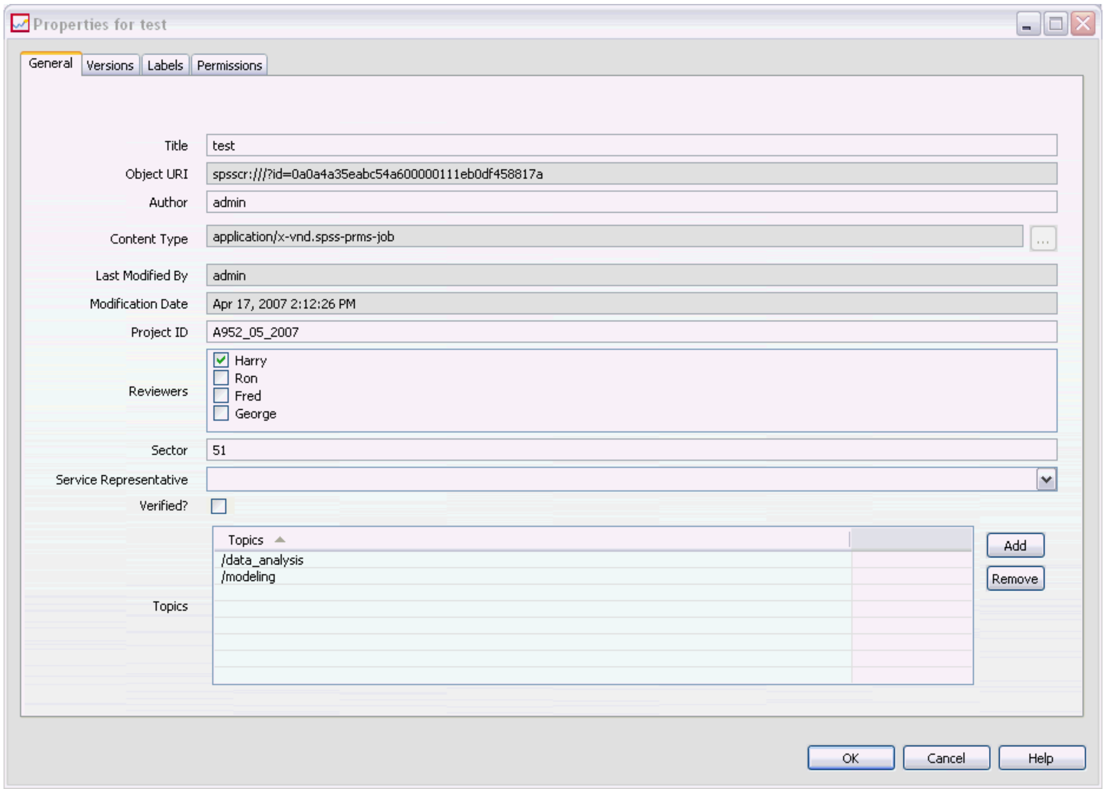
4. Click OK. The Properties dialog box reopens, and the topic is added to the *Topics* table. Note the format of the topic reference. Each slash in the topic reference represents a folder in the topic hierarchy. The first slash represents the uppermost *Topics* folder.

Removing Topics

To remove a topic assignment:

1. Navigate to the General Properties dialog box. The Topics table lists any previously assigned topics.

Figure 5-7
Assigning topics



Properties for test

Title: test
Object URI: spsscr:///?id=0a0a4a35eabc54a600000111eb0df458817a
Author: admin
Content Type: application/x-vnd.spss-prms-job
Last Modified By: admin
Modification Date: Apr 17, 2007 2:12:26 PM
Project ID: A952_05_2007
Reviewers: ☑ Harry □ Ron □ Fred □ George
Sector: 51
Service Representative:
Verified? □

Topics
/data_analysis
/modeling

2. Select a topic from the Topics table.

3. To remove a topic, click Remove. The topic is removed from the Topics table.

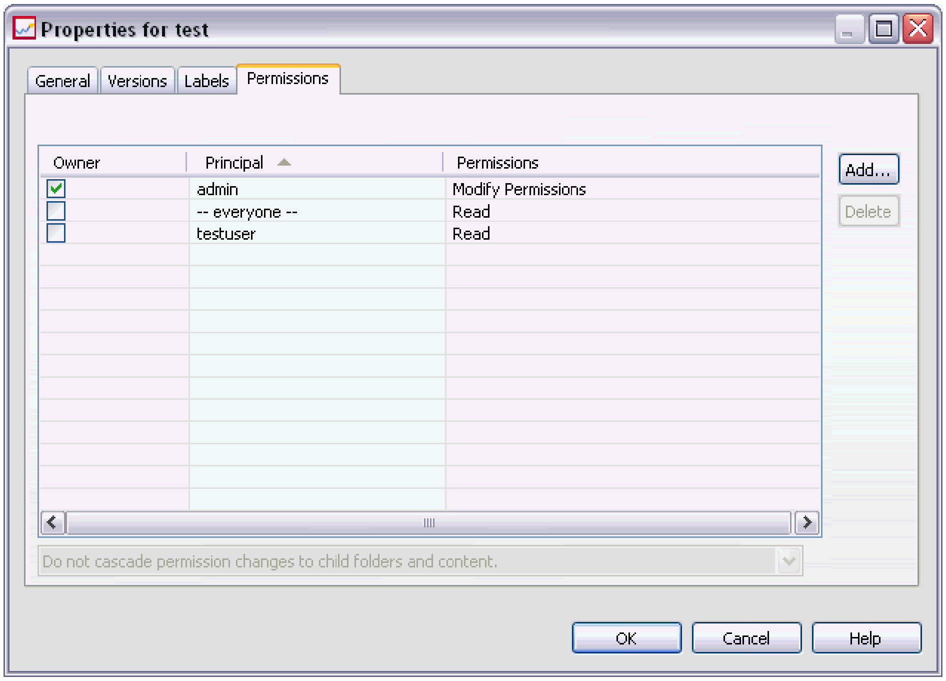
Modifying Permissions

Each object has a permission level associated with it. At times, you may want to modify permissions. For example, if you have added steps to a job and scheduled its execution for the next six months, you may want to restrict other users to read-only permissions to prevent changes to the job.

To access the permissions for the object:

▶ In the Properties dialog box, click the Permissions tab. A list of users and their corresponding permissions appears.

Figure 5-8
Viewing permissions information



Properties for test

Owner	Principal	Permissions
☑	admin	Modify Permissions
□	-- everyone --	Read
□	testuser	Read

Do not cascade permission changes to child folders and content.

You can perform the following tasks in this dialog box:

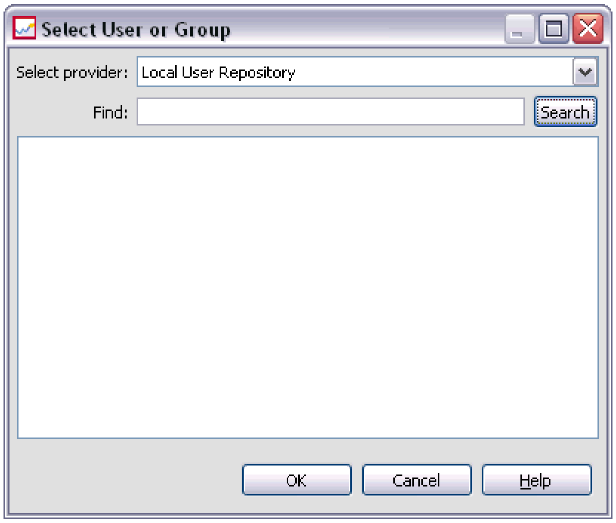
- Add a new user or group
- Modify permissions for an existing user or group
- Delete an existing user or group

Adding New Users or Groups

To add a new user or group and assign permissions:

1. Click Add. The Select User or Group dialog box opens.

Figure 5-9
Adding a user or group

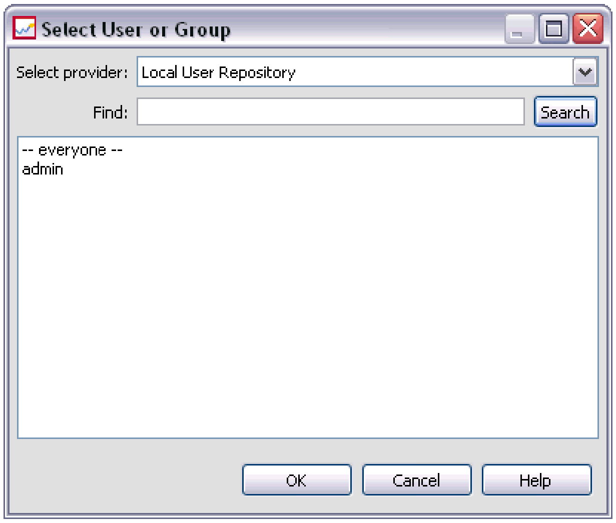


2. From the Select provider drop-down list, select the entity that contains your user and group information. The default is *Local User Repository*.

3. In the Find field, type the first few letters of the user ID you want to add. To search for all available user IDs, leave this field blank.

4. Click Search. The list of users and groups that correspond to your search appears in the dialog box.

Figure 5-10
Adding a user or group

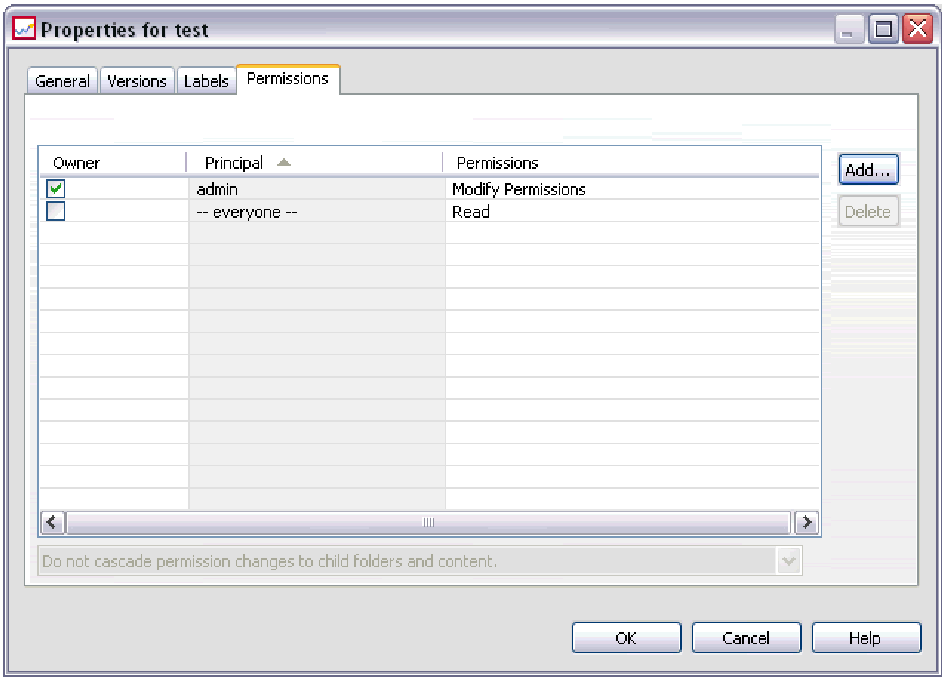


-- everyone --
admin

5. Select a user or group from the list.

6. Click OK. The user or group appears in the Principal list in the Permissions dialog box.

Figure 5-11
New user or group in the principal list



Owner	Principal	Permissions
☑	admin	Modify Permissions
□	-- everyone --	Read

Do not cascade permission changes to child folders and content.

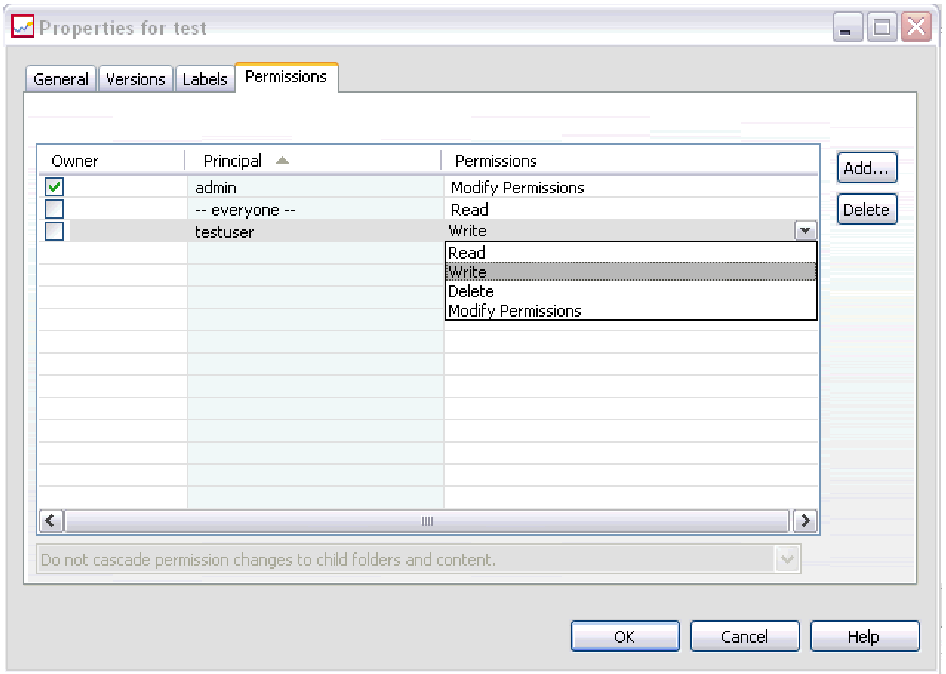
Modifying Permissions for an Existing User or Group

To modify permissions:

1. Click in the Permissions cell that corresponds to the user or group whose permissions you want to modify.

2. From the drop-down list, select a new permissions level. You can select from the following options:
   - Read
   - Write
   - Delete
   - Modify Permissions

Figure 5-12
Modifying permissions



Owner	Principal	Permissions
☑	admin	Modify Permissions
□	-- everyone --	Read
□	testuser	Write

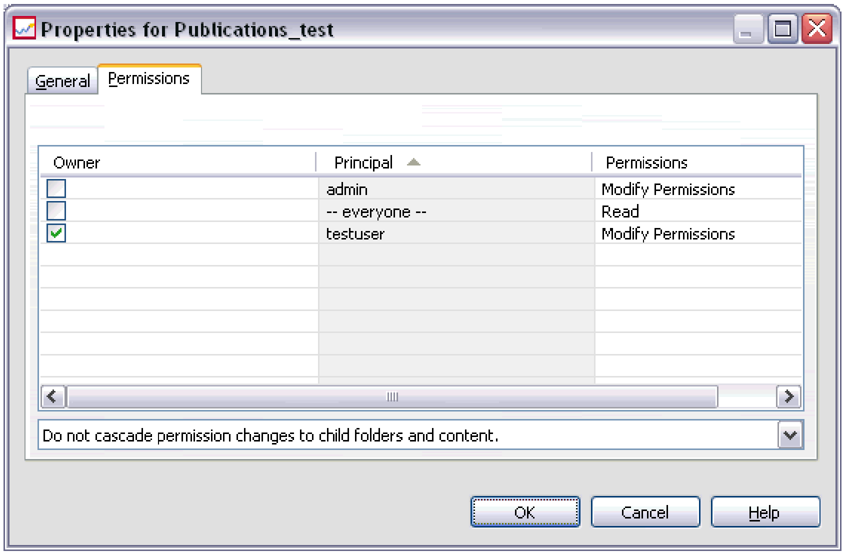
Do not cascade permission changes to child folders and content.

Cascading Permissions

After permissions have been established in the Permissions table, the permissions changes can be cascaded, which means that the content of the current folder will have the same permissions as the parent folder. Permissions can be cascaded completely, partially, or not at all. By default, permissions are not cascaded.

Note: The ability to cascade permissions to an object is controlled by the Modify Permissions option. Permissions are cascaded to expired objects as well as active objects. If an object is not visible because it is expired and the user is not assigned the View Expired action but has Modify Permissions permission, permissions will still be cascaded. If the user does not have Modify Permissions permission, then cascading permissions will not be applied.

Figure 5-13
Cascading permissions example



Owner	Principal	Permissions
□	admin	Modify Permissions
□	-- everyone --	Read
☑	testuser	Modify Permissions

Do not cascade permission changes to child folders and content.

To cascade permissions:

1. Select the level of cascading from the drop-down list. Valid values include:
   - Do not cascade permission changes to child folders and content.
   - Cascade only the updated permissions to all child folders and content.
   - Cascade all permissions to all child folders and content.
2. Click OK.

Deleting a User or Group from the Permissions List

To delete a user or group:

1. Click in the Principal cell of the user or group that you want to remove.
2. Click Delete. The Delete Confirmation dialog box opens.
3. Click OK. The user is removed from the Permissions list.

Working With Labels

The repository allows you to label the existing versions of objects.

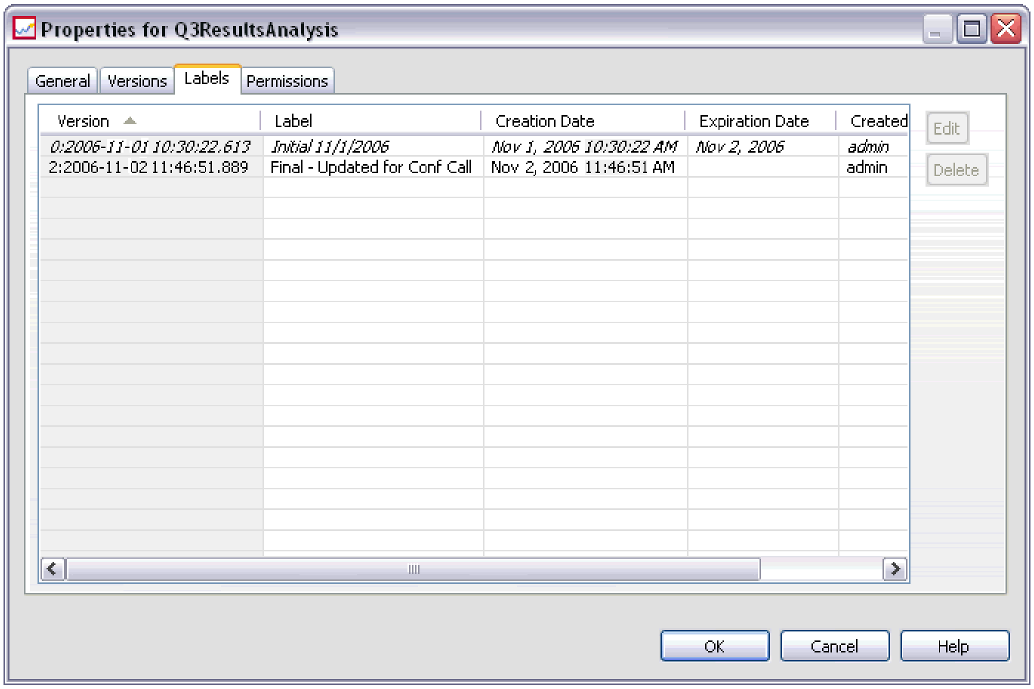
The following information appears on the Labels tab of the Properties dialog box:

Table 5-3
Label properties

Column	Description	Modifiable?
Version	The version number appended to the date and time when the version was created.	No

Column	Description	Modifiable?
Label	The labels for the version. If multiple version labels are available, the labels appear in this column, separated by commas. To view all of the labels, mouse over the Labels cell. The labels will appear in the tooltip. Labels are applied and removed in the Edit Version Labels dialog. For more information, see the topic Editing the Version Label on p. 48.	Yes
Creation Date	The date and time on which the version was created.	No
Expiration Date	The date when the version is set to expire. The expiration date is set on the Versions tab of the Properties dialog box. For more information, see the topic Version Properties on p. 61.	Yes
Created By	The user who created the version.	No

Figure 5-14
Labels tab of the Properties dialog box



Version	Label	Creation Date	Expiration Date	Created
0:2006-11-01 10:30:22.613	Initial 11/1/2006	Nov 1, 2006 10:30:22 AM	Nov 2, 2006	admin
2:2006-11-02 11:46:51.889	Final - Updated for Conf Call	Nov 2, 2006 11:46:51 AM		admin

On the Labels tab, you can perform the following actions:

- Apply a new version label
- Edit an existing version label
- Apply a predefined version label
- Specify multiple version labels
- Remove a version label
- Delete a version

Editing the Version Label

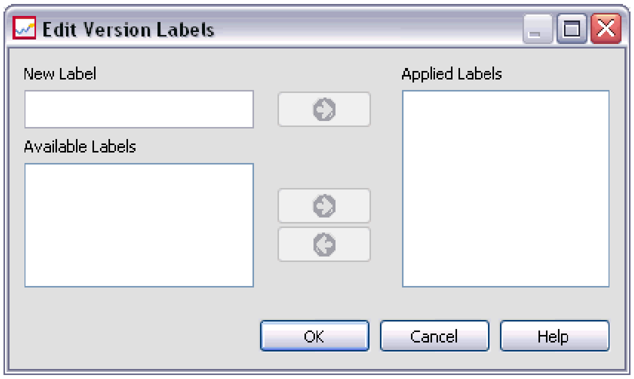
Version labels allow a version of an object to be distinguished from another version of the same object, using a user-defined name instead of a system-generated identifier. For example, version labels are useful when scheduling jobs. In addition, multiple version labels can be associated with an object. If multiple version labels have been applied, all of the labels associated with the object appear in the corresponding Label column of the Versions table.

Version labels persist until they are explicitly changed or removed. It is important to note that removing a version label does not remove the label from the repository. Removing a version label simply severs the connection between an object and the version label. The label is still available for use by other objects.

To edit the version label:

1. On the Labels tab of the Properties dialog, select the object version from the table.
2. Click Edit. The Edit Version Labels dialog box opens.

Figure 5-15
Editing version labels



3. Specify the label(s) to apply (or remove).

Specifying new labels. To apply a new label:

- Type the name in the New Label field. Labels cannot contain commas, apostrophes, semicolons, double quotes, greater than symbols, or less than symbols.
- Click the arrow button to add the label to the Applied Labels field.

Applying available labels. To apply an existing label:

- Select the label(s) from the Available Labels list. To select multiple labels, press Ctrl and make multiple selections.
- Click the arrow button to add the selected label(s) to the Applied Labels list.

Removing existing labels. Removing a label severs the association between the object and the label. To remove a label:

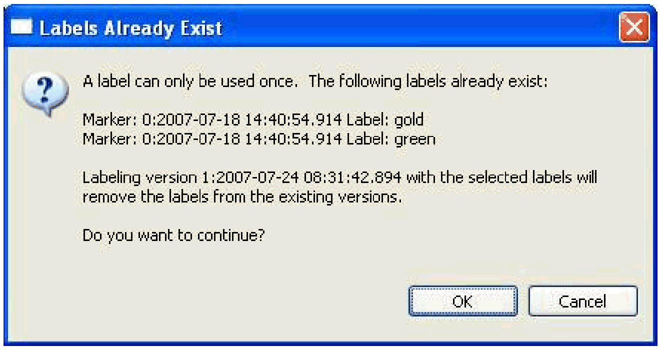
- Select the label from the Applied Labels list.
- Click the arrow button to remove the label from the Applied Labels list.

4. Click OK. The updated Versions table appears.

Resolving Duplicate Version Labels

Although multiple version labels can be applied to an object, each individual version label must be unique for that object. (However, a label with the same name can be used for another object.) In the event of a conflict, the Labels Already Exist dialog appears, which describes the version label duplication.

Figure 5-16
Duplicate version labels



A label can only be used once. The following labels already exist:

Marker: 0:2007-07-18 14:40:54.914 Label: gold
Marker: 0:2007-07-18 14:40:54.914 Label: green

Labeling version 1:2007-07-24 08:31:42.894 with the selected labels will remove the labels from the existing versions.

Do you want to continue?

To resolve a duplicate label conflict, the following options are available:

- **Apply the label.** To apply the label to the existing version and remove the label from the previous version, click OK.
- **Specify a new label.** To maintain the label with the previous version and specify a new label for the current version, click Cancel. The Edit Version Labels dialog reappears. Specify a new label in this dialog.

Deleting Versions

To delete an repository object version:

1. On the Labels tab of the Properties dialog box, select the version.
2. Click Delete. The Delete Confirmation dialog box opens.
3. Click OK. The version is removed from the list.

   Note: Once a version is deleted from repository, it cannot be recovered.

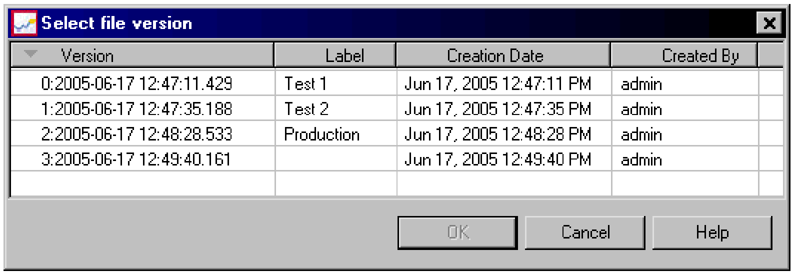
Selecting a Version

At various points in the system, you may be asked to select a version of a file. For example, if you want to download a file that has multiple versions, the system prompts you to select a version.

To select a version:

1. If your file has multiple versions, the Select file version dialog box opens.

Figure 5-17
Selecting a file version



Version	Label	Creation Date	Created By
0:2005-06-17 12:47:11.429	Test 1	Jun 17, 2005 12:47:11 PM	admin
1:2005-06-17 12:47:35.188	Test 2	Jun 17, 2005 12:47:35 PM	admin
2:2005-06-17 12:48:28.533	Production	Jun 17, 2005 12:48:28 PM	admin
3:2005-06-17 12:49:40.161		Jun 17, 2005 12:49:40 PM	admin

2. Select the version of the file on which you want to perform an action.

3. Click OK.

   If the selected version of the file has multiple labels associated with it and the version opens in the Deployment Manager environment, the first label appears with the name of the file in the tab title to assist in identifying the version.

Working with Expiration Dates and Expired Files

An expiration date specifies the date after which a file is no longer actively in use. When an item expires, it is no longer visible to the general user. However, the file owner and administrators can see expired files. Note that expiration does not equal deletion. An expired file is not removed from the repository. By default, files do not expire. An expiration date must be explicitly set. Expiration dates can be set, modified, or reactivated by any of the following:

- The owner of the file
- An administrator
- Any user with write permissions to the file

The file or job itself does not have an expiration date associated with it. Expiration dates are set at the version level. Version numbering continues sequentially. However, a gap in version numbers may appear on the versions list if some file versions have expired.

When a file version expires, any corresponding labels are unaffected. If an expired file version is opened, an error message appears, indicating that the file has expired. When the labeled version of a job expires, it cannot be executed after the expiration date. Thus, any schedules that contain the expired job version are affected.

Schedules are not deleted when the associated version expires. The system will still attempt to execute the scheduled job at the designated time. However, because the job version contained in the schedule has expired, the job will fail, generating an error message in the log. The failure of the scheduled job is recorded in the job history. For more information, see the topic Job History View in Chapter 12 on p. 263.

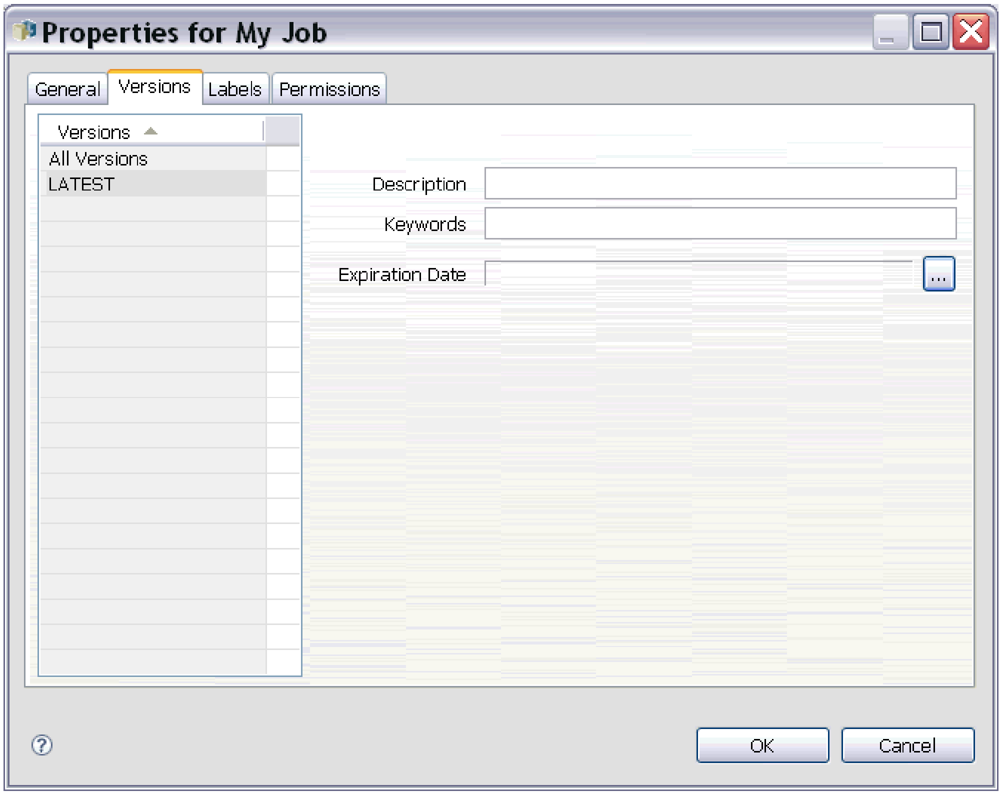
Setting and Modifying Expiration Dates

Expiration dates are set in the Properties dialog box. In addition, an expiration date can also be specified when a job is initially created. By default, objects do not expire. Expiration dates can be modified at any time. In addition, expiration dates can be retroactive. To set an expiration date on an object:

1. In the Content Explorer, right-click on the object and select Properties. The Properties dialog box opens.

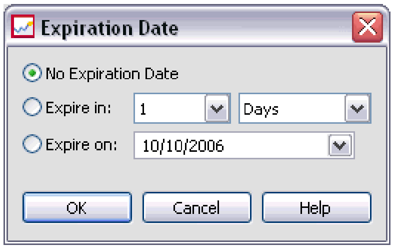
2. Click the Versions tab and select the version on which to specify an expiration date.

Figure 5-18
Expiration date in the General properties dialog box



3. Click the ellipsis button next to the Expiration Date field. The Expiration Date dialog box opens. By default, no expiration date is set.

Figure 5-19
Expiration Date dialog box



Expiration Date

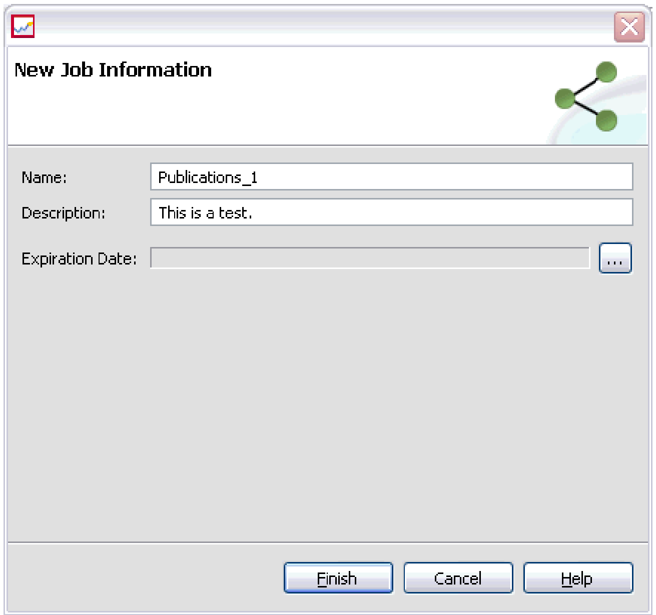
● No Expiration Date
○ Expire in: 1 Days
○ Expire on: 10/10/2006

4. Select one of the following options:
   - **No Expiration Date.** The version does not expire.
   - **Expire in.** The version expires at a time relative to today's date—for example, in 30 days. Valid relative time periods include days, weeks, months, or years. The maximum expiration date allowed is 10 years.
   - **Expire on.** The version expires on the date selected.
5. Click OK. The expiration date appears in the Expiration Date field. For both relative and specific dates, the exact calendar expiration date appears in this field. If no expiration date was specified, the field remains blank.

Specifying Expiration Dates When Creating a New Job

Alternatively, the expiration date can be specified when a new job is created. The process for accessing and supplying information for the expiration date dialog box is the same. However, the dialog box is accessed from the New Job Information dialog box instead of the Properties dialog box.

Figure 5-20
Setting an expiration date when creating a new job



New Job Information

Name: Publications_1

Description: This is a test.

Expiration Date:

Importing and Exporting Expiration Dates

When files are imported into and exported out of the system, the expiration date attached to a file version is included as well.

Restrictions on Expiration Dates

Expiration dates can be applied to most files within the Content Explorer. However, expiration dates cannot be applied to objects in the *Resource Definitions* folder—for example, server, credential, and data source definitions.

Viewing Expired Files

Expired files are visible depending upon whether or not all versions of a file have expired.

Files with some versions expired. For files with some expired versions, the expired versions are displayed. However, any expired versions are grayed out.

Files with all versions expired. The expired files are hidden in the Content Explorer. However, if the owner of the expired files or an administrator views the tree, the expired files are visible.

Searching for Expired Files

The process of searching for expired files is the same as the process of searching for other objects in the Content Explorer. For more information, see the topic Searching in Chapter 4 on p. 22.

To see expired files in the search results, a user must be one of the following:

- A user with the *view expired files* action
- The owner of the file
- An administrator

Reactivating Expired Files

Expired files can be reactivated. To reactivate an expired file version:

1. Navigate to the existing expiration date in the Properties dialog box.
2. Modify the expiration date to a future date.

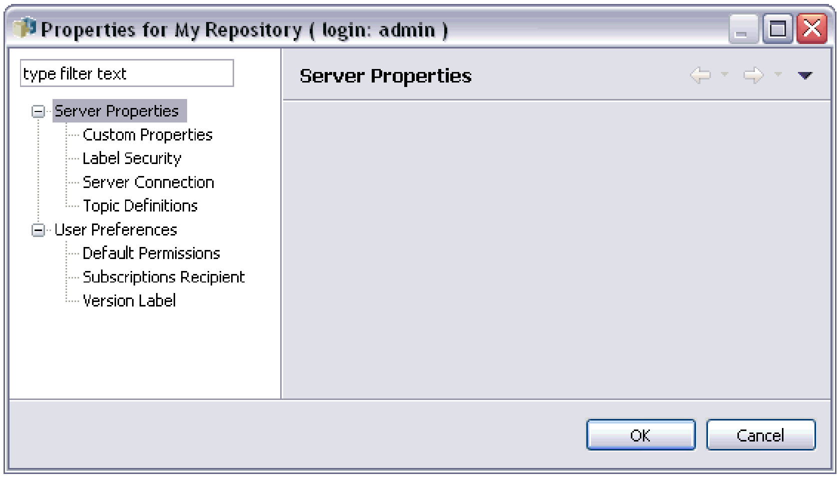
Working With Server Properties and User Preferences

Server properties are divided into the following subcategories:

- Server properties.
- User preferences.

Both types of properties apply to the current server instance (or user) only. To access the Properties dialog box for a server instance, right-click on a server name in the Content Explorer and choose Properties. The Server Properties dialog box opens.

Figure 5-21
Server properties and user preferences



Server Properties

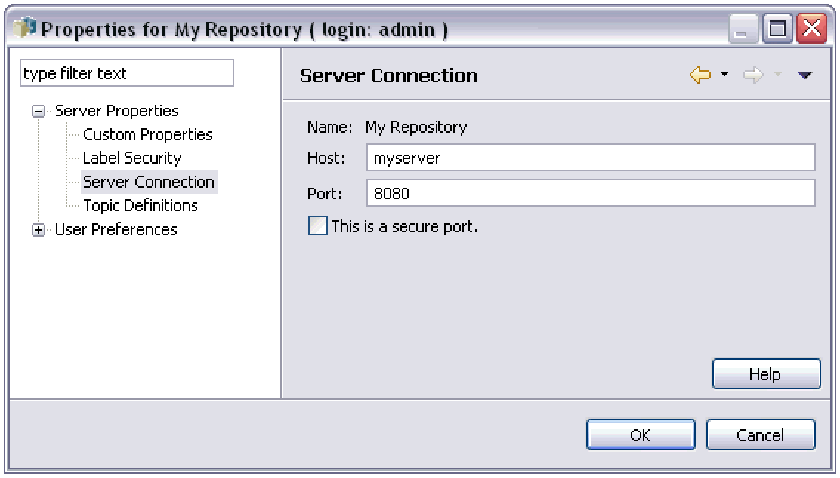
Server properties are similar to content object properties. However, server properties apply to the overall server instance. Users with the appropriate access privileges can view and modify the following:

- Custom properties. For more information, see the topic Working with Custom Properties on p. 64.
- Server connection
- Label security
- Topic definitions. For more information, see the topic Working with Topic Definitions on p. 73.

Server Connection

Server connection properties define the information needed to connect to a repository. To access the connection properties, click the+ to expand the server properties options in the Server Properties dialog box and select Server Connection.

Figure 5-22
Server Connection properties



Name: My Repository
Host: myserver
Port: 8080
□ This is a secure port.

Name. The name for the server connection.

Host. The name or IP address of the machine hosting the repository installation. The name cannot contain quotation marks. Including a prefix of http:// is not necessary.

Port. The port number on the specified host for accessing the repository.

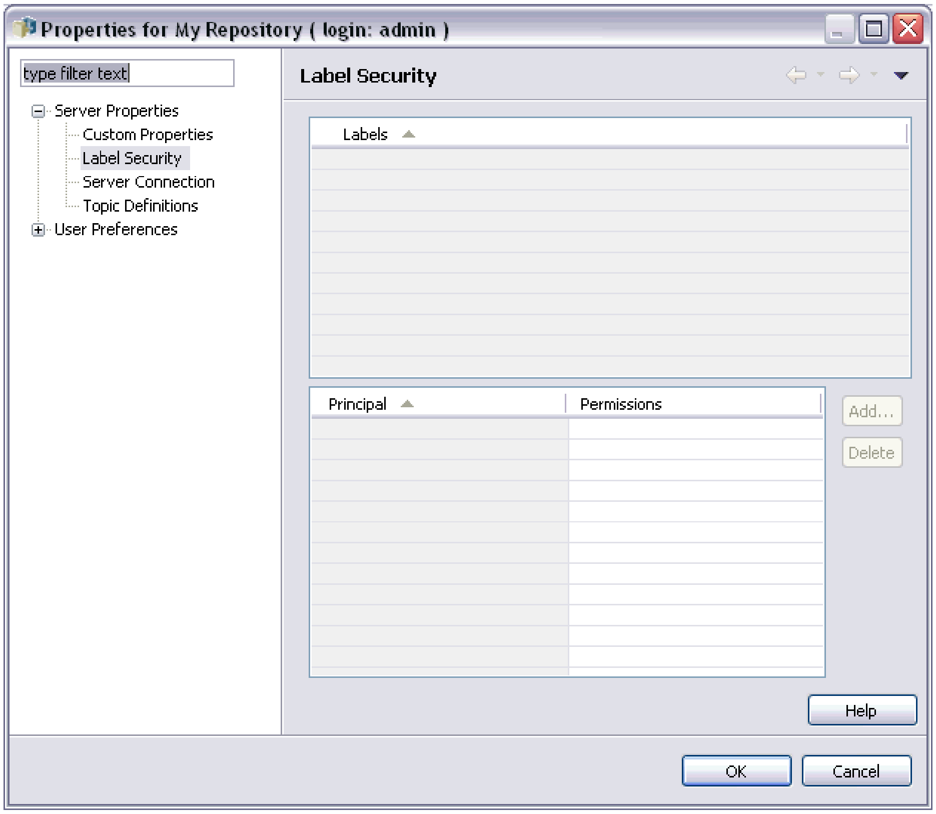
This is a secure port. Select this option if the host has been configured to use the Secure Sockets Layer (SSL) communication protocol.

Label Security

Label security provides control over which principals can view or modify user-defined version labels in the system, allowing the specification of which users can move or delete a label. Although users may be able to view a version of a resource in the repository, only those users with permissions for the labels associated with the version can access the labels. Moreover, label security applies to any version of any resource using the label. For example, to control who can assign production versions of resources, an administrator may restrict access to the *Production* label to prevent other users from moving that label from one version to a newer version. The security defined for the *Production* label applies to every version of every resource using that label.

To access label security properties, click the + to expand the server properties options in the Server Properties dialog box and select Label Security.

Figure 5-23
Label Security properties



The Labels table lists all labels in the system you can access. Select a label to view the associated principals and their permissions

For a selected label, the Principals table identifies label permissions for principals in the system. A principal can have one of the following permissions for a label:

- **Use versions.** Any principal with this permission can view the version of a resource associated with the label by referencing the label.
- **Manage label.** Principals with this permission can apply, move, and delete the label, as well as delete the version of a resource having the label. If a user does not have the *Manage label* permission for a label, that user cannot delete a resource or version of a resource to which the label is applied

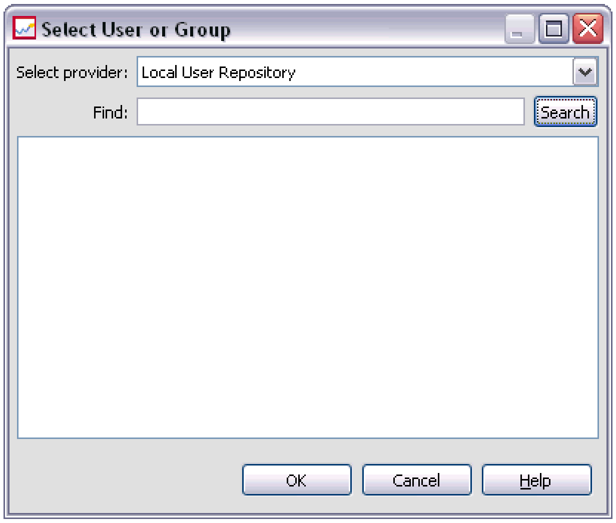
To edit the security for a label, such as adding or deleting principals, you must have the *Manage label* permission for the label. Administrators have this permission for all labels in the system. The *Manage label* permission does allow you to alter your own permission for a label, making it possible for you to edit your permissions for a label in such a way that you will be unable to work with the label in the future. For example, you might change your permission to *Use versions*, which would subsequently prevent you from changing the security for a label. If situations such as this occur, contact your administrator to have your permissions adjusted as needed.

Adding principals for a label

To add a principal for a label:

1. Select the label from the Labels table.
2. Click Add. The Select User or Group dialog box appears.

Figure 5-24
Select User or Group dialog box



Select provider: Local User Repository

Find:

3. Select the principal to add.
4. Click OK.

The principal appears in the Principals table with the default permission.

Modifying permissions for principals

To modify the label permissions for a principal:

1. In the Principals table, click the Permissions cell for the principal.
2. Select the desired permission.

Deleting principals for a label

To delete a principal for a label:

1. Select the label from the Labels table.
2. In the Principals table, select the principal to be deleted. To select multiple principals, hold down the Ctrl key while selecting principals.
3. Click Delete.

The principal is removed from the Principals table and will be unable to access the label.

User Preferences

User preferences apply to the user ID that is currently logged into the server. Users with the appropriate access privileges can view and modify the following:

- Default permissions.
- Subscriptions recipient.
- Version label.

Default Permissions

The default permissions option allows a user to set up a default permissions for new files and folders that are created. Thus, when a user creates a new object, such as a job, the object's permissions list defaults to the list of principals and permissions defined in the user's preferences for the instance of PASW Collaboration and Deployment Services.

In the Default Permissions dialog box, the following changes can be made:

- Add a new principal or delete an existing principal
- Modify permissions for a principal

To specify default permissions:

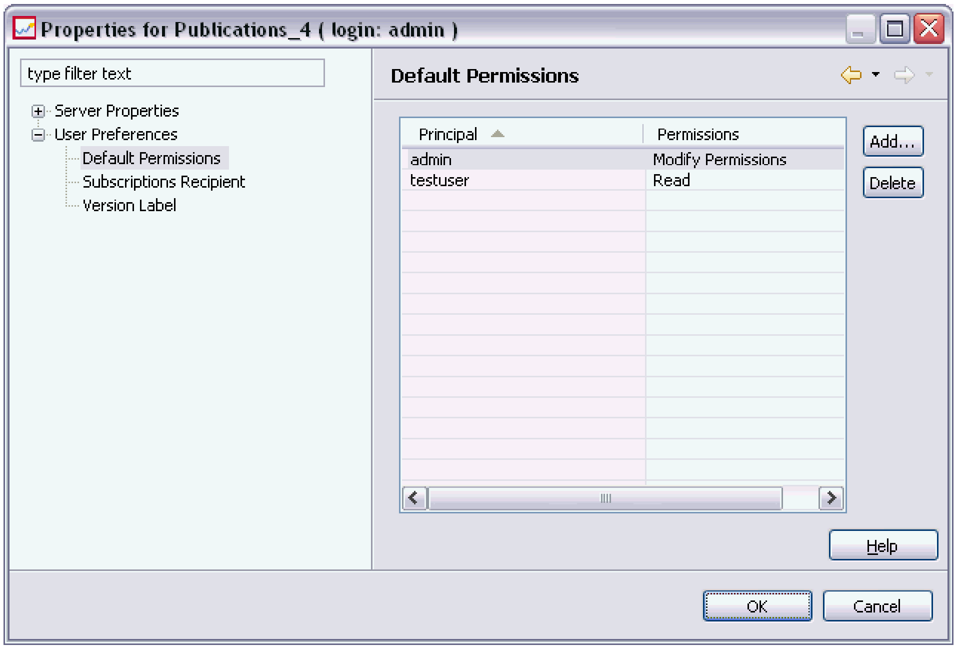
1. Navigate to the Properties dialog box for the server instance.
2. In the Properties tree, expand the User Preferences list and select Default Permissions.
3. Modify the list of principals as needed. Users or groups can be added to or deleted from the Principal list.

Add a new principal. To add a new principal, click Add. The Select User or Group dialog box opens. Select a user or group from the list and click OK.

Delete an existing principal. To delete an existing principal, select a principal from the list and click Delete. The Confirm Deletion dialog box opens. Click OK.

4. To modify the permissions associated with a principal, click the drop-down arrow in the Permissions column and select the permission level to associate with the principal. Valid values include:
   - Read
   - Write
   - Delete
   - Modify Permissions

Figure 5-25
Default Permissions dialog box



Principal	Permissions
admin	Modify Permissions
testuser	Read

5. Click OK.

Important! For members of *administrators* group, the permissions of the newly-created objects will default to Modify Permissions even if they if they are set to a lower level in user preferences.

Subscriptions Recipient

The Subscriptions Recipient option allows the user to choose between the following options:

- Typing in an e-mail address to be used as the default for subscriptions
- Using the e-mail address associated with the user from a backing directory, such as active directory

The e-mail address applies per server instance and per user, which means that a user can specify different e-mail addresses for different servers.

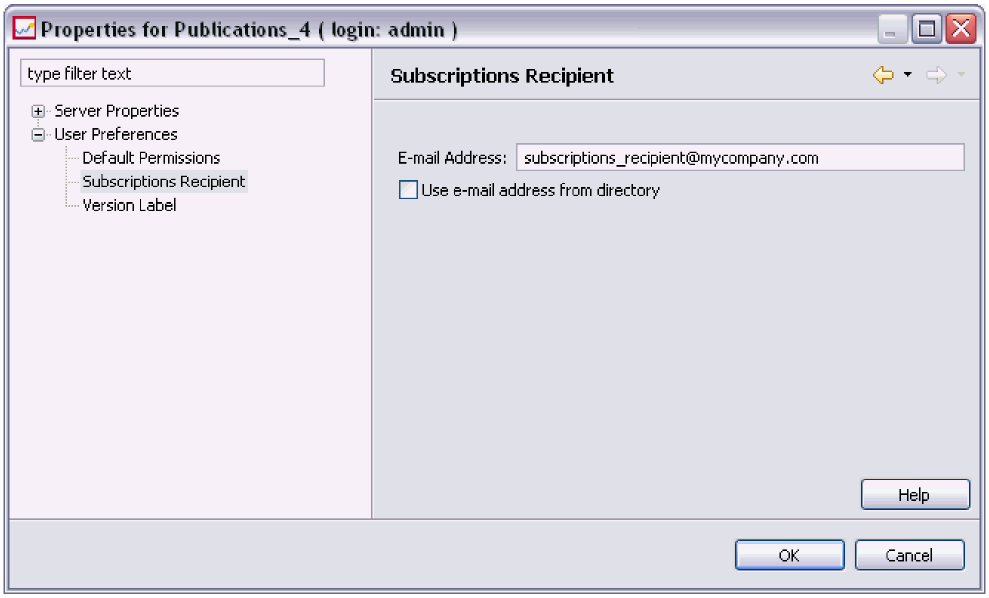
To specify an e-mail address for a subscriptions recipient:

1. Navigate to the Properties dialog box for the server instance.

2. In the Properties tree, expand the User Preferences list and select Subscriptions Recipient.

3. Select one of the following options:
   - In the E-mail Address field, type the e-mail address to use as the default.
   - Select the Use e-mail address from directory check box to use an e-mail address from the directory.

Figure 5-26
Default e-mail address



E-mail Address: subscriptions_recipient@mycompany.com

□ Use e-mail address from directory

4. Click OK.

Version Label (User Preferences)

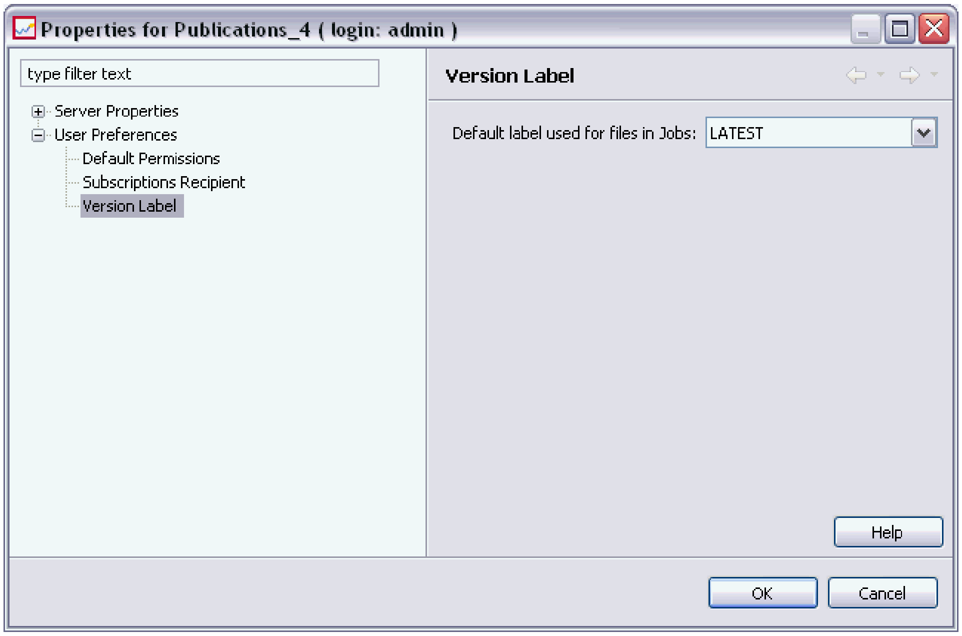
To access server version label properties:

1. In the Content Explorer, right-click on a server name and choose Properties. The Server properties dialog box opens.

2. Click + to expand the user preference options.

3. Click Version Label. The Version Label dialog box opens.

Figure 5-27
Server Version Label properties



Default label used for files in Jobs: LATEST

4. From the Default label used for files in Jobs drop-down list, select the default version label to apply to files within a job.

5. Click OK.

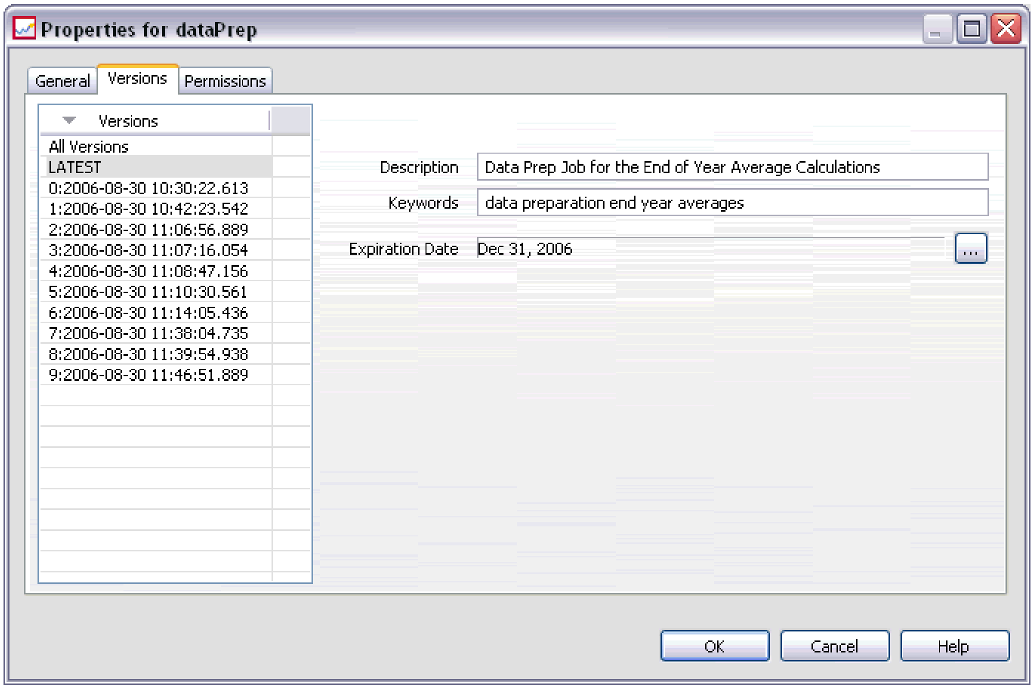
Version Properties

The repository allows you to maintain multiple versions of objects.

To access the version listing for an object:

▶ In the Properties dialog box, click the Versions tab. A list of versions for the object appears.

Figure 5-28
The Properties dialog box displaying object versions



Versions

All Versions
LATEST
0:2006-08-30 10:30:22.613
1:2006-08-30 10:42:23.542
2:2006-08-30 11:06:56.889
3:2006-08-30 11:07:16.054
4:2006-08-30 11:08:47.156
5:2006-08-30 11:10:30.561
6:2006-08-30 11:14:05.436
7:2006-08-30 11:38:04.735
8:2006-08-30 11:39:54.938
9:2006-08-30 11:46:51.889

Description: Data Prep Job for the End of Year Average Calculations
Keywords: data preparation end year averages
Expiration Date: Dec 31, 2006

The list displays the version number and the created timestamp of the version. The most recent version is labeled *LATEST*. *LATEST* is the version selected by default in the list. The list also displays the *All Versions* entry.

In addition to the general properties defined for repository objects, certain properties can also be defined on a version level.

Default version properties include:

- **Description.** A user-defined label for the version.
- **Keywords.** The metadata assigned to repository object versions for the purposes of content search.
- **Expiration Date.** The date after which repository object versions are no longer active. For more information, see the topic Working with Expiration Dates and Expired Files on p. 50.

The dialog box also displays custom version properties defined for the server. For more information, see the topic Creating Custom Properties on p. 66. The properties are displayed for a selected version in the right-hand pane of the properties dialog box. They can be set for individual versions, for multiple versions, or for all existing versions. The properties allow the user to enhance the metadata for a particular version of an object. For example, the version description property could be used to clearly state its differences from other versions, while specifying keywords makes the search more precise.

To set or modify properties for a single version:

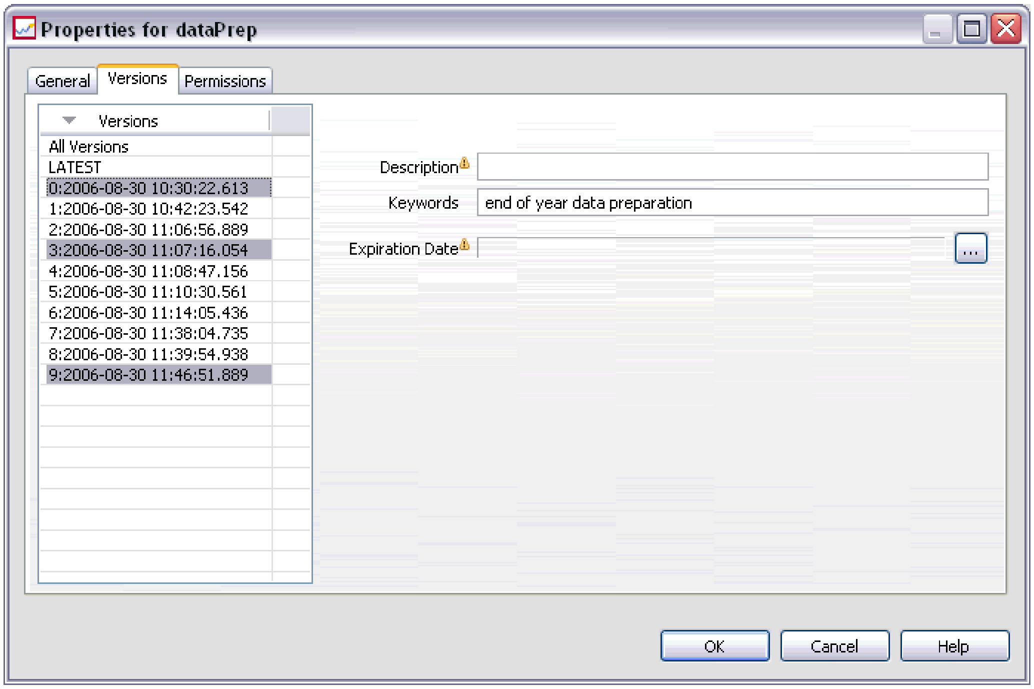
1. Select the version.

2. Modify the properties as necessary. The properties are set to the specified values for the selected version.

To set or modify properties for multiple versions:

1. Hold down the Shift key to select the versions. Any property fields that have different values between the selected versions are empty and marked by alert signs.

Figure 5-29
Properties dialog box when modifying properties for multiple versions



Versions
All Versions
LATEST
0:2006-08-30 10:30:22.613
1:2006-08-30 10:42:23.542
2:2006-08-30 11:06:56.889
3:2006-08-30 11:07:16.054
4:2006-08-30 11:08:47.156
5:2006-08-30 11:10:30.561
6:2006-08-30 11:14:05.436
7:2006-08-30 11:38:04.735
8:2006-08-30 11:39:54.938
9:2006-08-30 11:46:51.889

Description

Keywords: end of year data preparation

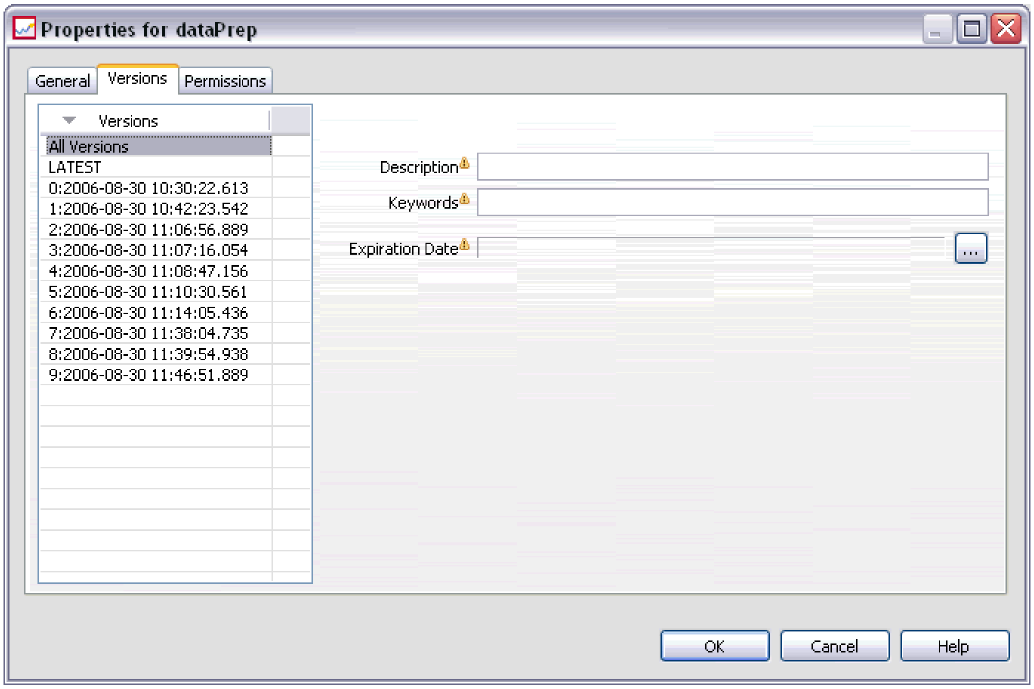
Expiration Date

2. Modify the properties as necessary. The properties are set to the specified values for the selected versions.

To set or modify properties for all versions:

1. Select *All Versions* in the list. Any property fields that have different values between the existing versions are empty and marked by alert signs.

Figure 5-30
Properties dialog box when modifying properties for all versions



2. Modify the properties as necessary. The properties are set to the specified values for all versions.

Working with Custom Properties

Custom properties consist of user-defined metadata that apply to objects within the repository. You must have the appropriate access privileges to create and modify custom properties.

Custom properties are applied per server instance. Custom properties persist across sessions and remain associated with the server instance until they are removed.

Currently, you cannot copy custom properties from one server instance to another. In addition, you cannot make any of the custom properties fields required.

The process for working with custom properties consists of the following tasks:

- Ensuring that you have access privileges to create custom properties
- Creating custom properties
- Setting values for custom properties

Typically, an administrator creates custom properties, and non-administrative users set values for custom properties.

Although custom properties are created at the server level, values for custom properties are set at the content object level. Specifically, the custom properties that you create in the Custom Properties dialog box appear in the General properties dialog box for each content object. For more information, see the topic Editing General Properties on p. 35.

Verifying Access Privileges to Create Custom Properties

Before you begin working with custom properties, you must have the appropriate access privileges to create and modify custom properties. If you log into a server and you do not have access privileges to create custom properties, the custom properties option will not be visible. A user with administrative privileges to the server instance automatically has permissions to create, edit, and delete custom properties.

Suppose that you log into a repository and you do not have the appropriate access privileges to create custom properties. If your administrator assigns you access privileges while you are still logged into the repository, you will need to log out of the repository and then log back into the repository before the updated access privileges take effect.

Accessing the Custom Properties Dialog Box

To access the Custom Properties dialog box:

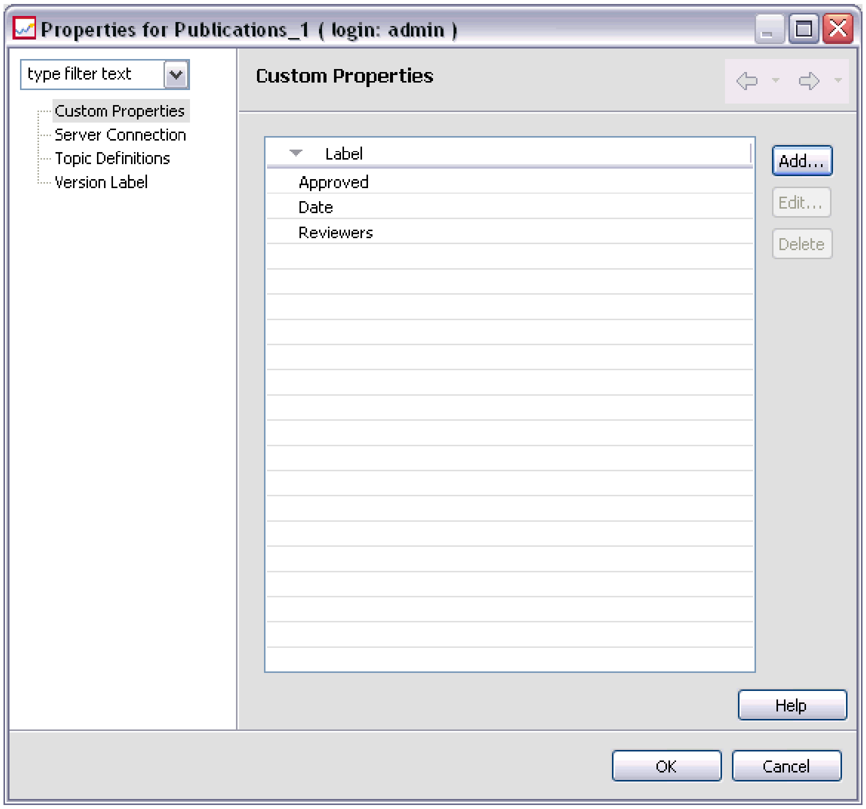
1. Log into the server for which you want to create custom properties. For more information, see the topicLogging in to a Server in Chapter 4 on p. 17.

2. Right-click on the server name and choose Properties. The Properties dialog box opens.

   Note: You must right-click on the server name to view custom properties. If you select any other object within the server folder, only general properties and permissions appear.

3. Click Custom Properties. The Custom Properties table appears. If you have previously defined custom properties, they appear in the table.

Figure 5-31
Custom Properties main dialog box



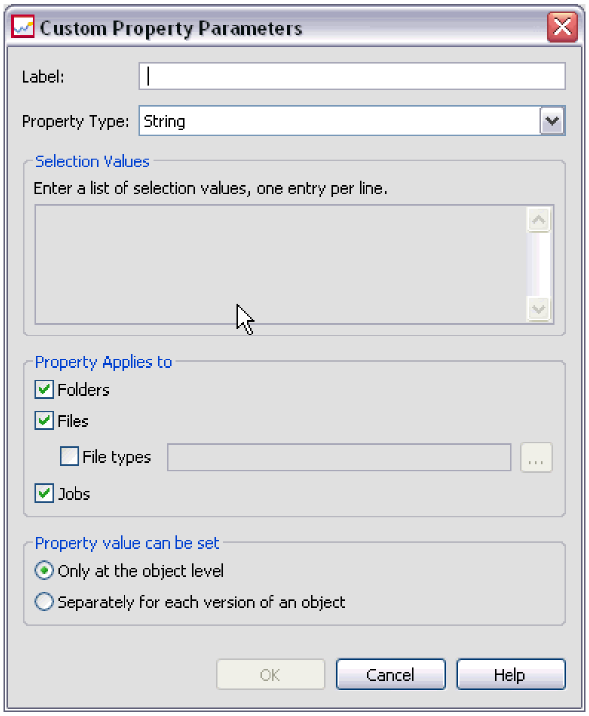
Creating Custom Properties

To create a new custom property:

1. Navigate to the Custom Properties dialog box. For more information, see the topicAccessing the Custom Properties Dialog Box on p. 65.

2. To create a new custom property, click Add. The Custom Property Parameters dialog box opens.

Figure 5-32
Custom Property Parameters dialog box



Custom Property Parameters

Label:

Property Type: String

Selection Values

Enter a list of selection values, one entry per line.

Property Applies to

☑ Folders

☑ Files

□ File types

☑ Jobs

Property value can be set

● Only at the object level

○ Separately for each version of an object

3. In the Custom Property Parameters dialog box, you need to provide the following information:

   - **Label**. The label is the name of the custom property as it will appear in the user interface. The label can be up to 128 characters in length and must be unique to the server instance. If you specify a duplicate name, the Duplicate Property dialog box opens indicating that the name is already in use. However, you may specify the same label for custom properties on two different servers.

   - **Property Type.** The property type describes the input value for the custom property. You have the following options:

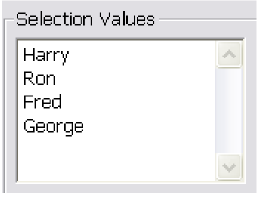
Table 5-4
Custom property types

Property type	**Description**	**Representation in the General properties dialog box**
String	Any valid string. String is the default property type.	Text field.
Number	Any numeric value.	Text field.
Single choice	A single selection from the list of options.	Drop-down list.
Multiple select	One or more selections from the list of options.	A series of check boxes.

Property type	Description	Representation in the General properties dialog box
Yes/No	Either yes or no. Selecting the check box indicates yes. A blank check box connotes no. By default, the check box is deselected. Thus, the default is no.	A single check box.
Date	A valid date in the format m/d/yyyy—for example, 5/18/2006.	A drop-down calendar that prompts the user to specify a date. Alternatively, the user can also type the date.

- **Selection values.** For custom property types that require the user to choose from a list of values (for example, single select and multiple select), you need to define the values available to the user in the Selection Values list. Items in the list are line delimited. Thus, when you add selection values, each value must appear on a new line.

Figure 5-33
Custom property selection values



Selection Values

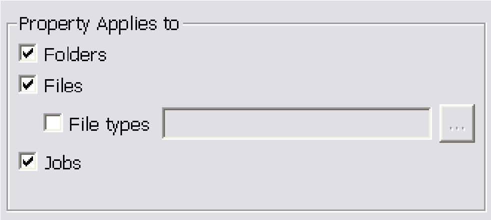
Harry
Ron
Fred
George

- **Property Applies to.** This field indicates the objects to which the custom property is applied. You must select at least one check box. You have the following options:

Property Applies to	Description
Folders	Applies to all folders in the repository.
Files	Applies to all files in the repository.
File types	Applies only to the file types you specify. If you select the File types option, the Files option is automatically selected. To specify a file type: 1. Click the ellipsis button. The File Types dialog box opens. 2. From the File Types list, select the file type you want to associate with the custom property. To select multiple file types, press the Ctrl key and select multiple rows. If you select one file type, that file type appears in the text box. If you select multiple file types, the first file type appears in the list followed by an ellipsis. 3. Click OK.
Jobs	Applies only to jobs in the repository.

By default Folders, Files, and Jobs are selected.

Figure 5-34
Custom Property Applies to default selections



Property Applies to

☑ Folders

☑ Files

□ File types

☑ Jobs

- **Property value can be set.** This field indicates whether the property can be set only at the object level or the version level.

4. To save your changes, click OK.

Editing Custom Properties

After you have created and saved a custom property, you can edit its attributes at any time. In this context, editing refers to modifying the values available to the user. Editing custom properties takes place at the server level. This process is separate from setting values for the custom property, which takes place at the content object level.

To edit a custom property:

1. Navigate to the Custom Properties dialog box. For more information, see the topicAccessing the Custom Properties Dialog Box on p. 65.

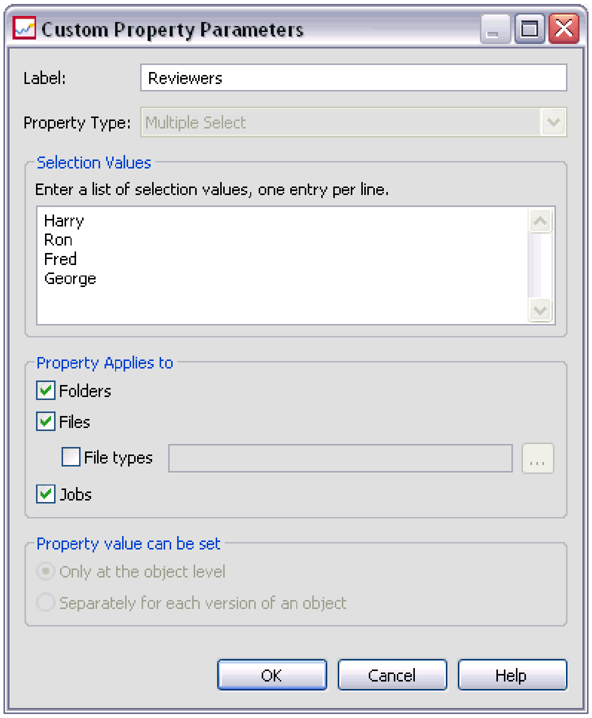
2. In the Custom Properties table, select the custom property that you want to edit.

3. Click Edit. The Custom Property Parameters dialog box opens.

4. You can modify any of the parameters in this dialog box except for the following:

   - **Property Type.** When you edit a custom property, the Property Type field is disabled. To change the property type, you must create a new custom property. For more information, see the topic Creating Custom Properties on p. 66.
   - **Property Value Can Be Set.** The level at which a custom property value is set cannot be modified. When you edit a custom property, this section of the dialog box is disabled.

Figure 5-35
Editing a custom property



Custom Property Parameters

Label: Reviewers

Property Type: Multiple Select

Selection Values

Enter a list of selection values, one entry per line.

Harry
Ron
Fred
George

Property Applies to

☑ Folders

☑ Files

□ File types

☑ Jobs

Property value can be set

● Only at the object level

○ Separately for each version of an object

5. After you have made changes, click OK to save them.

Editing Selection Values

For the single select and multiple select property types, you can modify the selection values available to users. It is important to note that if you add or delete any of the options in the Selection Values table, that change is propagated to all of the objects on the server associated with that custom property.

For example, suppose you have a multiple select property type with the following options:

- *Harry*
- *Ron*
- *Fred*
- *George*

If you remove *George* from the Selection Values list, the value *George* would be removed from any content objects to which it was previously assigned. In addition, any users that had previously selected *George* would no longer see *George* as an option.

Searching for Custom Properties

Custom properties are searchable and used in auditing reports, with the following exceptions:

- **Yes/no label values.** The label that you apply to any yes/no property types will not appear in a search.
- **Numeric values.** For any number property types, the numeric value that you type into the text field will not appear in a search.

For more information, see the topic Searching in Chapter 4 on p. 22.

Deleting Custom Properties

Custom property definitions are stored in the repository. If you delete a custom property, the custom property and all of its attributes are removed from the repository. Once a custom property is deleted, it cannot be recovered.

It is important to note that when you delete a custom property, the deletion affects all of the objects in the repository that contained that custom property. Once deleted, a custom property will no longer be applied to any object. It is strongly recommended that you do not delete custom properties, unless absolutely necessary.

For example, suppose you have a custom property called Reviewers that applies to all of the files on your server. After you delete the custom property and navigate to the Properties dialog box, the Reviewers custom property will no longer appear for any file on the server.

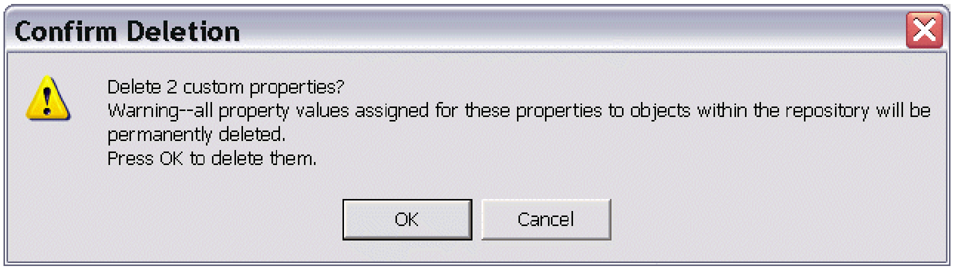
To delete a custom property:

1. Navigate to the Custom Properties dialog box. For more information, see the topicAccessing the Custom Properties Dialog Box on p. 65.

2. From the Custom Properties table, select the property that you want to delete. To select multiple custom properties, press the Ctrl key and select multiple rows.

3. Click Delete. The Confirm Deletion dialog box opens.

Figure 5-36
Custom Properties Confirm Deletion dialog box



Confirm Deletion

Delete 2 custom properties?
Warning--all property values assigned for these properties to objects within the repository will be permanently deleted.
Press OK to delete them.

4. Click OK. The Custom Properties dialog box reopens, and the Custom Properties table is refreshed.

Custom Properties and Server Connections

When you delete a server connection, the system removes the connection between the client and the server that hosts your repository. This process does not modify any items in the repository itself. Thus, deleting a server connection has no effect on the custom properties associated with that repository.

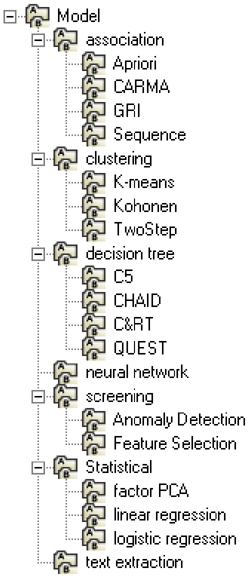
Working with Topics

Topics allow the definition of a classification system for the content stored in the repository, providing a hierarchical map to guide users to the resources they need. Topics function like a directory structure but differ from directories in that a single object can be listed under multiple topics.

For example, you might want to create a topic structure that mirrors your organization, with separate topics for marketing, finance, development, and so on. Users can then choose from the available topics when storing content. In addition, users can limit content searches to specific topics to accelerate the retrieval process. Because a given item can be listed under multiple topics, cross-indexing is also possible.

Alternatively, consider the topic hierarchy shown in the "Example topic hierarchy" figure. This hierarchy uses model type as the basis for classifying resources.

Figure 5-37
Example topic hierarchy



- Model
  - association
    - Apriori
    - CARMA
    - GRI
    - Sequence
  - clustering
    - K-means
    - Kohonen
    - TwoStep
  - decision tree
    - C5
    - CHAID
    - C&RT
    - QUEST
  - neural network
  - screening
    - Anomaly Detection
    - Feature Selection
  - Statistical
    - factor PCA
    - linear regression
    - logistic regression
  - text extraction

Any model in the repository could be assigned a topic in this hierarchy to assist in finding desired resources. For example, a user might want to find all association models that use a specific field. Alternatively, the search may be restricted to CARMA models only.

Process Overview

Topics consist of user-defined metadata that apply to objects within the repository. You must have the appropriate access privileges to create and modify topics. Working with topics consists of a two-part process.

Creating topic definitions. Topic definitions and the topic hierarchy are established in the Topic Definitions dialog box. The topics that you create in the Topic Definitions dialog box appear in the General properties dialog box for each content object. For more information, see the topic Editing General Properties on p. 35.

The following guidelines apply:

- Topic definitions are applied per server instance
- Topic definitions persist across sessions and remain associated with the server instance until they are removed
- Topic definitions cannot be copied from one server instance to another
- Topics that you define cannot be made into required fields

Assigning topic values. Although topic definitions are created at the server level, values for topics are set at the content object level. It is important to note that topics are assigned at the object level only. Topics cannot be assigned to folders. In addition, topics cannot be assigned to any items in the *Resource Definitions* folder of the Content Explorer.

Typically, an administrator creates topic definitions, and non-administrative users assign topics.

Verifying Access Privileges to Create Topic Definitions

Before you begin working with topic definitions, you must have the appropriate access privileges to create and modify topic definitions. If you log into a server and you do not have access privileges to create topic definitions, the topic definition option will not be visible. A user with administrative privileges to the server instance automatically has permissions to create, edit, and delete topic definitions.

Suppose that you log into a repository and you do not have the appropriate access privileges to create topic definitions. If your administrator assigns you access privileges while you are still logged into the repository, you will need to log out of the repository and then log back into the repository before the updated access privileges take effect.

Working with Topic Definitions

The following tasks can be performed in the Topic Definitions dialog box:

- Create new topic definitions
- Rename topic definitions
- Move topic definitions within the topic hierarchy
- Delete topic definitions

Accessing the Topic Definitions Dialog Box

To access the Topic Definitions dialog box:

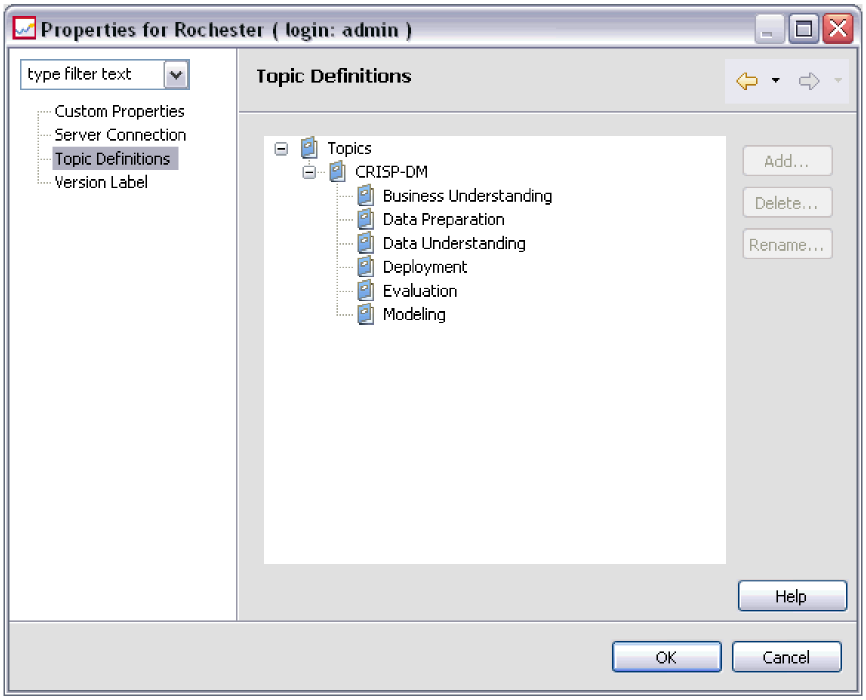
1. Log into the server for which you want to create topics. For more information, see the topicLogging in to a Server in Chapter 4 on p. 17.

2. Right-click on the server name and choose Properties. The Properties dialog box opens.

   Note: You must right-click on the server name to view topics. If you select any other object within the server folder, only general properties, permissions, and versions appear.

3. Click Topic Definitions. The topics hierarchy appears. If you have previously defined topics, they appear in the hierarchy. Otherwise, only the main *Topics* folder appears.

Figure 5-38
Topic Definitions dialog box



Creating New Topic Definitions

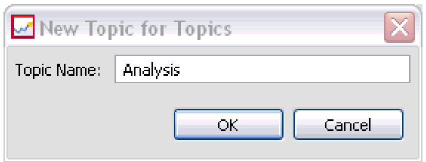
To create a new topic definition:

1. Navigate to the Topic Definitions dialog box. For more information, see the topicAccessing the Topic Definitions Dialog Box on p. 74.

2. Select the folder of the topic hierarchy in which to create the new topic definition.

3. To create a new topic definition, click Add. The New Topic dialog box opens.

Figure 5-39
New Topic dialog box



Topic Name: Analysis

4. In the Topic Name field, type a name for your topic.

5. Click OK. The Topic Definitions dialog box reopens and the topic hierarchy is refreshed.

Renaming Topic Definitions

Topic definitions can be renamed. After a topic definition is renamed, the change is propagated to all content objects to which the topic is assigned. Any topic definition can be renamed. However, the root *Topics* directory cannot be renamed. The Rename button is disabled when you select the *Topics* folder.

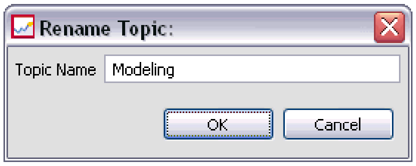
To rename a topic definition:

1. Navigate to the Topic Definitions dialog box. For more information, see the topicAccessing the Topic Definitions Dialog Box on p. 74.

2. From the topic hierarchy, select the topic you want to rename.

3. Click Rename. The Rename Topic dialog box opens.

Figure 5-40
Rename Topic dialog box



Topic Name Modeling

4. In the Topic Name field, type a name for your topic.

5. Click OK. The Topic Definitions dialog box reopens, and the topic hierarchy is refreshed.

Moving Topic Definitions

Topic definitions can be moved by cutting the topic definition and pasting it to a new location within the topic hierarchy. It is important to note that when a topic is cut from the topic hierarchy, all of the children of that topic are cut as well.

Any topic definition can be moved. The only exception is the root *Topics* directory. This directory cannot be moved. The Cut and Paste options are disabled when you select the *Topics* folder.

To move a topic definition:

1. Navigate to the Topic Definitions dialog box. For more information, see the topicAccessing the Topic Definitions Dialog Box on p. 74.

2. In the topic hierarchy, right-click on the topic you want to move and choose Cut.

3. Right-click on the folder into which you want to paste the topic definition and choose Paste. The topic definition is moved into its new location.

Deleting Topic Definitions

Topic definitions are stored in the repository. If you delete a topic, the topic and all of its attributes are removed from the repository. Once a topic definition is deleted, it cannot be recovered.

For example, suppose you have a topic definition called *Analysis*. After you delete the topic definition and navigate to the Properties dialog box, the *Analysis* topic will no longer appear for any content object on the server.

It is important to note the following about deleting topic definitions:

- When you delete a topic definition, the deletion affects all of the objects in the repository that contained that topic. Once deleted, a topic definition will no longer be applied to any object.
- When you delete a topic definition, all of the children of that topic definition are deleted as well.
- You cannot delete the root *Topics* directory. The Delete button is disabled when you select the *Topics* folder.

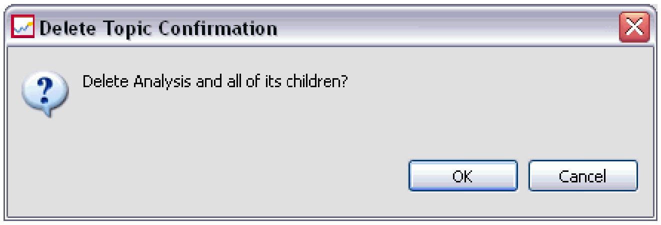
To delete a topic definition:

1. Navigate to the Topic Definitions dialog box. For more information, see the topicAccessing the Topic Definitions Dialog Box on p. 74.

2. From the topic hierarchy, select the topic that you want to delete.

3. Click Delete. The Delete Topic Confirmation dialog box opens.

Figure 5-41
Delete Topic Confirmation dialog box



Delete Analysis and all of its children?

4. Click OK. The Topic Definitions dialog box reopens, and the topic hierarchy is refreshed.

The Effect of Deleting Server Connections on Topics

When you delete a server connection, the system removes the connection between the client and the server that hosts your repository. This process does not modify any items in the repository itself. Thus, deleting a server connection has no effect on the topic definitions or assignments associated with that repository.

Deleting Topic Definitions vs. Removing Topic Assignments

Deleting a topic definition is different from removing a topic assignment. Deleting a topic definition removes the definition from the system, preventing any content object from being associated with that topic definition. Thus, deleting a topic definition affects all content objects associated with that topic definition.

Removing a topic assignment severs the link between a content object and the topic property. However, removing a topic assignment from a content object does not affect any other content objects that were assigned that topic. For more information, see the topic Removing Topics on p. 41.

Searching for Topics

Topics are searchable and used in auditing reports. For more information, see the topic Searching in Chapter 4 on p. 22.

Bulk Property Updates

You can update properties for multiple objects simultaneously (in bulk). The properties that can be updated in bulk include:

- Description
- Keywords
- Topics
- Permissions
- Expiration date
- Custom properties
- Content language

To update properties in bulk:

1. In the Content Explorer, select the objects. Selected objects can include files, folders, and jobs.

2. Right-click and choose Properties. The Shared Properties dialog box opens. The dialog box title bar indicates how many objects have been selected.

3. Modify properties on the General, Versions, and Permissions tabs as necessary. For more information see, Updating General Properties in Bulk on p. 78, Updating Version Properties in Bulk on p. 80, and Updating Permissions in Bulk on p. 81

Note: The Shared Properties dialog box does not display the Versions tab if any folders are selected for the bulk update.

4. Click OK.

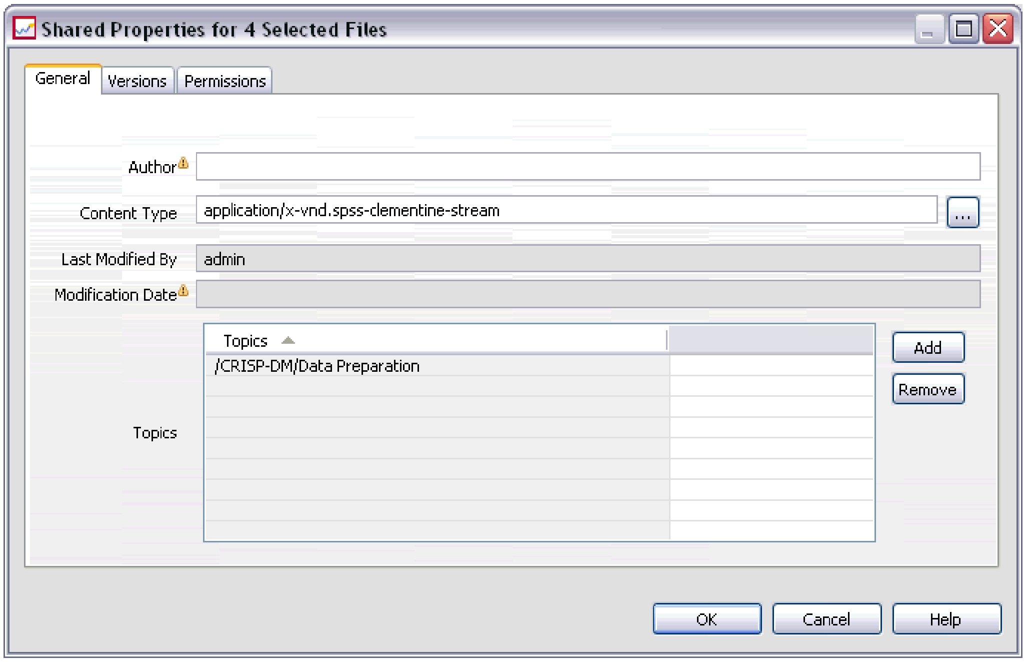
Updating General Properties in Bulk

To access the general properties for the selected objects:

1. In the Shared Properties dialog box, click General.

- If selected objects are files or jobs, Author, Content Type, Last Modified By, and Modification Date fields are displayed. The dialog box also displays object-level custom properties defined in the repository and the topics common to all objects.

Figure 5-42
Updating general properties in bulk for files and jobs



Shared Properties for 4 Selected Files

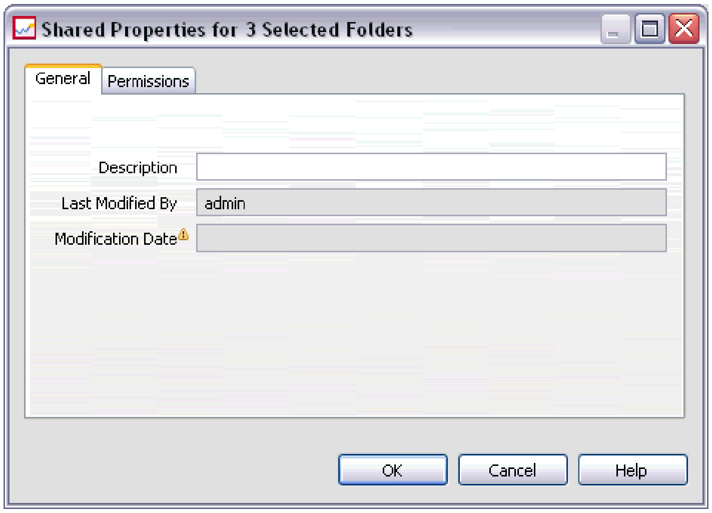
Content Type: application/x-vnd.spss-clementine-stream

Last Modified By: admin

Topics
/CRISP-DM/Data Preparation

- If selected objects are folders, only Description, Last Modified By, and Modification Date fields are displayed. The dialog box also displays all custom properties defined for the repository and the topics common to all objects.

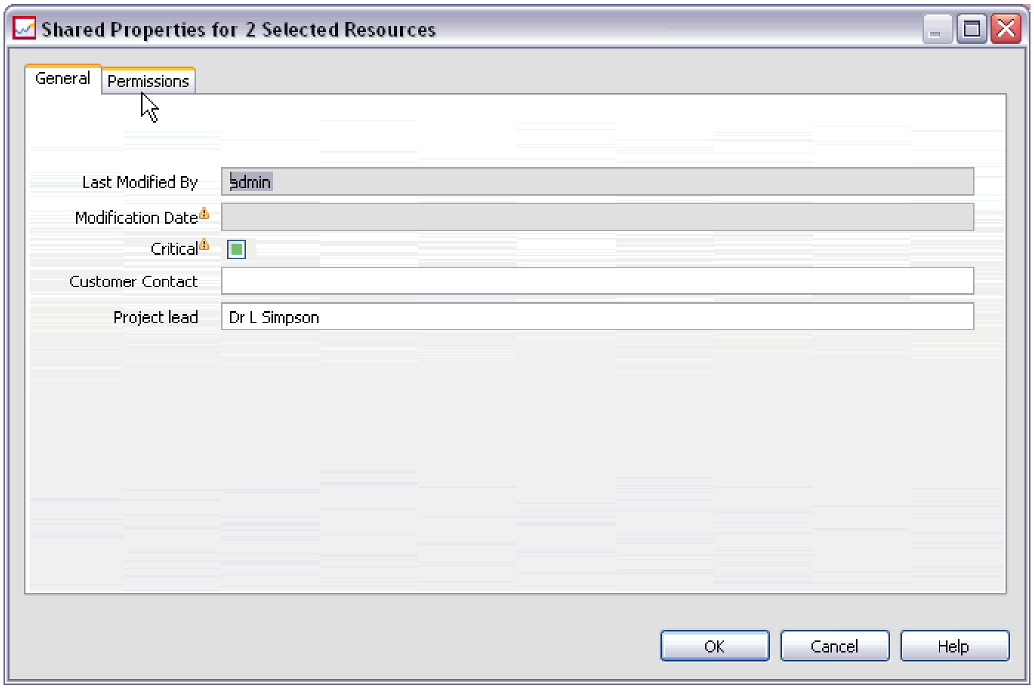
Figure 5-43
Updating general properties in bulk for folders


Shared Properties for 3 Selected Folders

Description
Last Modified By admin
Modification Date

- If selected objects include both files or jobs and folders, only Last Modified By and Modification Date fields are displayed. The dialog box also displays object-level custom properties defined for the repository. The dialog box does not display topics.

Figure 5-44
Updating general properties in bulk for files or jobs and folders


Shared Properties for 2 Selected Resources

Last Modified By admin
Modification Date
Critical 🟩
Customer Contact
Project lead Dr L Simpson

- Any property fields that have different values between the selected objects are empty and marked by alert signs. Green boxes are displayed for check box custom properties when the state of the check box is not common for all selected objects.

2. Modify the property values as necessary. To change the content type of the objects, click the ellipsis button to open the File Types dialog box and select the type. Click Add to open the Add Topic dialog box and select topics. To remove a topic, select the topic in the list and click Remove. For more information, see the topicWorking with Topics on p. 72.

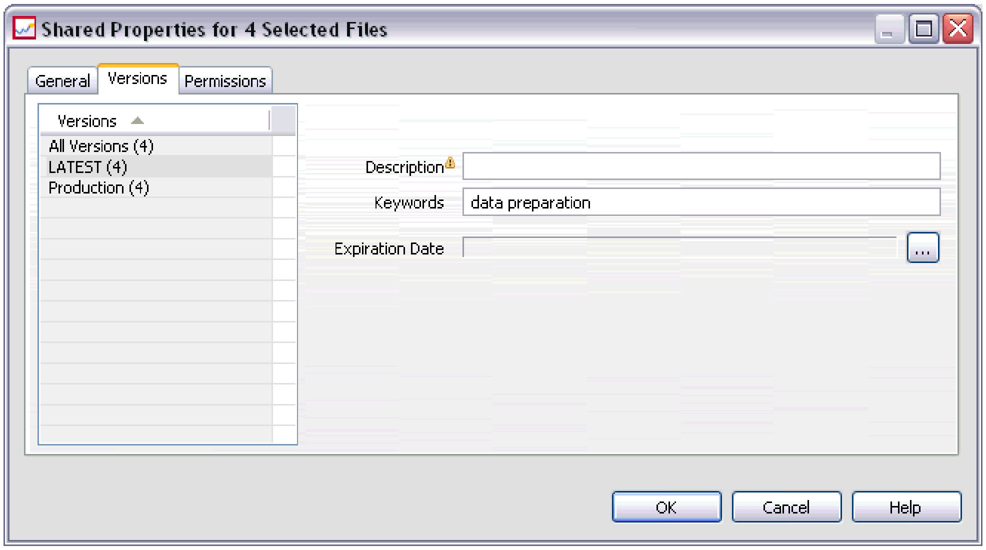
   Note: If there are jobs among selected objects, the Content Type field is read-only.

Updating Version Properties in Bulk

To access the version properties for the selected objects:

▶ In the Shared Properties dialog box, click Versions.

Figure 5-45
Updating version properties in bulk



Shared Properties for 4 Selected Files

Description
Keywords: data preparation
Expiration Date

- **Description.** A user-defined label for the version.
- **Keywords.** The metadata assigned to repository object versions for the purposes of content search.
- **Expiration Date.** The date after which repository object versions are no longer active. For more information, see the topic Working with Expiration Dates and Expired Files on p. 50.

The dialog box also displays custom version properties defined for the server. For more information, see the topic Creating Custom Properties on p. 66. The properties are displayed either for the latest versions of the selected objects or for all versions. Any property fields that have different values between the selected objects are empty and marked by alert signs. Green boxes are displayed for check box custom properties when the state of the check box is not common for all selected objects.

To set or modify properties for the latest versions of selected objects:

1. Select *LATEST* in the list.
2. Modify the properties as necessary.

To set or modify properties for all versions:

1. Select *All Versions* in the list.

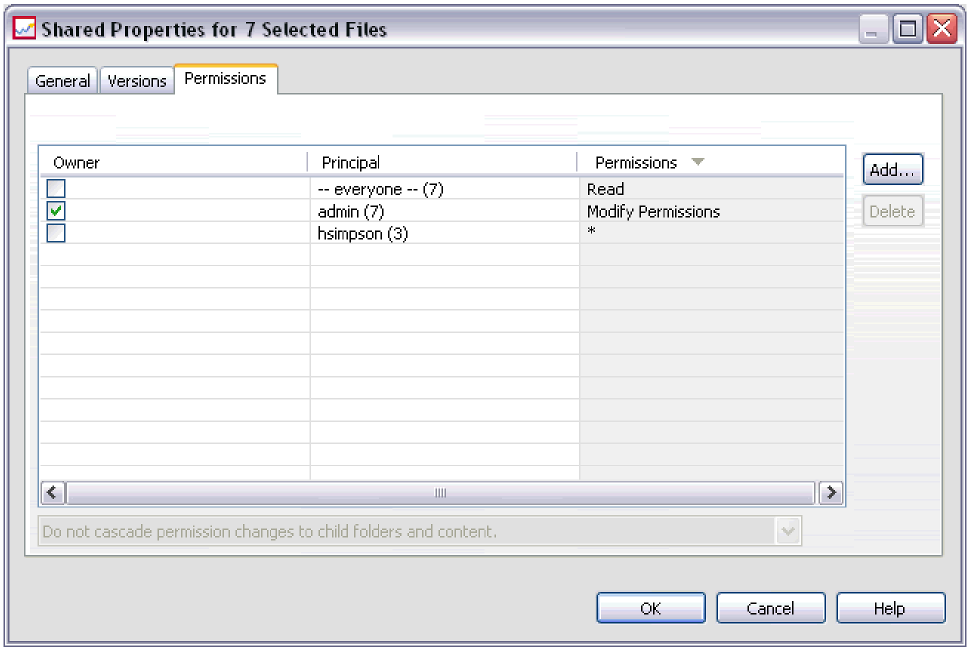
2. Modify the properties as necessary.

Updating Permissions in Bulk

To access the permissions for the selected objects:

▶ In the Shared Properties dialog box, click Permissions.

Figure 5-46
Updating permissions in bulk



Owner	Principal	Permissions
□	-- everyone -- (7)	Read
☑	admin (7)	Modify Permissions
□	hsimpson (3)	*

Do not cascade permission changes to child folders and content.

Owner. The users who created any of the selected objects are marked with a green box in the Owner column.

Principal. The user ID and the number of the selected objects for which the users have permissions defined. As shown in the example, user *hsimpson* has explicitly defined permissions to 3 of the 7 selected objects.

Permissions. The Permissions column displays the user's permissions for the selected objects. If the user is not defined in the permissions of all objects, an asterisk (*) is displayed, as in the example above. To set users permissions to the objects, click in the column and select a permissions level from the list.

Add. Add users to the list. For more information, see the topic Adding New Users or Groups on p. 42.

Delete. Delete users from the list. For more information, see the topic Deleting a User or Group from the Permissions List on p. 46.

Chapter

6

Resource Definitions

In the Content Explorer, the *Resource Definitions* folder contains credential definitions, data source definitions, message domains, server definitions, and server cluster specifications. These resources are often required for running jobs. For example, a PASW Modeler job step requires the definition of a PASW Modeler execution server to process the step.

To create new resource definitions, you must have the appropriate actions associated with your login credentials. In addition, you need *Write* permission to the folder that will contain the new definition. Contact your PASW Collaboration and Deployment Services administrator if your actions or permissions require modifications.

You cannot save any other items into the *Resource Definitions* folder. For example, jobs cannot be saved here. If you try to save or move an item into the *Resource Definitions* folder, an error message appears.

Adding New Credentials

Execution of some job steps requires the specification of credentials, allowing credential validation before executing a job. If needed, credentials can be stored within a domain. Write permissions to the Credentials folder are required to create and modify data credentials. For more information, see the topic Modifying Permissions for an Existing User or Group in Chapter 5 on p. 44. Also, the user must be assigned the *Define Credentials* action through the role.

To add a new credential:

1. In the Content Explorer, open the *Resource Definitions* folder.

2. Open the *Credential Definitions* folder.

3. Select the domain in which you want to create a new credential definition.

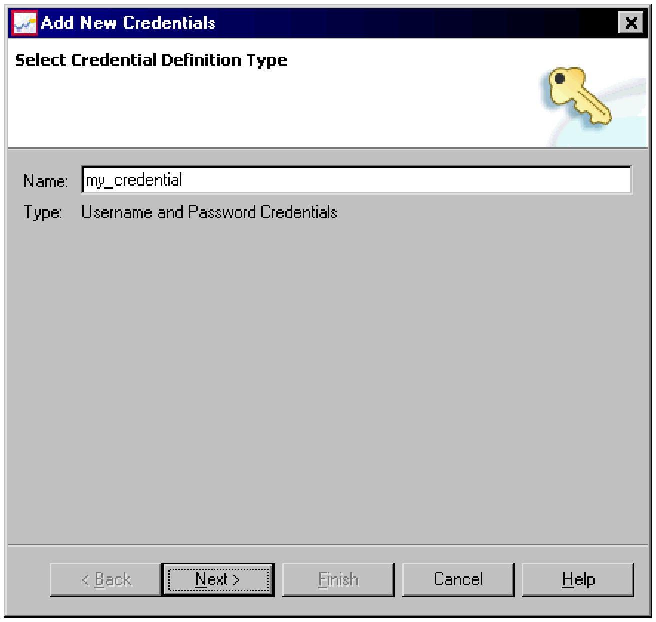
4. From the File menu, choose:

   New
     Credentials Definition

   The Add New Credentials wizard opens.

   Note: Alternatively, the new credentials dialog box can be accessed by clicking New next to the login field on the General tab for certain steps (for example, PASW Modeler job steps).

Figure 6-1
Adding a new credential



Name: my_credential

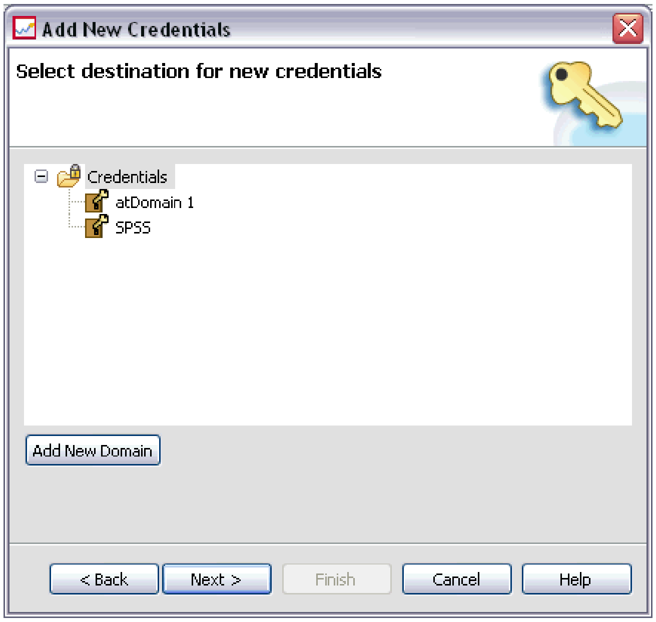
Type: Username and Password Credentials

5. In the Name field, enter a nickname for the credential. The credential destination dialog box opens.

Credential Destination

After you have specified the name, the credential destination dialog box opens.

Figure 6-2
Credentials destination



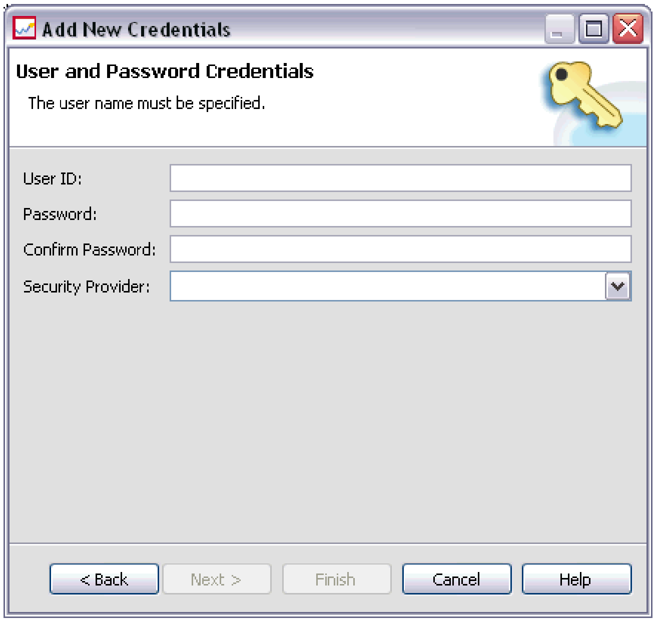
1. Navigate to the desired folder, and select it.

2. Click Next. The user and password credentials dialog box opens.

Specifying a User and Password for Credentials

After you have specified a name and destination for your credential and clicked Next, the User and password credentials dialog box opens.

Figure 6-3
Specifying the username and password



Add New Credentials

User and Password Credentials

The user name must be specified.

User ID:

Password:

Confirm Password:

Security Provider:

1. In the User ID field, enter the username for the credential.

2. In the Password field, enter the password that corresponds to the username.

3. In the Confirm password field, reenter the password.

4. If the credential must be validated against a security provider, select the provider using the Security Provider drop-down list. Leave this field blank if provider validation is not necessary for the credential.

5. Click Finish. The updated information appears in the *Credentials* folder.

Creating a New Domain

To create a new domain for your credentials:

1. In the Content Explorer, navigate to the *Resource Definitions* folder.

2. Right-click on the *Credential Definitions* folder and select:

   New
   Domain

3. A new domain appears in the *Credential Definitions* folder, called New Domain.

4. Assign a name to the new domain.

Working with Data Source Definitions

Open database connectivity (ODBC) provides a mechanism for client programs to access databases or data sources. Similarly, Java database connectivity (JDBC) determines how Java applications access databases.

Data source definitions allow other components of the system, such as Enterprise View, to access the data sources used in the system. For more information, see the topic Enterprise View Overview in Chapter 8 on p. 123.

Data source definitions are created in the Content Explorer and stored in the repository. The appropriate permissions are required to create and modify data source definitions. If a user does not have permissions, the *Data Source Definitions* folder is not accessible. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41. The process for adding a data source definition consists of the following steps:

- Selecting a data source definition type.
- Specifying data source definition parameters.

Data source definitions can be exported and imported.

Selecting a Data Source Definition Type

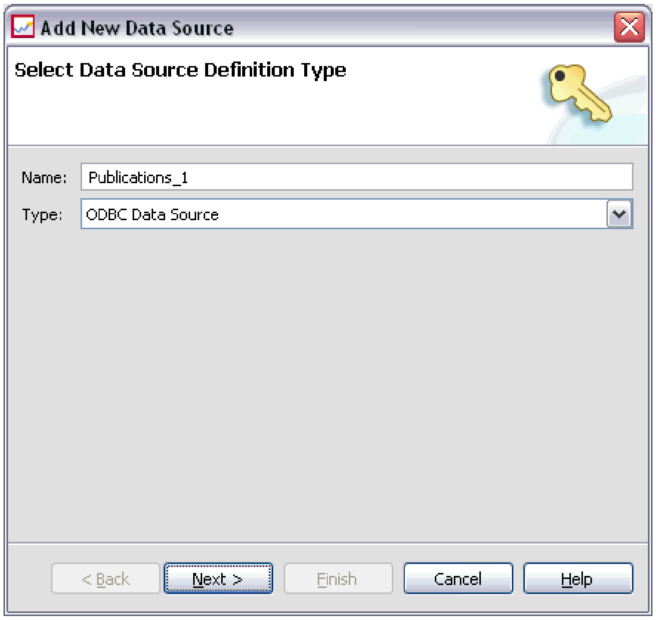
To create a new data source definition:

1. In the Content Explorer, expand the *Resource Definitions* folder. Right-click on the *Data Source Definitions* folder and select:

   New
   Data Source Definition

   The Select Data Source Definition Type dialog box opens.

Figure 6-4
Select Data Source Definition Type dialog box



Select Data Source Definition Type

Name: Publications_1

Type: ODBC Data Source

2. In the Name field, enter the name for your data source definition. This name, which will appear in the *Data Source Definitions* folder, is used when other components of the system, such as Enterprise View, invoke the data source definition.

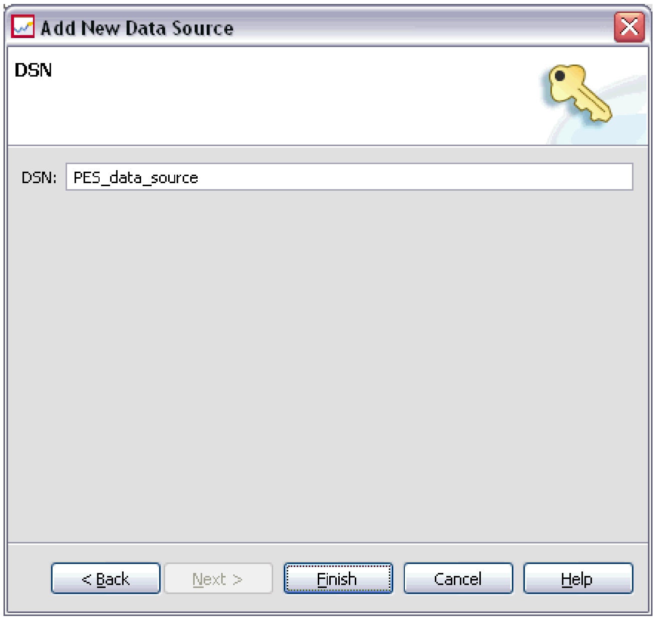
3. From the Type drop-down list, select the data source definition type. Valid values depend on the system configuration and may include *ODBC Data Source*, *JDBC Data Source*, *App Server Data Source*, and *Data Service Data Source*.

4. Click Next. The dialog box that appears depends on the data source type that was selected.
   - **ODBC.** The DSN dialog box opens.
   - **JDBC.** The JDBC Name and URL dialog box opens.
   - **App Server Data Source.** The JNDI Name dialog opens
   - **Data Service Data Source.** The Data Service Data Source Properties dialog opens.

Specifying a DSN for an ODBC Data Source

The data source name (DSN), which corresponds to the database associated with the data source, is specified in the DSN dialog box.

Figure 6-5
Specify DSN dialog box



DSN: PES_data_source

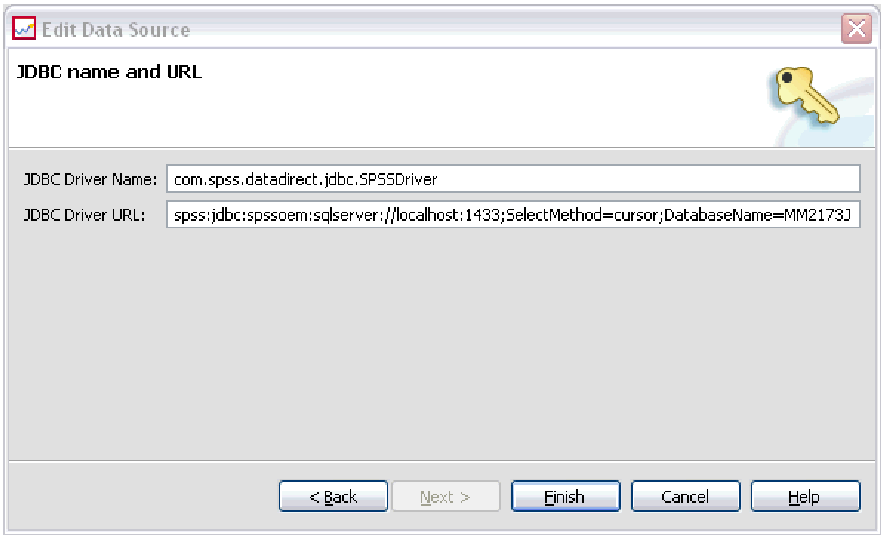
1. In the DSN field, enter the data source name. The name that you enter in this field must match your data source name exactly. The name is also case sensitive.

2. Click Finish. The new data source definition appears in the *Data Sources* folder.

Specifying a JDBC Name and URL

The JDBC name and URL, which define how Java applications connect to a database, are specified in the JDBC Name and URL dialog box.

Figure 6-6
Specify JDBC Name and URL dialog box



JDBC Driver Name: com.spss.datadirect.jdbc.SPSSDriver

JDBC Driver URL: spss:jdbc:spssoem:sqlserver://localhost:1433;SelectMethod=cursor;DatabaseName=MM2173J

1. In the JDBC Driver Name field, enter the class name of your JDBC driver.
2. In the JDBC Driver URL field, enter JDBC driver URL.
3. Click Finish. The new data source definition appears in the *Data Sources* folder.

PASW Collaboration and Deployment Services is installed with a set of JDBC drivers for all major database systems: DB2, Informix, Oracle, Microsoft SQL Server, MySQL, and Sybase. It also provides JDBC drivers for PASW Statistics file access. A special IBM i driver is provided for ShowCase users.

If single sign-on is enabled for your system, include the optional AuthenticationMethod property in the driver URL string to use single sign-on for the database connection. The only supported single sign-on method is *kerberos*. The NT LAN Manager (NTLM) authentication method is not supported.

DB2

Version: 9.5; 9.1; UDB V5R2, V5R3, V5R4 for IBM i

Class name: *spssoem.jdbc.db2.DB2Driver*

URL string template:

```
jdbc:spssoem:db2://<host>:<port>;DatabaseName=<database>[;AuthenticationMethod=kerberos]
```

Informix

Version: 11.5; 11.0; 10; 9.4; 9.3; 9.2

Class name: *spssoem.jdbc.informix.InformixDriver*

URL string template:

```
jdbc:spssoem:informix://<host>:<port>;InformixServer=<Informix server>
```

MS SQL Server

Version: 2008; 2005; 2000

Class name: *spssoem.jdbc.sqlserver.SQLServerDriver*

URL string template:

```
jdbc:spssoem:sqlserver://<host>:<port>;DatabaseName=<database>[;AuthenticationMethod=kerberos]
```

MySQL

Version: 5.0.x (commercially licensed)

Class name: *spssoem.jdbc.mysql.MySQLDriver*

URL string template:

```
jdbc:spssoem:mysql://<host>:<port>;DatabaseName=<database>
```

Oracle

Version: 11.1; 10.2; 10.1; 9.2; 9.1; 8.1.7; 8.1.6

Class name: *spssoem.jdbc.oracle.OracleDriver*

URL string template:

```
jdbc:spssoem:oracle://<host>:<port>;SID=<database>[;AuthenticationMethod=kerberos]
```

Sybase

Version: 15.0; 12.5.4; 12.5.3; 12.5.2; 12.5.1; 12.5; 12.0; 11.9; 11.5

Class name: *spssoem.jdbc.sybase.SybaseDriver)*

URL string template:

```
jdbc:spssoem:sybase://<host>:<port>;databaseName=<database>[;AuthenticationMethod=kerberos]
```

ShowCase IBM i

Class name: *com.showcasecorp.jdbc.SCDriver*

URL string template:

```
jdbc:showcasecorp://<host>:<port>/<IASPname>;translate binary=true
```

<IASPname> is the name of the Independent Auxiliary Storage Pool (IASP).

If defining a JDEdwards (JDE) World/OneWorld IBM i data source in PASW BIRT Report Designer, the enableDialogSupport parameter must be specified to display an additional dialog which lists the JDE environments if using a JDE multi-library list user profile.

URL string template:

```
spss:jdbc:showcasecorp://servername:port;translate binary=true;showcasecorp.jdbc.enableDialogSupport=true
```

PASW Statistics Data Files

Class name: *com.spss.statistics.datafile.jdbc.openaccess.OpenAccessDriver*

URL String template:

```
jdbc:paswstatistics://<hostname>:<port>;ServerDatasource=SAVDB;
CustomProperties=(<data_file_identifier>;UserMissingIsNull=<1|0>;MissingDoubleValueAsNAN=<1|0>)
```

Notes:

- The *hostname* parameter is the host name or IP address of the computer on which the driver is running. The default value is localhost.
- The default is port is 18886.
- Replace <*data_file_identifier*> with a string identifying the data file. This may be specified in a variety of ways. Use CONNECT_STRING=<path_to_file> to specify the path to the file, where the path is either a repository URI or a path relative to the host on which the driver is running. The path cannot contain any equals sign or semicolon characters. Alternatively, for data files stored in the repository, use REPOSITORY_ID=<file_id>;REPOSITORY_VERSION=<file_version> to specify the repository file and version identifiers. The file version may be specified as a label or as a version marker, with spaces replaced by the standard escape character of %20.
- The *UserMissingIsNull* parameter is optional and specifies the treatment of user-defined missing values. 0 indicates that user-defined missing values are read as valid values. 1 indicates user-defined missing values are set to system-missing for numeric variables and blank for string variables. If *UserMissingIsNull* is not specified, it is set to a default value of 1.
- The *MissingDoubleValueAsNAN* parameter is optional and specifies the treatment of missing numeric values. 0 indicates that user missing values are displayed with the original missing value in the data file. 1 indicates that user missing values are read as not a number (NaN). For JDBC, *UserMissingIsNull* should always be set to 1.

Third-Party JDBC Drivers

If PASW Collaboration and Deployment Services does not include a driver for a needed database, you can update your environment to include a third-party driver for the database. For example, if access to a Netezza or Teradata database is needed, obtain the appropriate driver from the vendor and update your system. To add a JDBC driver to Deployment Manager:

1. Close Deployment Manager if it is running.

2. Create a folder named *JDBC* at the root level of the Deployment Manager installation.

3. Place the driver files in the *JDBC* folder.

Note that for Netezza, the version 5.0 driver should be used to access both version 4.5 and 5.0 databases.

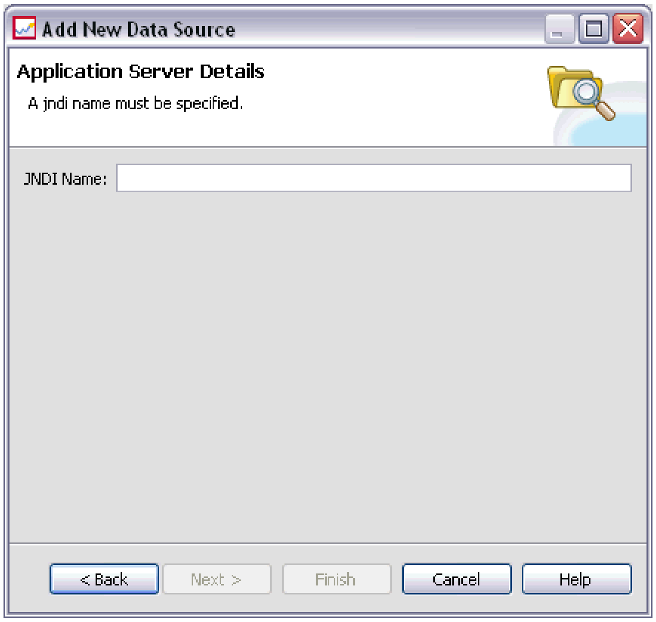
After adding the driver files to your environment, the driver can be used in a data source definition. In the JDBC Name and URL dialog box, type the name and URL for the driver. Consult the vendor documentation for the driver to obtain the correct class name and URL format.

If the data source will be used by the repository server, such as in scheduled jobs or reports, the server must also be updated to include the driver files. Contact your administrator to perform the necessary modifications.

Specifying a JNDI Name for an Application Server Data Source

The Java Naming and Directory Interface (JNDI) name is specified in the JNDI dialog box.

Figure 6-7
Specify JNDI name dialog box



Add New Data Source

Application Server Details

A jndi name must be specified.

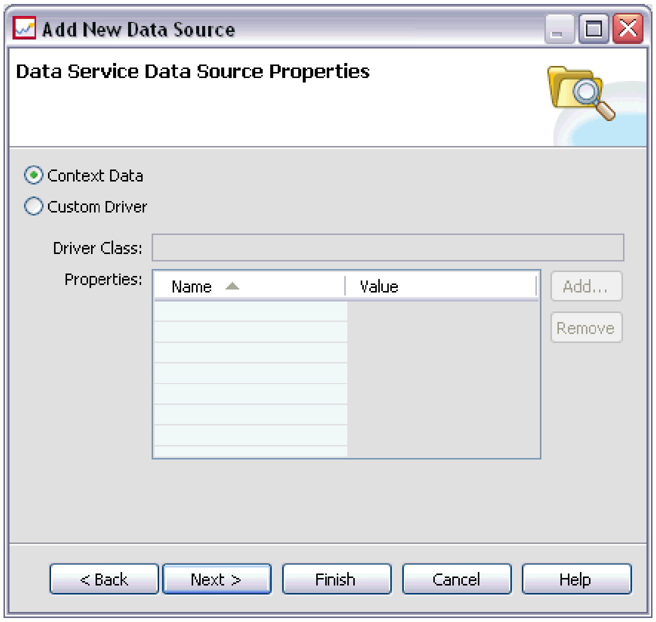
JNDI Name:

1. In the JNDI Name field, enter the JNDI name. The name that you enter in this field must match your data source name exactly. The name is also case sensitive.
2. Click Finish. The new data source definition appears in the *Data Sources* folder.

Specifying Properties for Data Service Data Sources

Properties for data service data sources are specified in the Data Service Data Source Properties dialog.

Figure 6-8
Data Service Data Source Properties dialog



Add New Data Source

Data Service Data Source Properties

● Context Data
○ Custom Driver

Driver Class:

Properties:

Name	Value

To specify data service data source properties:

1. Specify whether to use context data or a custom driver.

   Context data. If the context data option is selected, tables can also be defined.

   Custom driver. If the custom driver option is selected, a driver class must be provided. Additional driver properties can also be added.

2. Optionally, tables can be defined for both the context data and custom driver options. For more information, see the topicDefining Tables for Data Service Data Sources on p. 92.

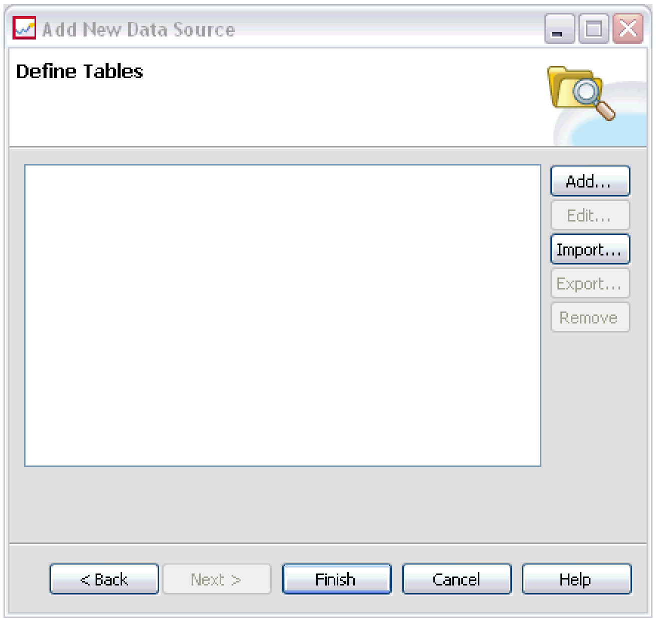
3. Click Finish. The new data source definition appears in the *Data Sources* folder.

Defining Tables for Data Service Data Sources

Optionally, tables can be defined or imported for both the context data and custom driver options. To define tables:

1. After the context data or custom driver parameters have been specified, click Next. The Define Tables dialog appears.

Figure 6-9
Defining tables for a data service data source



Add. Tables can be created in this dialog. Columns of various types can be created—for example, *Boolean*, *Date*, *Decimal*, *String*, and *Timestamp*.

Import. Tables can be imported from other source files.

2. After table information has been provided, click OK.

3. Click Finish. The new data source definition appears in the *Data Sources* folder.

Modifying Data Source Definitions

After a data source definition has been established, definition properties can be modified. For example, a JDBC driver URL may need to be updated.

Although data source parameters can be modified, the data source definition type cannot be changed. An ODBC data source cannot be converted into a JDBC data source, and vice versa. To change the data source definition type, a new data source definition must be created.

To modify the properties of an existing data source definition:

1. Navigate to the *Data Source Definitions* folder.

2. Double-click on the data source definition to be modified. The dialog box that opens depends on the data source definition type.

   - **ODBC.** The DSN dialog box opens.
   - **JDBC.** The JDBC Name and URL dialog box opens.

3. Make the desired modifications.

4. Click Finish. The changes are applied to the data source definition.

Message Domains

Message domains define JMS (Java Messaging Service) queues or topics used by PASW Collaboration and Deployment Services to communicate with third-party applications. Messages domains are used to set up message-based schedules and message-based job steps for PASW Collaboration and Deployment Services jobs. Message domain definitions reference the messaging service which is set up outside PASW Collaboration and Deployment Services, usually using the JMS infrastructure of the underlying application server.

Write permissions to the Message Domains folder are required to create and modify data credentials. For more information, see the topic Modifying Permissions for an Existing User or Group in Chapter 5 on p. 44. Also, the user must be assigned the *Define Message Domains* action through the role.

Deployment Manager provides the ability to create new message domains and modify and delete domain definitions.

Creating a New Message Domain

To add a message domain definition:

1. In the Content Explorer, open the *Resource Definitions* folder.

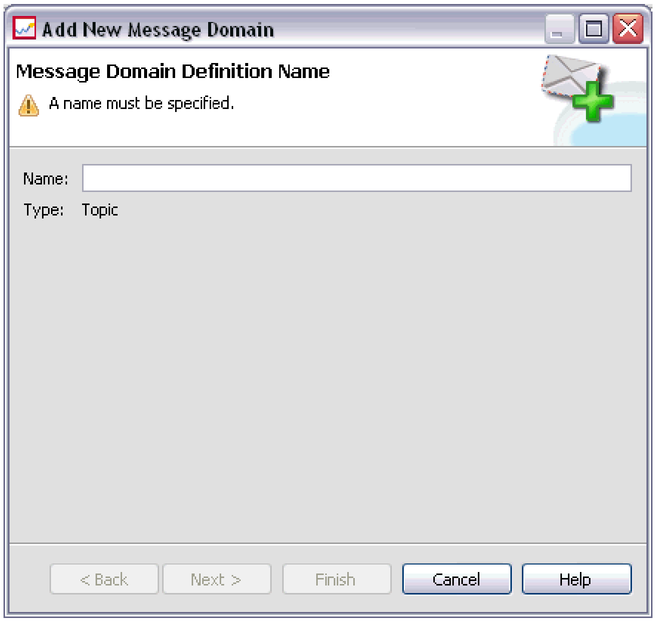
2. Open the *Message Domains* folder.

3. From the File menu, choose:

   New
     Message Domain

   The Add New Message Domain wizard opens.

Figure 6-10
Add New Message Domain wizard



The wizard provides the capability to specify new message domain name and JMS properties.

Message Domain Definition Name

The first step of adding a new message domain involves the definition of the domain name.

1. In the Name field, enter the name for the message domain.

   Note: Currently the only supported message domain type is topic.

2. Click Next. The Message Domain Page of the wizard opens.

Message Domain Properties

1. On Message Domain Properties dialog, specify the following JMS properties:
   - **Destination Name.** The name of the topic or queue.
   - **Credentials.** Optional credentials. The credentials will only be required in certain instances, depending upon the configuration of the JMS server and how we are required to connect to the JMS server (whether the JMS Message topic is secured or not).

   Note: If JBoss JMS messaging service is being used, message domain must be specified in the topic/<topic name> format.

2. Click Finish. The updated information appears in the Message Domains folder.

Modifying Message Domain Definitions

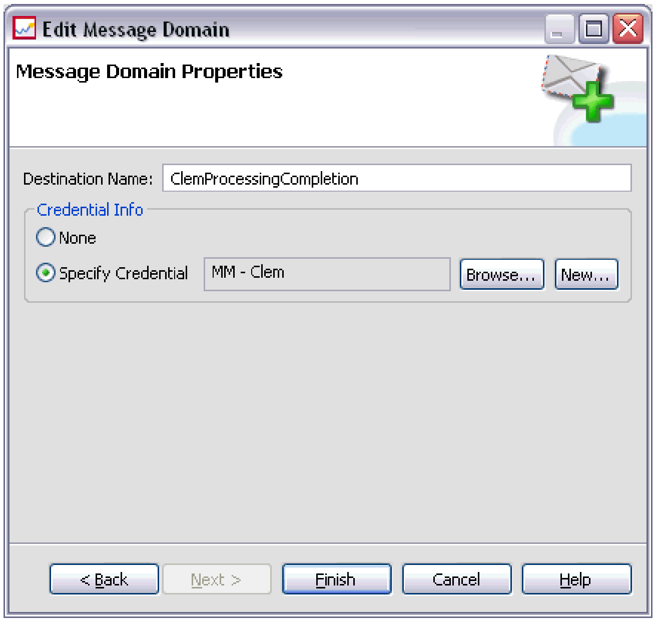
To modify a message domain definition:

1. In the Content Explorer, open the *Resource Definitions* folder.

2. Open the *Message Domains* folder.

3. Double-click the message domain. The Edit Message Domain dialog opens.

Figure 6-11
Edit Message Domain dialog



Edit Message Domain

Message Domain Properties

Destination Name: ClemProcessingCompletion

Credential Info

○ None

● Specify Credential MM - Clem

4. Modify message domain properties as necessary.
   - **Destination Name.** The name of the topic or queue.
   - **Credentials.** Optional credentials. The credentials will only be required in certain instances, depending upon the configuration of the JMS server and how we are required to connect to the JMS server (whether the JMS Message topic is secured or not).

5. Click Finish. The updated information appears in the Message Domains folder.

Server Definitions

Executing a repository resource as a job step requires the specification of an appropriate corresponding server to process the instructions contained in the job step. The connection information for such a server is specified within a **server definition**. Server definitions can be classified as either execution servers or repository servers.

- Execution servers process the contents of a repository resource. The execution server type must correspond to the resource type being processed. A SAS job step, for example, requires a SAS server definition.
- A repository server corresponds to a PASW Collaboration and Deployment Services repository installation. A server of this type is typically used by job steps that need to return result artifacts to a repository.

Server definitions are contained in the *Resource Definitions* folder of the Content Explorer. Specifically, they are defined in the *Servers* subfolder

Adding New Server Definitions

To add a new server:

1. In the Content Explorer, open the *Resource Definitions* folder.

2. Click on the *Servers* folder.

3. From the File menu, choose:

   New
     Server Definition

   The Add New Server Definition wizard opens. Alternatively, the new server definition dialog box can be accessed by clicking New next to a server field on the General tab for some steps. The process for defining new servers consists of:

1. Naming the server definition and specifying its type. For more information, see the topicServer Definition Type on p. 97.

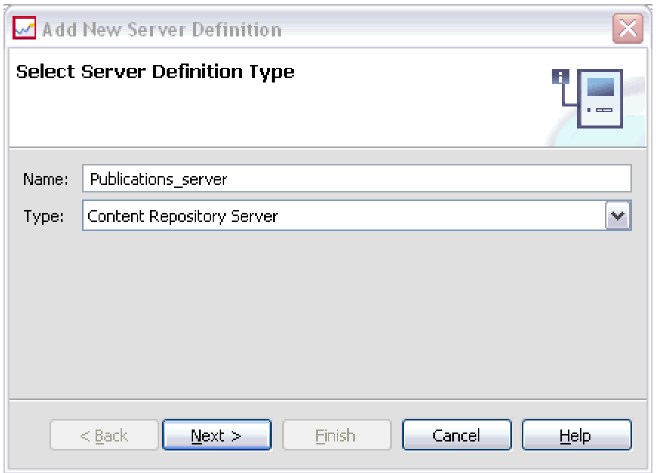
2. Selecting a location in the *Servers* folder for the definition. For more information, see the topicServer Destination on p. 98.

3. Specifying parameters for the server that define connection or execution information. The parameter set depends on the server type.

Server Definition Type

The Server Definition Type page specifies identification information for a server definition.

Figure 6-12
Defining a new server



Add New Server Definition

Select Server Definition Type

Name: Publications_server

Type: Content Repository Server

1. In the Name field, enter the name that you want to assign to your server definition.

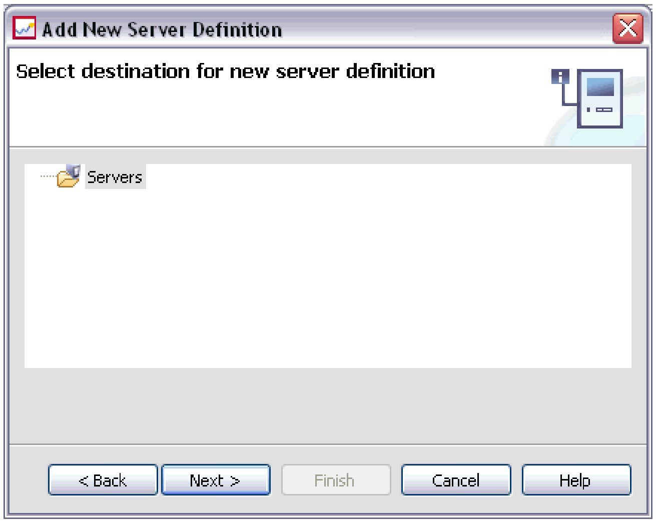
2. From the Type drop-down list, select the server type.

   Click Next to define additional parameters.

Server Destination

The Server Destination page specifies the location in the repository in which the definition is stored.

Figure 6-13
Server destination



Add New Server Definition

Select destination for new server definition

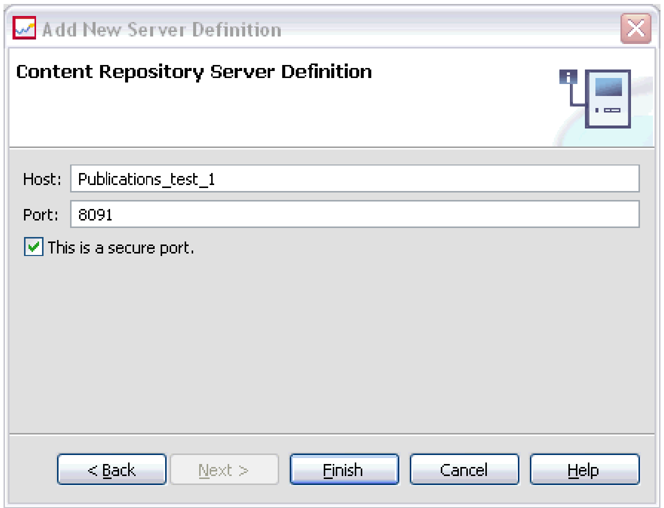
1. Navigate to the desired folder, and select it.

2. Click Next. The next dialog box that opens depends on the server type that you selected.

Content Repository Server Definition

A Content Repository Server Definition specifies the connection parameters for a repository server.

Figure 6-14
Defining a new content repository server


Add New Server Definition
Content Repository Server Definition
< Back
Next >
Finish
Cancel
Help

Host: Publications_test_1

Port: 8091

☑ This is a secure port.

1. In the Host field, enter the name of the host where the server resides.

2. In the Port field, enter the port number to be used to connect to the host.

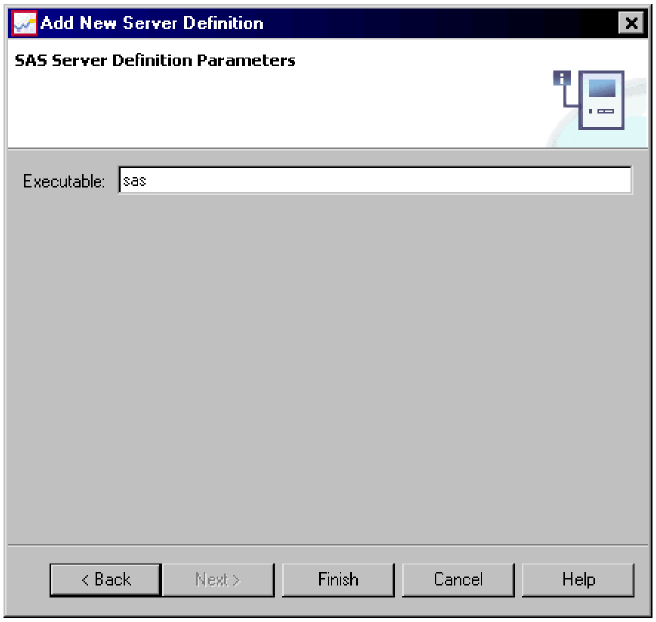
3. To specify that Secure Socket Layer (SSL) is to be used for the server connection, select This is a secure port.

4. Click Finish. The new definition appears in the *Servers* folder.

SAS Server Parameters

A SAS Server Definition specifies the executable used for processing SAS job steps. The executable must be installed on the same host as the repository. Alternatively, a remote process server could be installed on a remote machine to allow PASW Collaboration and Deployment Services to access an executable installed on the remote machine.

Figure 6-15
Defining a new SAS execution server



Add New Server Definition

SAS Server Definition Parameters

Executable: sas

In the Executable field, specify the full path to the *sas.exe* file to use as the execution server. If the system path includes the location of this file, the full path can be omitted and the default value of *sas* used.

On some systems, spaces in the path for the executable may cause failures for jobs using the execution server. These problems can be eliminated by specifying the path using 8.3 notation. For example, instead of:

```
C:\Program Files\SAS Institute\SAS\V8\sas.exe
```

specify:

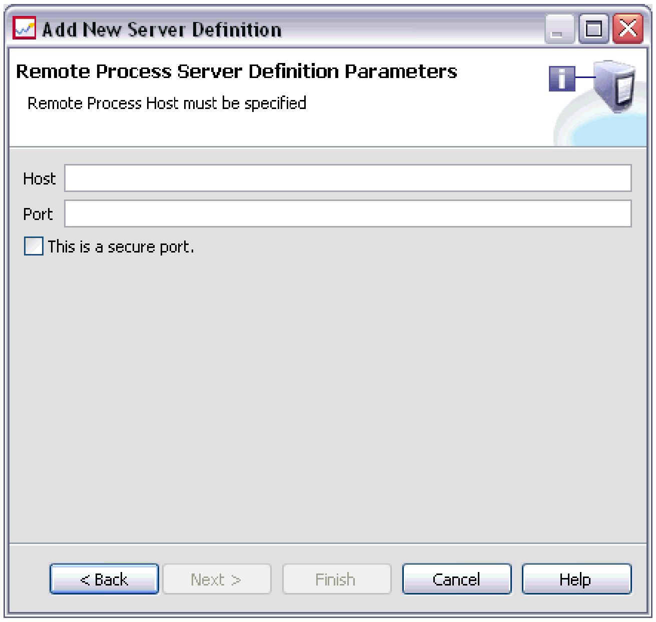
```
C:\Progra~1\SASINS~1\sas\v8\sas.exe
```

After defining the executable path, click Finish. The new definition appears in the *Servers* folder.

Remote Process Server Parameters

A Remote Process Server Definition specifies the connection parameters for a server configured for remote processing. A remote process server allows PASW Collaboration and Deployment Services to access functionality installed on the remote machine.

Figure 6-16
Defining a new remote process server



Add New Server Definition

Remote Process Server Definition Parameters

Remote Process Host must be specified

Host

Port

☐ This is a secure port.

1. In the Host field, enter the name of the host where the remote process server resides.

2. In the Port field, enter the port number to be used to connect to the host.

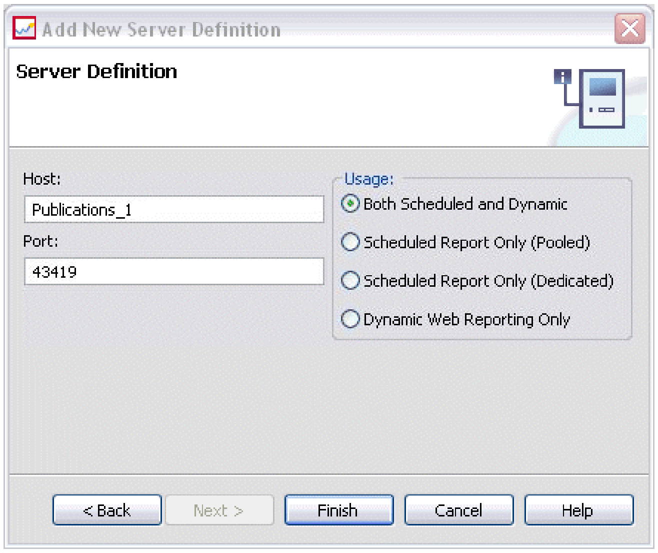
3. To specify that Secure Socket Layer (SSL) is to be used for the server connection, select This is a secure port.

4. Click Finish. The new definition appears in the *Servers* folder.

ShowCase Enterprise Server Definition

After you have specified the name and selected ShowCase Enterprise Server as the type, the Server Definition dialog box opens.

Figure 6-17
Defining a new ShowCase Enterprise Server



Host:
Publications_1
Port:
43419
Usage:
● Both Scheduled and Dynamic
○ Scheduled Report Only (Pooled)
○ Scheduled Report Only (Dedicated)
○ Dynamic Web Reporting Only

1. In the Host field, enter the machine name where the server resides.

2. In the Port field, enter the port number to be used to connect to the host (43419 is the default).

3. In the Usage section, select from the available options:

   - **Both Scheduled and Dynamic:** Select this option if you will use the server for both scheduled reporting and dynamic reporting via Deployment Portal.
   - **Scheduled Report Only (Pooled):** Select this option if you will configure multiple servers for scheduled reporting. A pooled server handles incoming scheduled report requests. Pooled servers logically act as one server, and the server with the lightest load processes incoming requests.
   - **Scheduled Report Only (Dedicated):** Select this option if you will use the server only for scheduled reporting. A dedicated server handles specific incoming scheduled report requests.
   - **Dynamic Web Reporting Only:** Select this option if you will use the server only for dynamic reporting via Deployment Portal.

4. Click Finish. The new definition appears in the *Servers* folder.

Modifying Server Definitions

To modify a server definition:

1. In the Content Explorer, open the *Resource Definitions* folder.

2. Open the *Servers* folder.

3. Double-click the server to be modified. The Edit Server Definition dialog opens.

4. Modify the server definition parameters as necessary.

5. Click Finish to save your changes.

Server Clusters

PASW Statistics, PASW Modeler, and remote process servers can be grouped into server clusters to allow load balancing. When a job step uses a server cluster for execution, PASW Collaboration and Deployment Services determines which managed server in the cluster is best suited to handle processing requests at that time. The algorithm determining which server reacts to a request depends on several criteria, including the server weights and current processing loads.

Server cluster definitions are created in the Content Explorer and stored in the repository. Write permissions to the Server Clusters folder are required to create and modify server clusters. For more information, see the topic Modifying Permissions for an Existing User or Group in Chapter 5 on p. 44. Also, the user's role must include the *Define Server Clusters* action.

Deployment Manager provides the ability to create new server clusters and modify and delete cluster definitions.

Creating a New Server Cluster

To add a server cluster definition:

1. In the Content Explorer, open the *Resource Definitions* folder.

2. Select the *Server Clusters* folder.

3. From the File menu, choose:

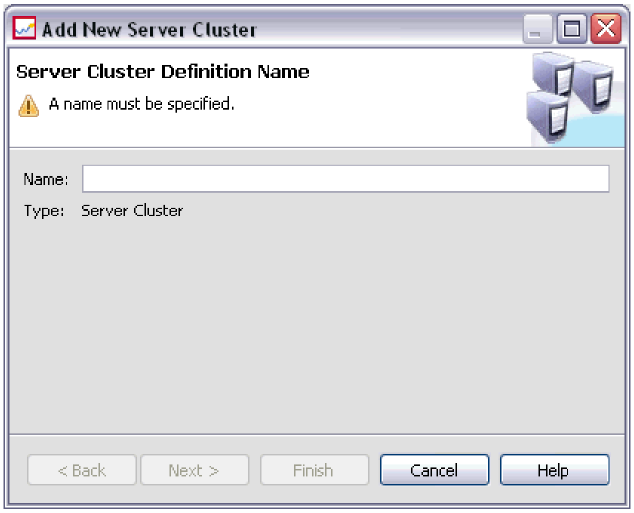
   New
     Server Cluster Definition

   The Add New Server Cluster wizard opens. This wizard provides the capability to specify the cluster name and settings.

Server Cluster Definition Name

The first step of adding a new server cluster involves the definition of the cluster name.

Figure 6-18
Add New Server Cluster wizard



Add New Server Cluster

Server Cluster Definition Name

A name must be specified.

Name:

Type: Server Cluster

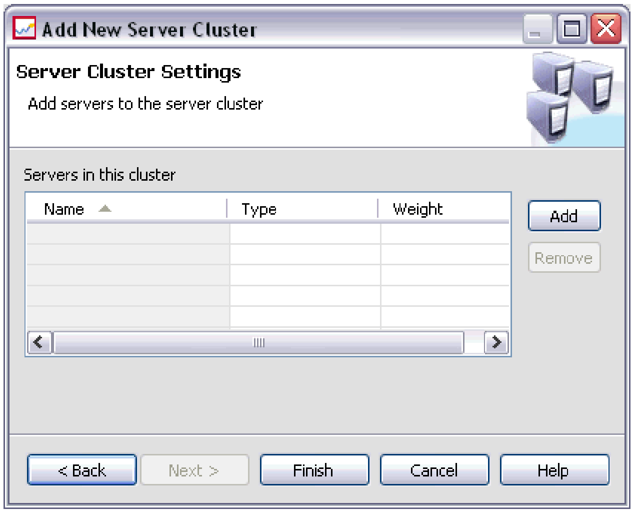
1. In the Name field, enter the name for the cluster.

2. Click Next. The Server Cluster Settings page of the wizard opens.

Server Cluster Settings

The settings for a server cluster define the servers included in the cluster and the weight associated with each server.

Figure 6-19
Server Cluster Settings



Add New Server Cluster

Server Cluster Settings

Add servers to the server cluster

Servers in this cluster

Name	Type	Weight

To add a server to the cluster, click the Add button. The Add Servers to Cluster dialog box appears. For more information, see the topic Add Servers to Cluster on p. 105.

To remove a server from the cluster, select the server to be removed in the server list and click the Remove button. To select multiple servers for removal, press the Ctrl key when selecting servers.

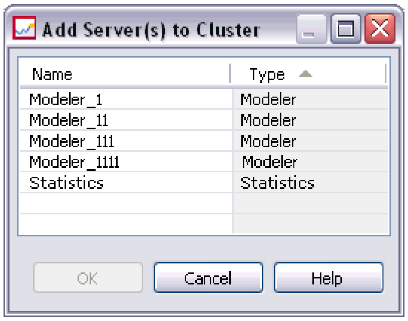
To modify the weight for a server, click the weight to be modified and click the ellipsis button that appears in the cell. The Set Server Weight dialog box appears. For more information, see the topic Set Server Weight on p. 105.

After defining the cluster settings, click Finish.

Add Servers to Cluster

The Add Servers to Cluster dialog box lists all servers on the network currently registered with PASW Collaboration and Deployment Services.

Figure 6-20
Server Cluster Settings



Add Server(s) to Cluster

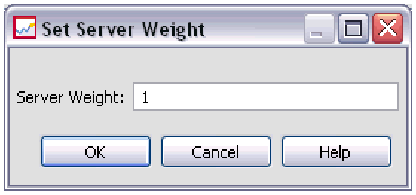
Name	Type
Modeler_1	Modeler
Modeler_11	Modeler
Modeler_111	Modeler
Modeler_1111	Modeler
Statistics	Statistics

Select the server to be added to the cluster and click the OK button. To select multiple servers, press the Ctrl key when selecting servers.

Set Server Weight

The Set Server Weight dialog box allows the specification of a weight for the server.

Figure 6-21
Server Cluster Settings



Set Server Weight

Server Weight: 1

Weights must be between 1 and 100, inclusive. Weights represent a relative value for workload capacity. For example, a server with a weight of 10 has 10 times the capacity of a server with a weight of 1. Click OK to apply the weight value to the server and return to the Server Cluster settings page.

Modifying a Server Cluster

To edit a server cluster definition:

1. In the Content Explorer, open the *Resource Definitions* folder.
2. Select the *Server Clusters* folder.
3. Right-click the cluster to be modified and select Open.

   The Edit Server Cluster dialog box opens. This dialog box provides the capability to modify the cluster settings. For more information, see the topic Server Cluster Settings on p. 104.

Importing Resource Definitions

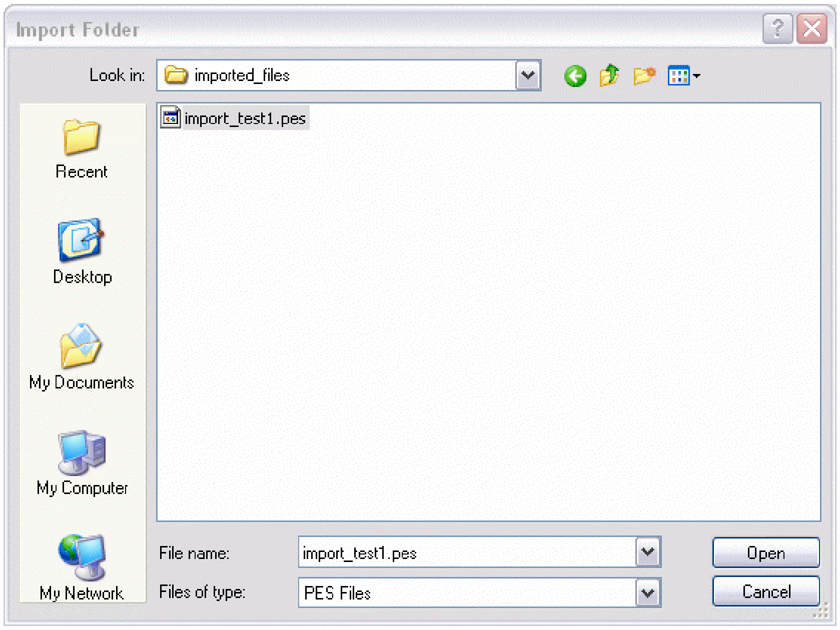
Like other repository objects, resource definitions can be exported and imported. The procedure for exporting resource definitions is similar to the procedure for exporting regular folders. For more information, see the topic Exporting Folders in Chapter 7 on p. 110. Note that all existing credential, data source, and server information is included in the export file.

Resource definitions in a PASW Collaboration and Deployment Services export file can be imported "wholesale" or separately. The procedure for importing all resource definitions from an export file is similar to the procedure for regular folder import. For more information, see the topic Importing Folders in Chapter 7 on p. 111.

To import resource definitions separately:

1. In the Content Explorer, expand the *Resource Definitions* folder.
2. Right-click on a resource definition and select Import. The Import Folder dialog box opens.

Figure 6-22
Import Folder dialog box



Look in: imported_files

import_test1.pes

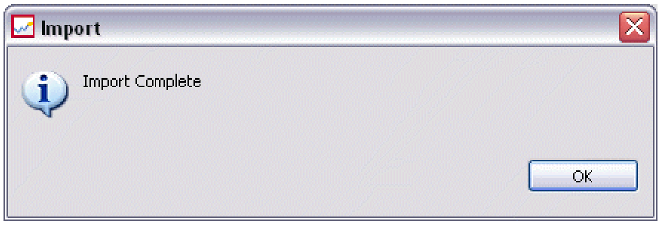
File name: import_test1.pes

Files of type: PES Files

3. Navigate to the *.pes* import file and select it.

4. Click Open. The progress dialog box appears. When the import is complete, the Import dialog box opens.

Figure 6-23
Import complete



Import

Import Complete

5. Click OK. The imported folder and its contents appear in the Content Explorer tree.

   Note: The import process performs a check of execution server definitions. If one of the definitions is invalid, the entire process fails.

Chapter

7

Exporting and Importing Folders

Export and Import Overview

Administrators can export content out of and import content into the repository. For example, it may be necessary to transfer content from one repository to another. Only repository folders can be exported and imported.

Individual files are stored in the content repository using the Add File to Repository option and retrieved using the Download File option. For more information, see Adding Files to the Repository on p. 20 and Downloading Files from the Repository on p. 21. Jobs cannot be added and downloaded as individual files. Jobs can be exported and imported as part of folder content.

When a folder is exported, a compressed archive file with the extension *.pes* is generated. The export process includes the entire contents of the folder, including subfolders, and any custom properties applied to the folder.

Important! In a clustered environment, import and export cannot be performed by accessing PASW Collaboration and Deployment Services through the load balancer. The user performing export and import must access PASW Collaboration and Deployment Services by using the URL of a dedicated node in the cluster.

Job Components That Are Migrated During Export and Import

When importing and exporting jobs, the following components of a job are migrated:

- Versions
- Notifications
- Job schedules

The job history associated with a job is not migrated during the import or export process.

Exporting External References

External references are resources that are defined outside of the folder that is being exported. For example, external references may consist of PASW Modeler streams and other resources (e.g., syntax files) that defined in different folder (i.e., not in the folder selected for export). By default, external references are included during the export process. However, the Include external references option can be disabled when exporting objects. For more information, see the topic Exporting Folders on p. 110.

It is important to note that selecting the Include external references check box has no effect in the following instances because external references are not applicable.

- When exporting from the root of the Content Repository tree.
- When exporting the Resource Definitions folder.

Export and Import Restrictions

When exporting and importing, the system enforces certain restrictions.

Export Restrictions

The content of the *Submitted Jobs* folder cannot be exported.

Import Restrictions

- All jobs containing streams must have the streams in the same relative location on the target server unless the streams are included in the *.pes* file.
- The environment of the target repository must include any adaptors needed for working with the artifacts in the archive being imported. For example, if the archive contains PASW Modeler server definitions, the target environment must include the PASW Modeler adaptors.
- When importing large archive files, it may be necessary to use a disk mapping implementation for identifier lookup. Use the Deployment Manager to change the *Resource Transfer Lookup Table* configuration value to *DISK* before attempting the import. For more information, see the Administrator's Guide.

Recommended Import Order

When importing content into repository, it is recommended that the following order be followed:

- For Enterprise View users, import Enterprise View objects (usually performed once). For more information, see the topic Importing and Exporting Enterprise View Objects in Chapter 8 on p. 206.
- Import PASW Collaboration and Deployment Services Resource Definitions (usually performed once). For more information, see the topic Importing Resource Definitions in Chapter 6 on p. 106.
- Import other objects that will be stored in the repository (usually performed multiple times).

Security Permissions When Importing Folders

The following security restrictions apply when importing folders:

- The system accepts the security configuration (permissions) of the parent folder. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.
- If a job contains a server-related job step, the system tries to match the server and credential names between the source and target machines. If the server and credential definitions do not match, the import process will fail. For more information, see the topic Adding New Credentials in Chapter 6 on p. 82.

Exporting Folders

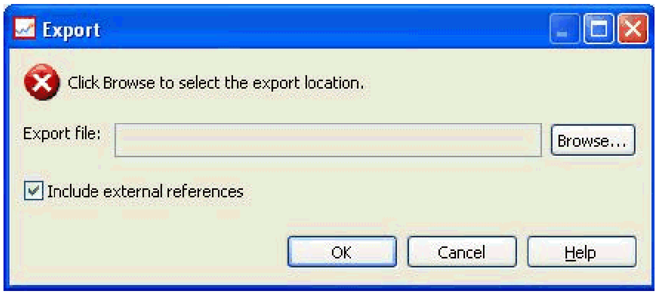
The process for exporting files consists of the following tasks:

- Specifying a location for the exported file.
- Determining whether to export external references.

To export a folder:

1. In the Content Explorer, right-click on the folder to export, and select Export. The Export dialog appears.

Figure 7-1
Specify the export path



Export

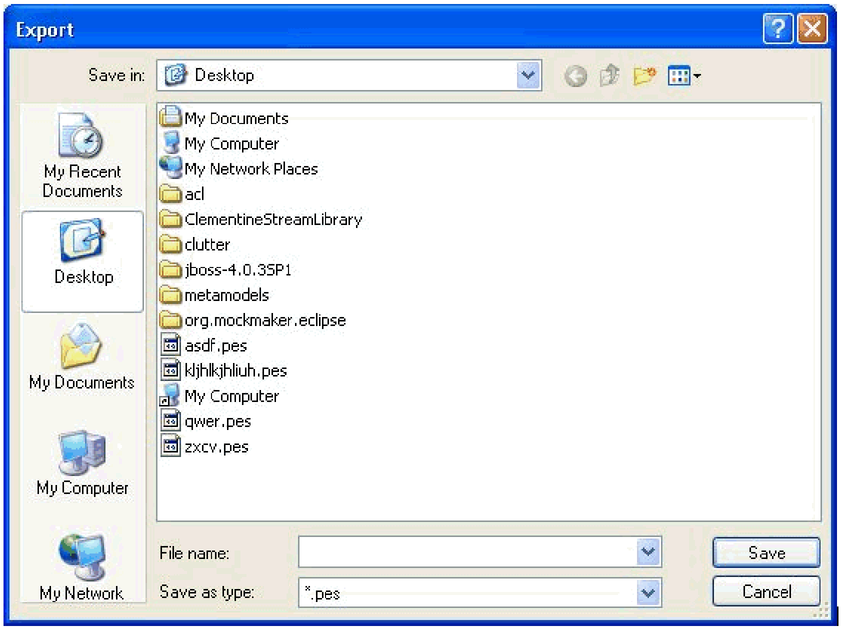
Click Browse to select the export location.

Export file:

☑ Include external references

2. To select an export destination, click Browse. The Browse for Folder dialog box opens.

Figure 7-2
Browse for the export location



Export

Save in: Desktop

My Documents
My Computer
My Network Places
acl
ClementineStreamLibrary
clutter
jboss-4.0.3SP1
metamodels
org.mockmaker.eclipse
asdf.pes
kljhlkjhliuh.pes
My Computer
qwer.pes
zxcv.pes

File name:

Save as type: *.pes

3. Select the location in which you want to store the exported folder. An existing folder can be selected or a new folder can be created.

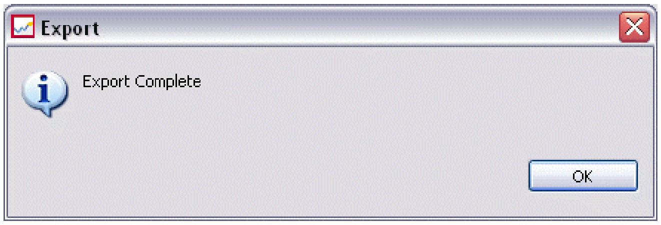
- **Using an existing folder.** Navigate to an existing folder within the tree, and select it.
- **Creating a new folder.** To create a new folder, click Make New Folder. A new folder is inserted into the tree with the name *New Folder*. Rename this folder.

4. Click Save. The Export dialog reappears.

5. Determine whether to include external references in the exported file. By default, external references are included during the export process. To exclude external references, deselect the Include external references check box. For more information, see the topicExporting External References on p. 108.

6. Click OK. The progress dialog box appears. When the export is complete, the Export dialog box opens.

Figure 7-3
Export complete



Export Complete

7. Click OK.

Importing Folders

The process for importing files consists of the following tasks:

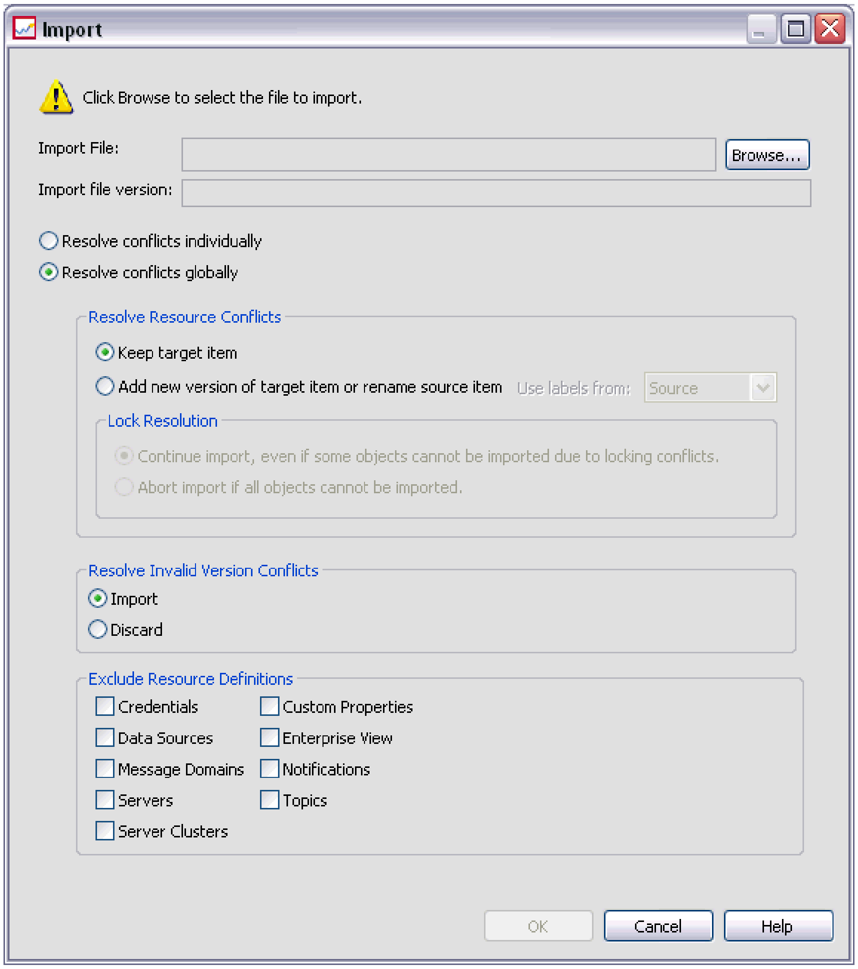
- Specifying a path.
- Determining how the system should resolve import conflicts.

Important: The version import is processed differently for Enterprise View objects. For more information, see the topic Importing and Exporting Enterprise View Objects in Chapter 8 on p. 206.

To import a file:

1. In the Content Explorer, right-click on the directory into which you want to import the folder and select Import. The Import dialog box opens.

Figure 7-4
Import dialog



Import

Click Browse to select the file to import.

Import File:

Import file version:

○ Resolve conflicts individually

● Resolve conflicts globally

Resolve Resource Conflicts

● Keep target item

○ Add new version of target item or rename source item Use labels from: Source

Lock Resolution

● Continue import, even if some objects cannot be imported due to locking conflicts.

○ Abort import if all objects cannot be imported.

Resolve Invalid Version Conflicts

● Import

○ Discard

Exclude Resource Definitions

□ Credentials □ Custom Properties

□ Data Sources □ Enterprise View

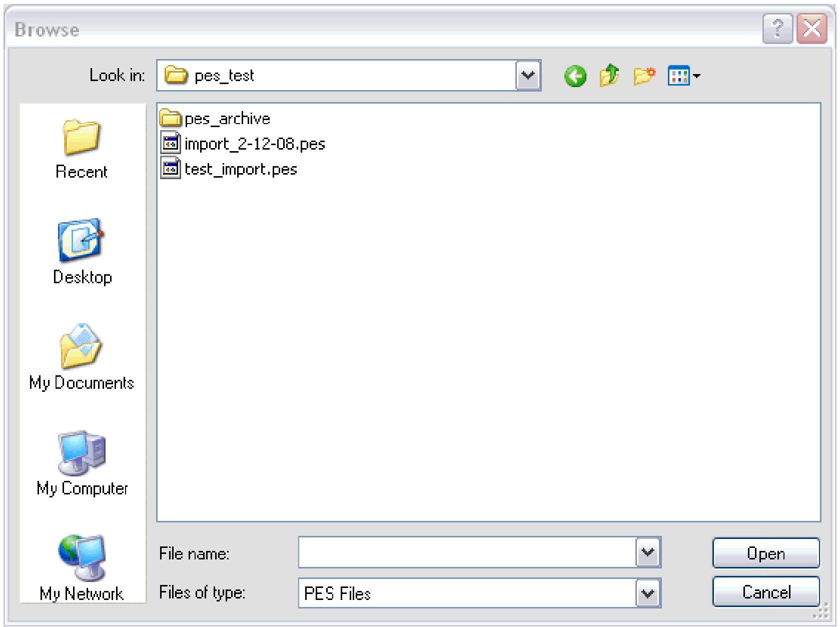
□ Message Domains □ Notifications

□ Servers □ Topics

□ Server Clusters

2. Click **Browse** to navigate to the file that you want to import . The Browse for Folder dialog appears.

Figure 7-5
Browse for the import location



pes_archive
import_2-12-08.pes
test_import.pes

3. Select the file to import, and click Open. The Import dialog returns. The name of the files selected for import appears in the Import File field. The version product version number associated with the imported file appears in the Import file version field. The import file version is determined by the system and is not modifiable.

4. Specify whether the system should resolve import conflicts individually or globally by selecting the applicable option. By default, Deployment Manager resolves conflicts individually. If global conflict resolution is selected more options are enabled in the Import dialog. Specify the settings for global conflict resolution. For more information, see the topicResolving Import Conflicts Globally on p. 115.

5. Click OK to launch the import process.

6. If any conflicts arise during the import process, the Import Conflicts dialog appears. For more information, see the topicResolving Import Conflicts on p. 114.

7. If no conflicts occurred, the progress dialog box appears. When the import is complete, the Import complete dialog box opens.

Figure 7-6
Import complete

Import Complete

8. Click OK. The imported folder and its contents appear in the Content Explorer tree.

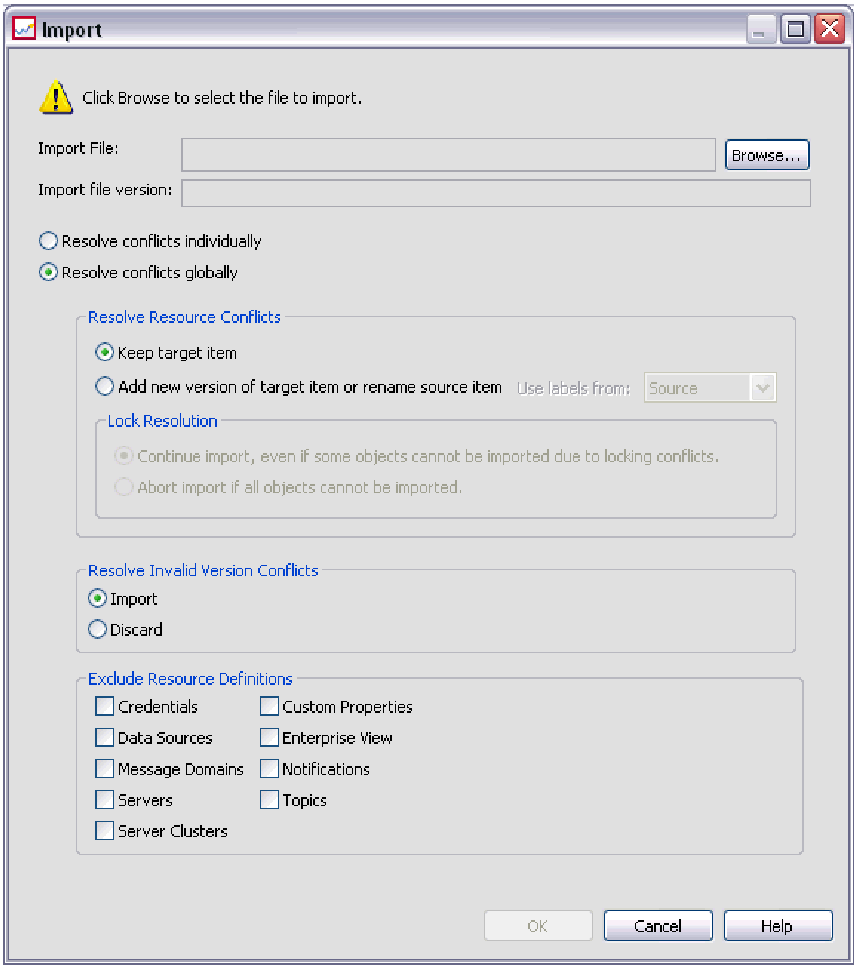
Resolving Import Conflicts

At times, an object that is imported may conflict with an existing object in the repository. For example, the imported object may be a duplicate of an object that is currently in the repository. When conflicts occur, they must be resolved before the import process can continue.

Conflicts can be resolved in a global or individual basis. Global conflict resolution is done by the system. Individual conflict resolution requires a user-specified decision for each conflict.

The method used to resolve conflicts is specified in the Import dialog. By default, the system resolves conflicts individually. Any setting changes made in this dialog persist until they are modified in a future session.

Figure 7-7
Resolving import conflicts globally or individually



Import

Click Browse to select the file to import.

Import File: [Browse...]

Import file version:

○ Resolve conflicts individually
● Resolve conflicts globally

Resolve Resource Conflicts

● Keep target item
○ Add new version of target item or rename source item Use labels from: Source

Lock Resolution

● Continue import, even if some objects cannot be imported due to locking conflicts.
○ Abort import if all objects cannot be imported.

Resolve Invalid Version Conflicts

● Import
○ Discard

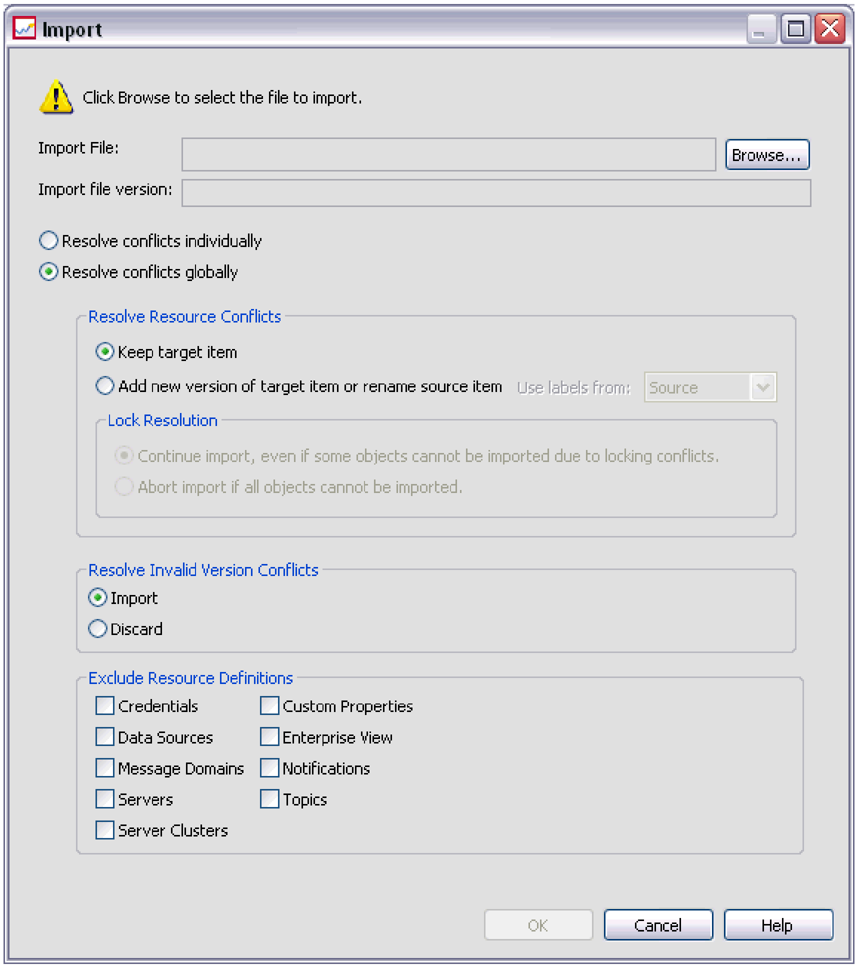
Exclude Resource Definitions

□ Credentials
□ Data Sources
□ Message Domains
□ Servers
□ Server Clusters
□ Custom Properties
□ Enterprise View
□ Notifications
□ Topics

Resolving Import Conflicts Globally

If *Resolve conflicts globally* was selected in the *Import* dialog, the *Resolve Resource Conflicts* section of the Import dialog is enabled. The settings specified in this dialog determine how the system resolves conflicts that arise during the import process. Unlike individual conflict resolution, a list of conflicts prompting user decisions for each conflict will not appear if global conflict resolution is selected. The system will automatically deal with the conflicts that arise based on the criteria specified in this dialog.

Figure 7-8
Resolving import conflicts globally



Import

Click Browse to select the file to import.

Import File:

Import file version:

○ Resolve conflicts individually
● Resolve conflicts globally

Resolve Resource Conflicts
● Keep target item
○ Add new version of target item or rename source item   Use labels from: Source

Lock Resolution
● Continue import, even if some objects cannot be imported due to locking conflicts.
○ Abort import if all objects cannot be imported.

Resolve Invalid Version Conflicts
● Import
○ Discard

Exclude Resource Definitions
□ Credentials
□ Data Sources
□ Message Domains
□ Servers
□ Server Clusters
□ Custom Properties
□ Enterprise View
□ Notifications
□ Topics

The global resolution options are organized as follows:

Resolve Resource Conflicts. If duplicate IDs or names are encountered during the import process, the following options are available:

- *Keep existing item.* The target item will be maintained. The source item with the duplicate ID, which is contained in the *.pes* file, will be ignored. This option is the default.
- *Add new version of target item or rename source item.* This option is typically used to resolve ID conflicts or naming conflicts. If a duplicate ID conflict occurs between the source object and the target object, then a new version of the object will be created in the target location. If a naming conflict occurs, the imported object will be renamed in the target location. Typically, renamed objects are appended with _1, _2, and so forth. In the event that two versions of an object have the same label, the system keeps one label and discards the duplicate label because no two versions of the same item can have the same label. Whether the source label or the target label is maintained depends upon the value specified in the *Use labels from* drop-down field. By default, the label in the source object will be maintained when a conflict occurs. To use labels from the target directory, select Target from the *Use labels from* drop-down list.

Lock Resolution

Resource locks can affect the import process. When setting import defaults for lock resolution, the following options are available:

- *Continue import, even if some objects cannot be imported due to locking conflicts.* Selecting this option may result in an impartial import. This option is the default.
- *Abort import if all objects cannot be imported.* If any conflicts are encountered due to object locks, the import process will be terminated, and the import process will fail.

For more information, see the topic Object Locking in Chapter 4 on p. 31.

Resolve Invalid Version Conflicts. If an invalid version is encountered during the import process, the following options are available:

- *Import.* The invalid version will be imported. This option is the default.
- *Discard.* The invalid version will be deleted.

Exclude Resource Definitions. By default, resource definitions associated with repository objects are imported. However, categories of resource definitions can be excluded from the import process. Options include:

- *Credentials*
- *Data sources*
- *Message domains*
- *Servers*
- *Server clusters*
- *Custom properties*
- *Enterprise View*

- *Notifications*
- *Topics*

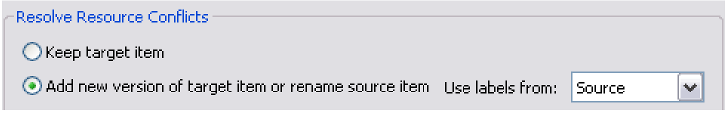
To exclude a resource definition type, select the corresponding check box.

Troubleshooting Import Failures Caused By Duplicate Name Conflicts

Occasionally, the import process may fail due to import configuration settings. If the import failure was caused by naming conflicts, it is possible that changing a setting in the Import dialog may resolve the issue. To resolve an import failure due to naming conflicts:

1. Repeat the import process. For more information, see the topicImporting Folders on p. 111.
2. In the *Resolve Resource Conflicts* section of the Import dialog, select the Add new version of target item or rename source item option.

Figure 7-9
Resolve resource conflicts



Resolve Resource Conflicts

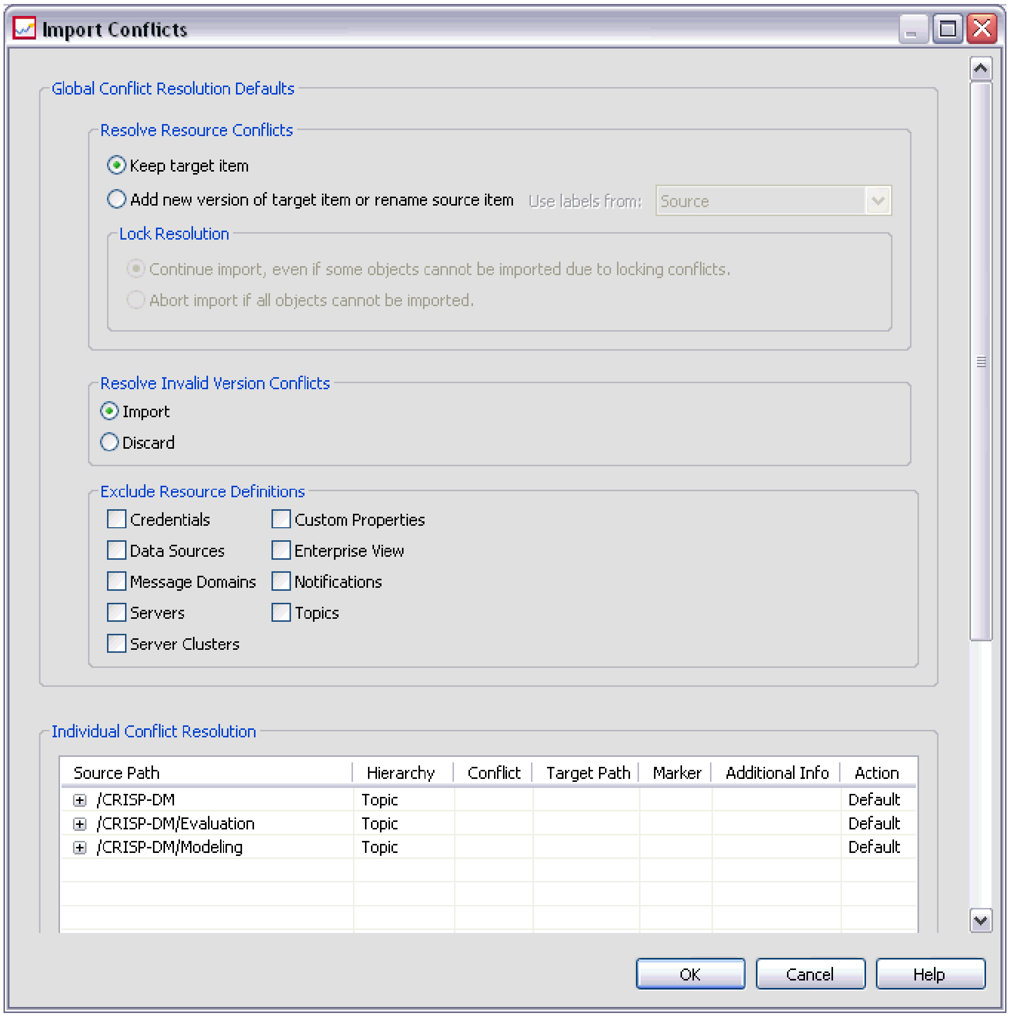
○ Keep target item

● Add new version of target item or rename source item Use labels from: Source

Resolving Import Conflicts Individually

If *Resolve conflicts individually* was selected in the *Import* dialog and conflicts arise during the process, the Import Conflicts dialog appears.

Figure 7-10
Resolving import conflicts



Import Conflicts

Global Conflict Resolution Defaults

Resolve Resource Conflicts

● Keep target item

○ Add new version of target item or rename source item Use labels from: Source

Lock Resolution

● Continue import, even if some objects cannot be imported due to locking conflicts.

○ Abort import if all objects cannot be imported.

Resolve Invalid Version Conflicts

● Import

○ Discard

Exclude Resource Definitions

□ Credentials □ Custom Properties
□ Data Sources □ Enterprise View
□ Message Domains □ Notifications
□ Servers □ Topics
□ Server Clusters

Individual Conflict Resolution

Source Path	Hierarchy	Conflict	Target Path	Marker	Additional Info	Action
⊞ /CRISP-DM	Topic					Default
⊞ /CRISP-DM/Evaluation	Topic					Default
⊞ /CRISP-DM/Modeling	Topic					Default

The process for resolving individual conflicts consists of the following tasks:

- Specifying conflict resolution defaults.
- Determining if defaults need to be overridden in the import conflicts table.

Individual Conflict Resolution Defaults

Defaults for the following import conflicts need to be specified:

Resolve Resource Conflicts. If duplicate IDs or names are encountered during the import process, the following options are available:

- *Keep existing item.* The target item will be maintained. The source item with the duplicate ID, which is contained in the *.pes* file, will be ignored. This option is the default.
- *Add new version of target item or rename source item.* This option is typically used to resolve ID conflicts or naming conflicts. If a duplicate ID conflict occurs between the source object and the target object, then a new version of the object will be created in the target location. If a naming conflict occurs, the imported object will be renamed in the target location. Typically, renamed objects are appended with _1, _2, and so forth. In the event that two versions of an object have the same label, the system keeps one label and discards the duplicate label because no two versions of the same item can have the same label. Whether the source label or the target label is maintained depends upon the value specified in the *Use labels from* drop-down field. By default, the label in the source object will be maintained when a conflict occurs. To use labels from the target directory, select Target from the *Use labels from* drop-down list.

Lock Resolution

Resource locks can affect the import process. When setting import defaults for lock resolution, the following options are available:

- *Continue import, even if some objects cannot be imported due to locking conflicts.* Selecting this option may result in an impartial import. This option is the default.
- *Abort import if all objects cannot be imported.* If any conflicts are encountered due to object locks, the import process will be terminated, and the import process will fail.

For more information, see the topic Object Locking in Chapter 4 on p. 31.

Resolve Invalid Version Conflicts. If an invalid version is encountered during the import process, the following options are available:

- *Import.* The invalid version will be imported. This option is the default.
- *Discard.* The invalid version will be deleted.

Exclude Resource Definitions. By default, resource definitions associated with repository objects are imported. However, categories of resource definitions can be excluded from the import process. Options include:

- *Credentials*
- *Data sources*
- *Message domains*
- *Servers*
- *Server clusters*
- *Custom properties*
- *Enterprise View*

- *Notifications*
- *Topics*

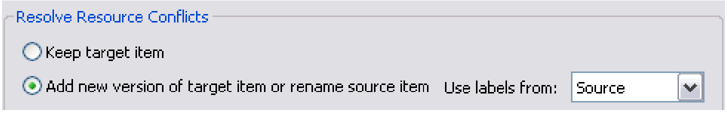
To exclude a resource definition type, select the corresponding check box.

Troubleshooting Import Failures Caused By Duplicate Name Conflicts

Occasionally, the import process may fail due to import configuration settings. If the import failure was caused by naming conflicts, it is possible that changing a setting in the Import dialog may resolve the issue. To resolve an import failure due to naming conflicts:

1. Repeat the import process. For more information, see the topicImporting Folders on p. 111.
2. In the *Resolve Resource Conflicts* section of the Import dialog, select the Add new version of target item or rename source item option.

Figure 7-11
Resolve resource conflicts



Resolve Resource Conflicts

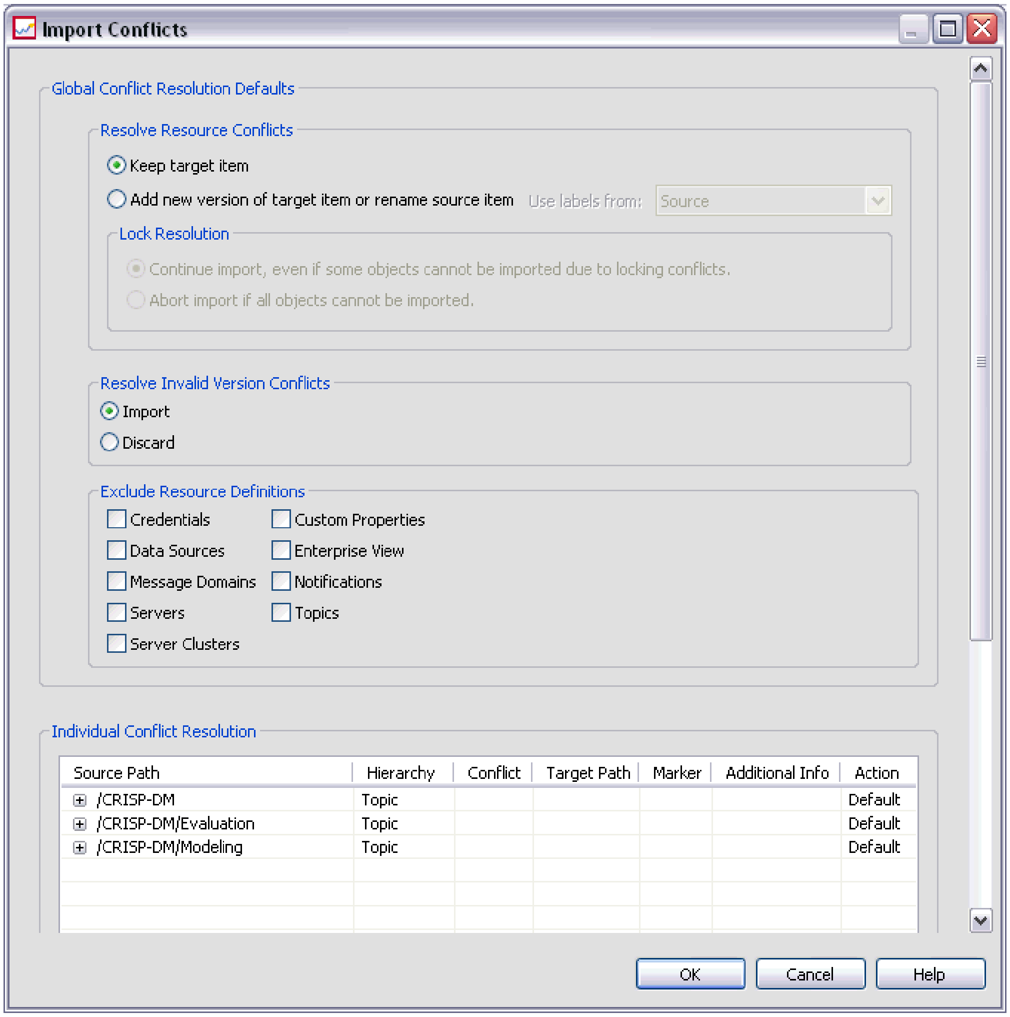
○ Keep target item

● Add new version of target item or rename source item   Use labels from: Source

Working With the Import Conflicts Table

Individual import conflicts are listed in the Import Conflicts table.

Figure 7-12
Resolving import conflicts



Import Conflicts

Global Conflict Resolution Defaults

Resolve Resource Conflicts

● Keep target item

○ Add new version of target item or rename source item Use labels from: Source

Lock Resolution

● Continue import, even if some objects cannot be imported due to locking conflicts.

○ Abort import if all objects cannot be imported.

Resolve Invalid Version Conflicts

● Import

○ Discard

Exclude Resource Definitions

□ Credentials
□ Data Sources
□ Message Domains
□ Servers
□ Server Clusters
□ Custom Properties
□ Enterprise View
□ Notifications
□ Topics

Individual Conflict Resolution

Source Path	Hierarchy	Conflict	Target Path	Marker	Additional Info	Action
⊞ /CRISP-DM	Topic					Default
⊞ /CRISP-DM/Evaluation	Topic					Default
⊞ /CRISP-DM/Modeling	Topic					Default

The Import Conflicts table contains the following information. With the exception of the Action column, none of the information in the table can be modified.

Source Path. The source location of the object to be imported.

Hierarchy. The location within the repository hierarchy—for example, folder.

Conflict. The nature of the import conflict. Examples include *Duplicate* and *Invalid version*.

Target. The destination location of the object to be imported. The Destination path field is populated only for duplicates.

Marker. The object identifier of the item. Conflicts are determined by this identifier. For example, the system searches for duplicate object identifiers.

Additional Info. Any supplementary information about the import conflict. For example, *Missing PASW Modeler stream* might appear in this column, if that is the cause of an invalid version.

Action. The *Default* that appears in the Action column refers to the default actions specified in the *Conflict Resolution Defaults* section of this dialog. In Action column, the default can be overridden on an individual conflict basis. The options that appear vary depending upon the type of conflict. For example:

- Actions for duplicates—Actions for duplicates include *Use global default*, *Keep target item*, and *Use imported item*.
- Actions for invalid versions—Actions for invalid versions include *Default*, *Import*, and *Ignore*.

To modify the default Action setting:

1. Highlight the conflict whose default setting you want to modify.
2. Click the ellipsis button next to *Default* in the Action column. A dialog, which is specific to the type of conflict appears.
3. Select the desired options within the dialog and click OK. The import conflicts table returns. For items whose default actions were modified, *Custom* appears in the Action column.
4. After all desired default actions have been modified in the table, click OK to continue the import process.

Chapter
8

Enterprise View

Enterprise View Overview

Enterprise View is a component of PASW Collaboration and Deployment Services and provides a single, consistent view of enterprise data. Enterprise View allows users to define and maintain a common view of warehoused and transaction data needed to perform analytics, optimization, deployment, and reporting. Underlying data may come from a variety of sources, including a data warehouse, an operational data store, and an online transaction database. Enterprise View ensures a consistent use of enterprise data and hides the complexities of stored data structures from the end user. Enterprise View is the data backbone for the predictive enterprise.

Data discovery requires a major investment of resources from the organizations deploying predictive analytics. The process is labor intensive—it can involve representatives from departments across the organization and often entails resolving differences in data structure and semantics across organizational boundaries. Enterprise View provides a mechanism for recording the outcomes of the data discovery process, versioning and securing the resulting schema, and tracking changes over time.

Enterprise View Builder Interface

The Enterprise View Builder user interface is an extension to the Deployment Manager interface. Enterprise View Builder includes a set of editing and administration interfaces used by data experts to create and manage Enterprise View. Refer to Deployment Manager for an overview.

The Enterprise View Builder user interface consists of:

- **Enterprise View Editor**—Records information about the entities and attributes (tables and columns) used in predictive applications. In a given repository, the Enterprise View is a singleton object; it may have versions, but there cannot be more than one version tree. For more information, see the topic Enterprise View on p. 124.
- **Application View Editor**—Overlays a subset of the Enterprise View, recording how the Enterprise View tables and columns are used in the context of a specific application (for example, call center or fraud detection). For more information, see the topic Application View on p. 146.
- **Data Provider Definition Editor**—Records information needed to access business data in a particular context. This allows data to flow from different sources depending on what the using-application does with the data. For example, the values for CustomerID might flow from a warehouse table in a PASW Modeler stream used to build an analytic, while the values might flow from physical data source. The Data Provider Definition supplies

the information needed at runtime to access data. For more information, see the topic Data Provider Definition on p. 155.

- **Data Provider Definition - Real Time Editor**—Used exclusively for Real Time Server interactions with applications. For example, the values for CustomerID might flow from a warehouse table in a PASW Modeler stream used to build an analytic, while the values might flow from a user-interface (via a data object) when the scorable analytic produced by the stream is used in a real-time scoring scenario. Unlike a Data Provider Definition, a Data Provider Definition - Real Time can only retrieve one row of data at a time. For more information, see the topic Data Provider Definition - Real Time on p. 162.

Credential Definitions

Enterprise View objects, namely the Enterprise View and Data Provider Definition, require access to data from physical data sources. In order to log in to these data sources, credential definitions must be created for each data source. Each credential definition contains a user ID and password. For more information, see the topic Adding New Credentials in Chapter 6 on p. 82.

Data Source Definitions

Enterprise View Builder requires both an ODBC and JDBC data source definition for every data source. Both definition types should point to the same data source. For more information, see the topic Working with Data Source Definitions in Chapter 6 on p. 85.

Keep in mind the following:

- The Data Provider Definition - Real Time requires only JDBC data sources.
- For ODBC data source definition types, the referenced ODBC data source names (DSNs) must exist on the same system on which the repository is installed.
- When configuring DSNs, regardless of type (Oracle, SQL Server, DB2, and so on), ensure that the **quoted identifier** option is enabled (if available).
- When configuring DSNs, regardless of type (Oracle, SQL Server, DB2, and so on), you must provide the appropriate database information.
- Third-party JDBC drivers, such as for Netezza or Teradata, can be used by augmenting your installation with the driver files. For more information, see the topic Third-Party JDBC Drivers in Chapter 6 on p. 90.

Enterprise View

The Enterprise View is an abstracted dictionary of the enterprise data used in predictive analytic applications. It provides a simplified business-oriented picture of the enterprise data available for data mining and reporting. The Enterprise View is constructed and maintained by a Data Expert (a database administrator, for example). The Enterprise View Editor in Deployment Manager provides the Data Expert with a graphical user interface to build, maintain, and validate global information about enterprise data. Taken together, the Enterprise View Editor, the Application

View Editor, the Data Provider Definition Editor, and the Data Provider Definition - Real Time constitute the Enterprise View Builder user interface used by data experts in constructing, validating, and managing Enterprise View.

Upon installation, the Enterprise View is automatically created as an empty singleton object alongside the repository. The data expert populates the empty Enterprise View with table and column metadata used in predictive analytic applications. Changes to the Enterprise View are handled in the same manner as all repository objects; a new version is created and all previous versions continue to exist, unless they are deleted manually.

It is not possible to delete the Enterprise View from the repository completely. At least one version is required. Only users who are members of the **Administrators** role, and those assigned to a role with the **Manage Enterprise View** action, are allowed to work with the Enterprise View.

Enterprise View Editor

The Enterprise View Editor allows the data expert to work with the Enterprise View definition. The editor provides options for creating, editing, and removing the logical entities and attributes required by enterprise level applications. The data expert can create table and column entries manually or import table/column information from physical database tables.

Figure 8-1
Enterprise View Editor

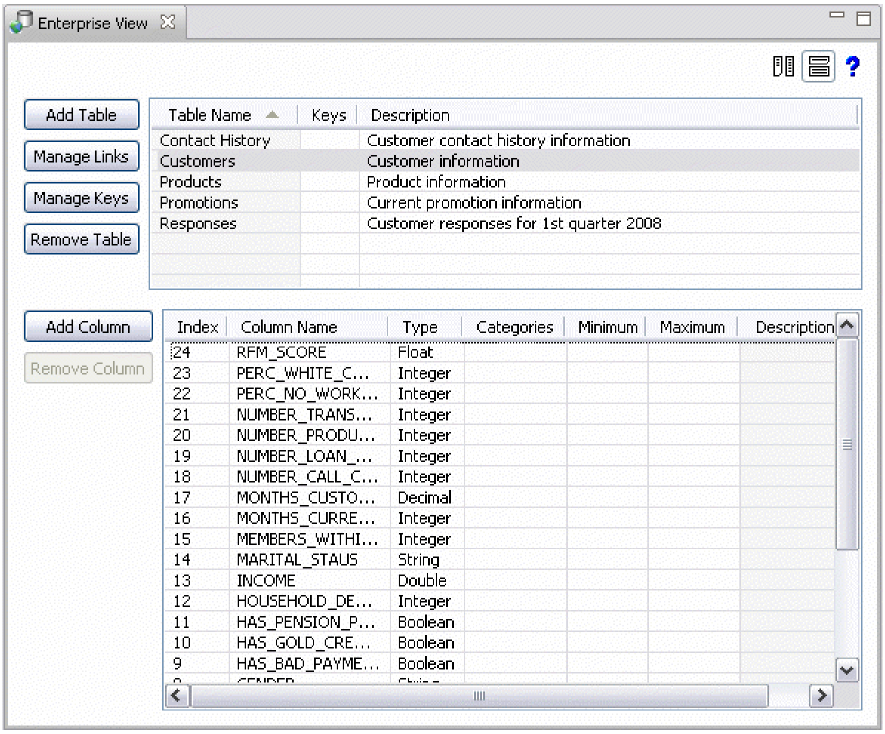


Table Name	Keys	Description
Contact History		Customer contact history information
Customers		Customer information
Products		Product information
Promotions		Current promotion information
Responses		Customer responses for 1st quarter 2008

Index	Column Name	Type	Categories	Minimum	Maximum	Description
24	RFM_SCORE	Float				
23	PERC_WHITE_C...	Integer				
22	PERC_NO_WORK...	Integer				
21	NUMBER_TRANS...	Integer				
20	NUMBER_PRODU...	Integer				
19	NUMBER_LOAN_...	Integer				
18	NUMBER_CALL_C...	Integer				
17	MONTHS_CUSTO...	Decimal				
16	MONTHS_CURRE...	Integer				
15	MEMBERS_WITHI...	Integer				
14	MARITAL_STAUS	String				
13	INCOME	Double				
12	HOUSEHOLD_DE...	Integer				
11	HAS_PENSION_P...	Boolean				
10	HAS_GOLD_CRE...	Boolean				
9	HAS_BAD_PAYME...	Boolean				
8	GENDER	String				

Note: Only users with **ADMIN** authority, and those assigned to a role with the proper authority, are allowed to work with the Enterprise View.

Adding Tables to the Enterprise View

Figure 8-2
Enterprise View Editor

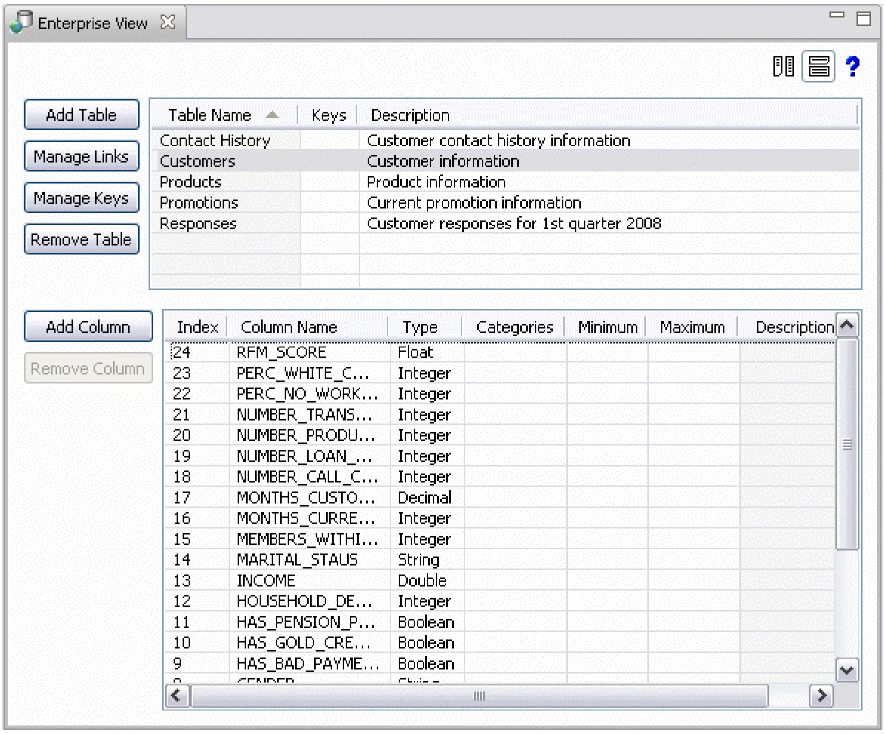
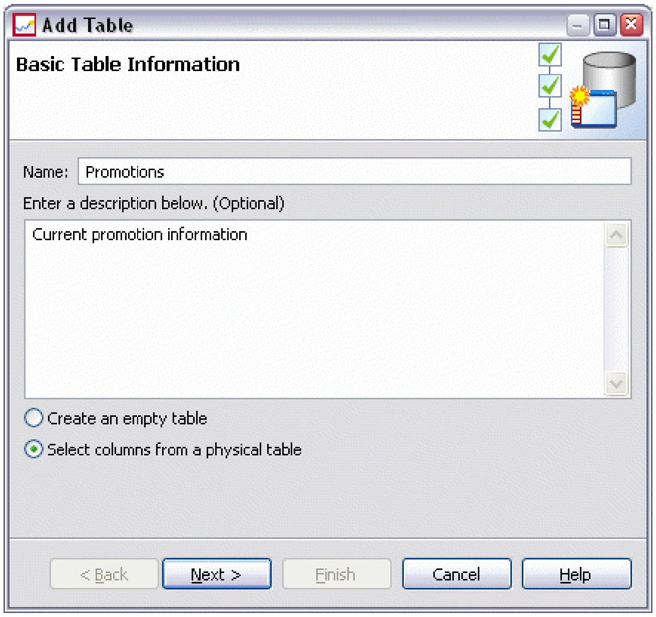


Table Name	Keys	Description
Contact History		Customer contact history information
Customers		Customer information
Products		Product information
Promotions		Current promotion information
Responses		Customer responses for 1st quarter 2008

Index	Column Name	Type	Categories	Minimum	Maximum	Description
24	RFM_SCORE	Float				
23	PERC_WHITE_C...	Integer				
22	PERC_NO_WORK...	Integer				
21	NUMBER_TRANS...	Integer				
20	NUMBER_PRODU...	Integer				
19	NUMBER_LOAN_...	Integer				
18	NUMBER_CALL_C...	Integer				
17	MONTHS_CUSTO...	Decimal				
16	MONTHS_CURRE...	Integer				
15	MEMBERS_WITHI...	Integer				
14	MARITAL_STAUS	String				
13	INCOME	Double				
12	HOUSEHOLD_DE...	Integer				
11	HAS_PENSION_P...	Boolean				
10	HAS_GOLD_CRE...	Boolean				
9	HAS_BAD_PAYME...	Boolean				

Clicking Add Table in the Enterprise View Editor launches the Add Table Wizard. The wizard provides options for specifying a new table name and accompanying description as well as allows you to create an empty table or copy the table structure from an existing data source.

Figure 8-3
Enterprise View Add Table dialog box



Name: Promotions

Enter a description below. (Optional)

Current promotion information

○ Create an empty table

● Select columns from a physical table

Name. Enter an appropriate table name. Table names are limited to 256 characters and cannot contain leading or trailing spaces. Each Enterprise View table name must be unique.

Enter a description below (optional). Enter an appropriate table description.

Create an empty table. When this option is selected, you must provide a table name and description (optional) and then click Finish to create an empty table.

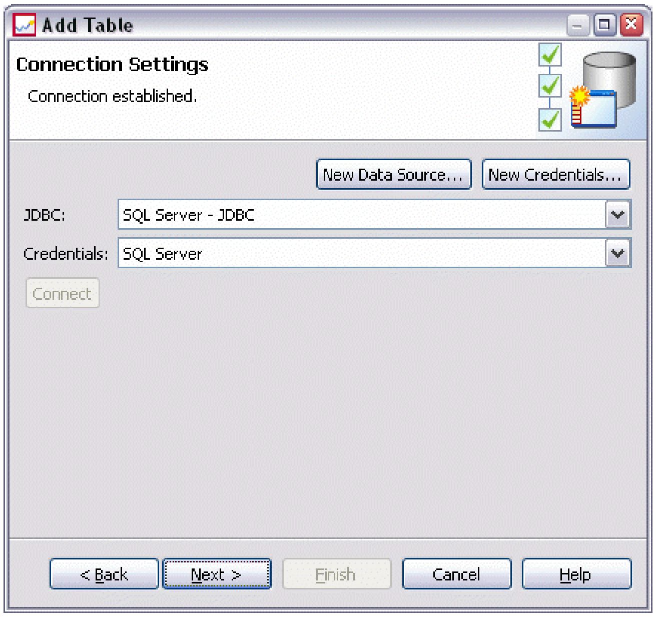
Select columns from a physical table. When this option is selected, additional wizard steps become available, allowing you to define additional table attributes.

▶ After entering the appropriate information, click Next to continue to the Add Table/Column: Connection Settings step.

Add Table/Column: Connection Settings

The Connection Settings dialog box provides options for selecting a JDBC data source provider and data source credentials. The dialog also provides options for creating new data sources and credentials. Refer to Selecting a Data Source Definition Type and Adding New Credentials for more information.

Figure 8-4
Add Table/Column: Connection Settings



JDBC: SQL Server - JDBC

Credentials: SQL Server

JDBC. Select an appropriate JDBC data source definition.

Credentials. Select an appropriate credential definition.

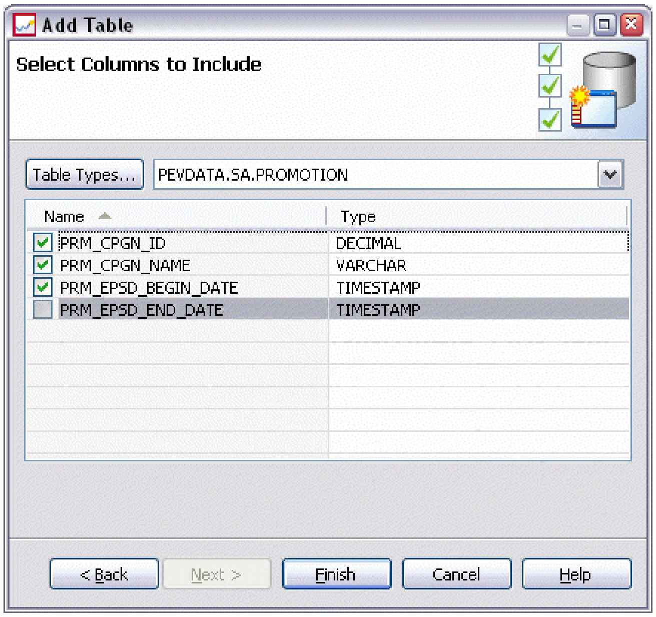
Connect. You must click Connect to verify the defined data source connection.

▶ If the connection step does not return any errors, click Next to continue the Add Table/Column: Select Columns to Include step.

Add Table/Column: Select Columns to Include

The Select Columns to Include dialog box provides options for selecting an existing data source table and its associated column definitions.

Figure 8-5
Add Table/Column: Select Columns to Include



PEVDATA.SA.PROMOTION

	Name	Type
☑	PRM_CPGN_ID	DECIMAL
☑	PRM_CPGN_NAME	VARCHAR
☑	PRM_EPSD_BEGIN_DATE	TIMESTAMP
□	PRM_EPSD_END_DATE	TIMESTAMP

Table Types... Click to select what to return from the physical data source.

- **System Table—**Returns data source system tables (for example, sysdevices and sysfiles).
- **Table—**Returns regular data source tables (for example, CUSTOMERS and ORDERS).
- **View—**Returns table views. Views are virtual tables whose contents are defined by a query. As with a real table, a view consists of a set of fields and records. The records and fields come from tables referenced in the query defining the view and are produced dynamically when the view is referenced.

Note: Other options may be available, depending on the data source configuration.

Note: When using the Add Table Wizard, you can select or deselect any available physical column. When using the Add Column Wizard, you can select and deselect only physical columns that are not already defined for the Enterprise View. For example, if the column CUSTOMER_ID from physical table CUSTOMERS is already defined in the Enterprise View, you cannot select or deselect the column in the Select Columns to Include dialog box.

▶ After selecting the appropriate table and columns, click Finish to return to the Enterprise View Editor.

Note: By default, all available columns are selected automatically.

The new table appears in the table section (when using the Add Table Wizard), and the selected column definitions that were copied from the defined data source are displayed in the column section.

Figure 8-6
Populated Enterprise View

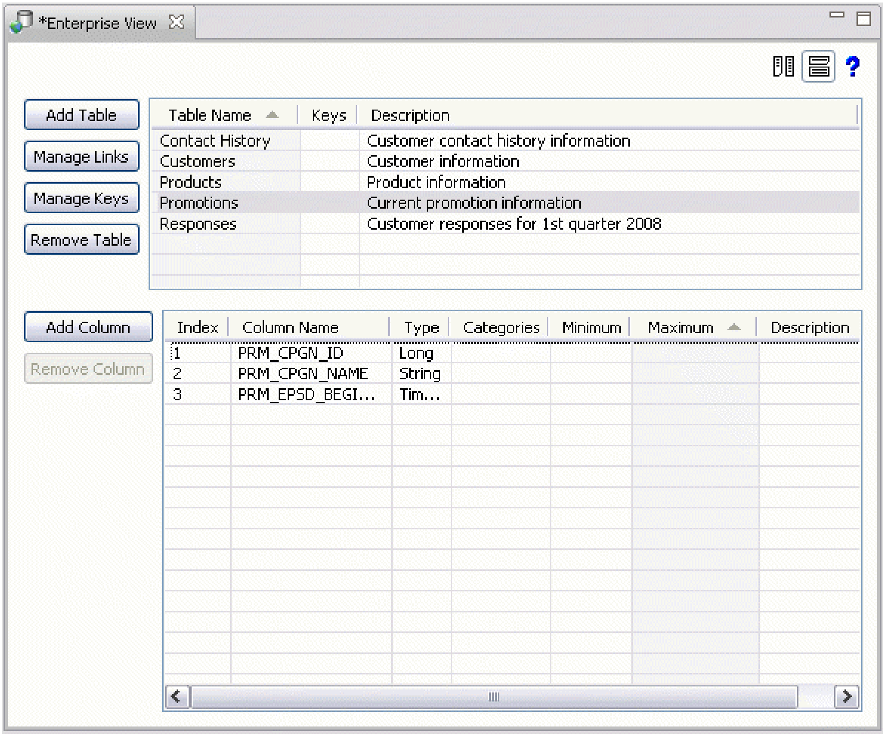


Table Name	Keys	Description
Contact History		Customer contact history information
Customers		Customer information
Products		Product information
Promotions		Current promotion information
Responses		Customer responses for 1st quarter 2008

Index	Column Name	Type	Categories	Minimum	Maximum	Description
1	PRM_CPGN_ID	Long				
2	PRM_CPGN_NAME	String				
3	PRM_EPSD_BEGI...	Tim...				

Removing Tables from the Enterprise View

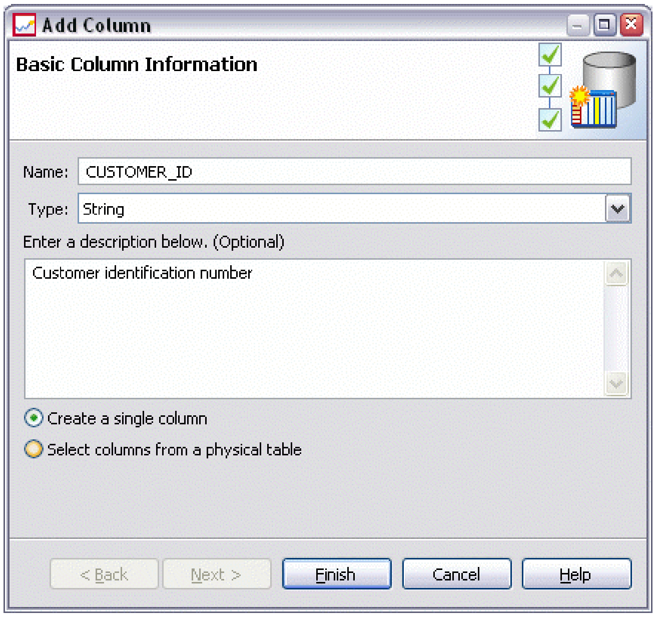
Existing tables can be removed by:

- ▶ Select the desired table.
- ▶ Click Remove Table. A dialog box opens, informing you that the table and all associated columns will be deleted and that the action cannot be undone.

Adding Columns to the Enterprise View

Clicking Add Column from the Enterprise View Editor launches the Add Column Wizard. The wizard provides options for specifying a new column name and accompanying description as well as allows you to create a single column or select the column structure from an existing data source.

Figure 8-7
Enterprise View Add Column Wizard



Basic Column Information

Name: CUSTOMER_ID

Type: String

Enter a description below. (Optional)

Customer identification number

● Create a single column

○ Select columns from a physical table

Name. Enter an appropriate column name. Each Enterprise View column name must be unique. Column names are limited to 256 characters and cannot contain leading or trailing spaces.

When creating column names, you should take into account the enterprise data environment. The type of database that will be mapped to the columns can influence the lengths of column names. For example, when an Oracle database supplies the data for the columns, the column names can have a maximum of 30 characters due to restrictions within that database. Column names exceeding this length will need to be modified to accommodate the physical data.

Type. Select the appropriate column data type. Each column can store data consisting of only a single data type. For more information, see the topic Type on p. 134.

Enter a description below (optional). Enter an appropriate column description.

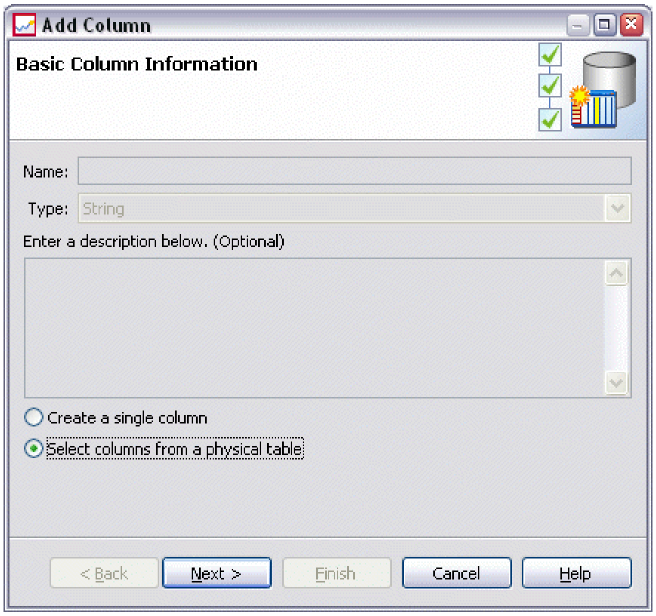
Create a single column. When this option is selected, you must provide a column name, type, and description (optional) then click Finish to create an empty column.

Select columns from a physical table. When this option is selected, additional wizard steps become available, allowing you to define which columns to add from the physical data source.

▶ To create a column manually, select Create a single column, enter an appropriate column name, type, and description (optional), and click Finish. You are returned to the Enterprise View Editor and the new column for the selected table.

- To copy one or more column definitions from a physical data source, select Select columns from a physical table and click Next to continue to the Add Table/Column: Connection Settings step.

Figure 8-8
Enterprise View Add Column Wizard



Add Column

Basic Column Information

Name:

Type: String

Enter a description below. (Optional)

○ Create a single column

● Select columns from a physical table

Sorting Columns

All columns can be sorted in ascending or descending order. To sort a column, click an appropriate column heading from the column section. The arrow displayed in the heading indicates whether the column is sorted in ascending or descending order.

Note: Sorting does not affect how columns are displayed for consuming applications. The index order dictates how columns are displayed for consuming applications. For more information, see the topic Index on p. 149..

Removing Columns from the Enterprise View

To remove existing columns:

- Select the desired columns.

- Click Remove Column. A dialog box opens, informing you that the column will be deleted and that the action cannot be undone.

Changing Enterprise View Column Attributes

Column attributes can be changed by double-clicking in the desired column cell. You can also select multiple rows, right-click, and apply the same attribute changes to all selected rows (for example, you can apply the same **Type** to all selected rows).

Note: Changing row attributes via multirow selection erases the Deployment Manager Undo/Redo history.

Index. The *Index* column represents the order in which the Enterprise View columns will be presented to consuming applications. For more information, see the topic Index on p. 133.

Column Name. Displays the current column name. For more information, see the topic Column Name on p. 133.

Type. Specifies the type of data stored in a column. Each column can store data consisting of only a single data type. For more information, see the topic Type on p. 134.

Categories. The categories attribute allows you to define categorical values for a column.For more information, see the topic Categories on p. 135.

Minimum/Maximum. The minimum/maximum attributes function as user aids. For more information, see the topic Minimum/Maximum on p. 139.

Description. Allows you to enter detailed table/column description text. For more information, see the topic Description on p. 140.

Index

The column index order dictates how columns are displayed for consuming applications. In the following table, the columns will be displayed to consuming applications in the order *Income*, *State*, *Ownership*, *Mortgage*, and *Second Mortgage*.

Index	Column name
4	Mortgage
2	State
5	Second Mortgage
3	Ownership
1	Income

By default, the index order is determined by the order in which the columns were defined in the Enterprise View. The index order can be edited manually at the Enterprise View or the Application View level.

Note: When the index order is changed at the Application View level, it does not affect the Enterprise View indexing order.

Column Name

The column name attribute displays the current column name inherited from the Enterprise View.

When creating column names, you should take into account the enterprise data environment. The type of database that will be mapped to the columns can influence the lengths of column names. For example, when an Oracle database supplies the data for the columns, the column names can have a maximum of 30 characters due to restrictions within that database. Column names exceeding this length will need to be modified to accommodate the physical data.

Type

The type attribute specifies the type of data stored in a column. Each column can store data consisting of only a single data type. For more information, see the topic Enterprise View Data Types on p. 207.

Table 8-1
Data types

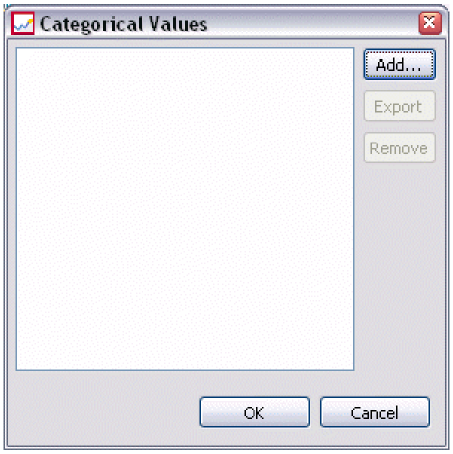
Type	**Description**
Boolean	Used to store binary digits (0 or 1). These can be used to denote Boolean values, such as Yes/No, True/False, or On/Off.
Date	Values must have the form CCYY-MM-DD, where CCYY denotes the year, MM denotes the number of the month (January = 1, February = 2, and so on), and DD denotes the day (starting with 1). Optionally, the year can be preceded by + or –. Leading zeros for the year, month, and day are optional. Spaces are not allowed before or after the – separator.
Daytime	Values must have the form HH:MM:SS.xxx, where HH denotes the hours (ranging from 0 to 23), MM denotes the minutes (0 to 59), SS denotes the seconds (0 to 59), and xxx, the milliseconds (0 to 999). First, note that only the 24-hour representation is allowed. Furthermore, the seconds and milliseconds are optional. Leading zeros for the hours, minutes, and seconds are optional. Spaces are not allowed before or after the : separator.
Decimal	Used to store exact numbers. The scale and maximum precision can be configured. The precision denotes the maximum number of digits each value may consist of. The scale indicates how many of those digits may occur to the right of the decimal point (that is, the fractional part). The value of this parameter must be greater than or equal to zero and less than or equal to the precision.
Double	Represents signed, approximate numerical values. Double permits much larger numbers than Float.
Float	Represents signed, approximate numerical values. Double permits much larger numbers than Float.
Integer	Used for whole numbers (for example, 5 or 110).
Long	Used for character data of variable length up to two gigabytes.
String	Used to store a sequence of ISO-8859-Latin-1 characters. The maximum length of such a sequence is configurable. Furthermore, the string may be empty and cannot contain the newline (ASCII 10) or null (ASCII 0) characters.
Timestamp	Allows values of the form CCYY-MM-DDTHH:MM:SS.xxx in accordance with the ISO 8601 standard. It is easy to see that it is nothing more than the concatenation of the Date and the Time types, using a delimiter of *T* to indicate the beginning of the Time portion. Consequently, the rules for the Date and the Time types are also applicable to the Timestamp type.

Categories

The categories attribute allows you to define categorical values for a column. The category attributes are used by consuming applications to populate the values for selection fields or menus. Categorical values can be added to any column regardless of its Type.

▶ Double click inside a Categories column cell to display the ... icon. Click the icon to display the Categorical Values dialog box.

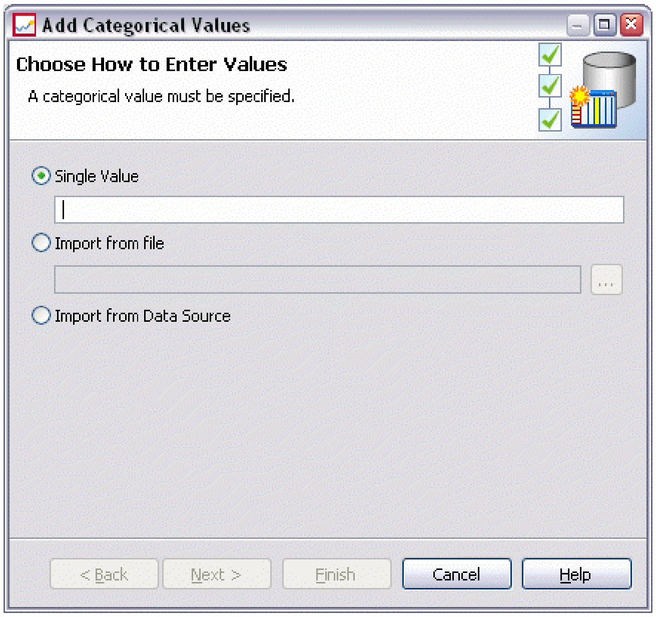
Figure 8-9
Enterprise View Categorical Values dialog box



Defining Categories

▶ Click Add... to launch the Add Categorical Values wizard.

Figure 8-10
Enterprise View Categorical Values dialog box

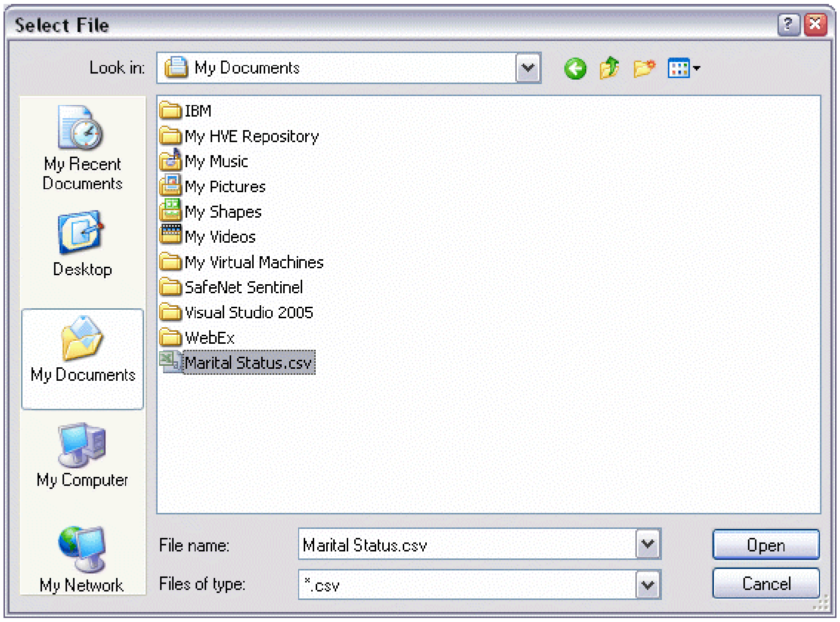


Single Value. Select this option to manually enter a single categorical value. After you enter an appropriate value, click Finish to add the entry to the Categorical Values list.

Import from file. Select this option to import categorical values from an existing *.csv* file.

▶ Click ... to display the Select File dialog box.

Figure 8-11
Enterprise View Select File dialog box



Look in: My Documents

File name: Marital Status.csv

Files of type: *.csv

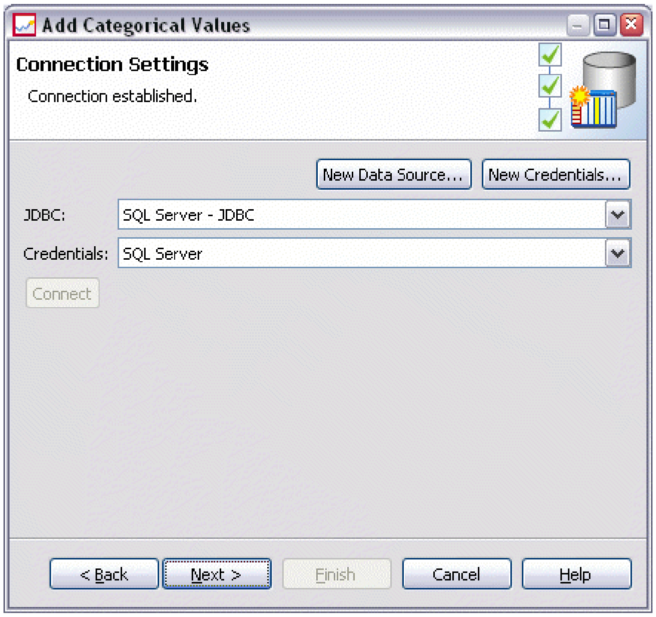
▶ Locate the appropriate *.csv* file and click Open. You are returned to the Add Categorical Values dialog box, and the path for the selected *.csv* file displays in the Import from file field.

▶ Click Finish to add every unique value in the *.csv* file to the Categorical Values list.

Import from Data Source. Select this option to import categorical values from an existing data source table column.

▶ Click Next to display the Connection Settings step.

Figure 8-12
Enterprise View Connection Settings step



Add Categorical Values

Connection Settings

Connection established.

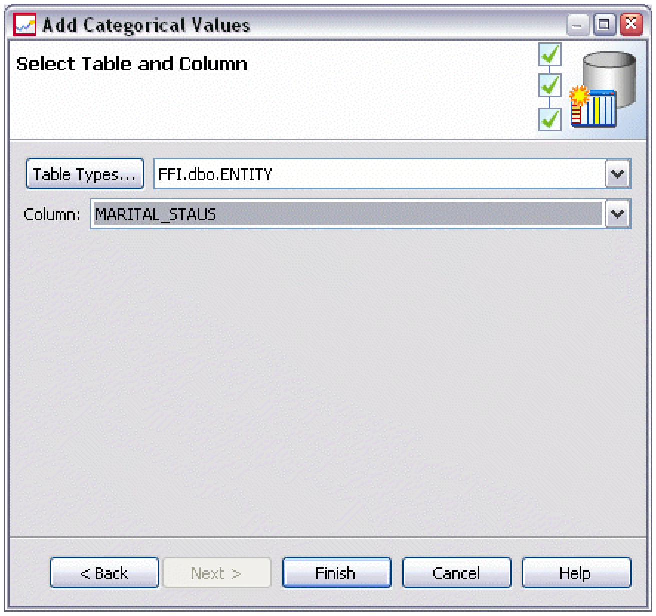
JDBC: SQL Server - JDBC

Credentials: SQL Server

- Select the appropriate data source JDBC and Credentials entries, click Connect, and then click Next to display the Select Table and Column step.

Note: You may need to create new data source and credential definitions if they do not already exist.

Figure 8-13
Enterprise View Select Table and Column step



Add Categorical Values

Select Table and Column

Table Types...: FFI.dbo.ENTITY

Column: MARITAL_STAUS

- Click Finish to add the categorical values, for the selected column, to the Categorical Values list.

▶ Click Finish to return to the Enterprise View Editor. The selected column now displays the categorical values you defined.

Figure 8-14
Enterprise View Defined Categories for MARITAL_STATUS column

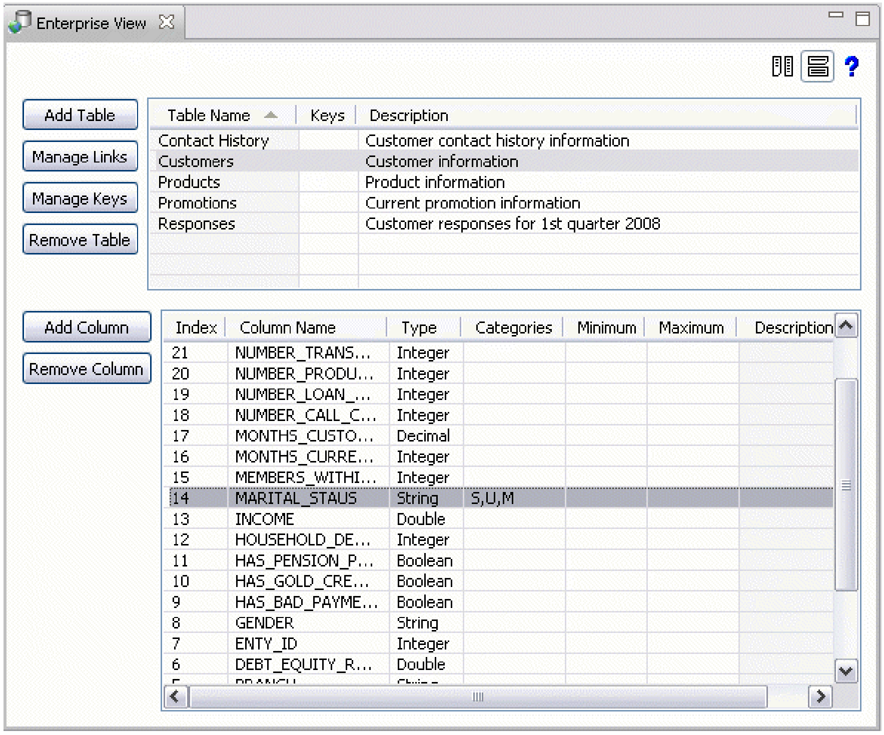


Table Name	Keys	Description
Contact History		Customer contact history information
Customers		Customer information
Products		Product information
Promotions		Current promotion information
Responses		Customer responses for 1st quarter 2008

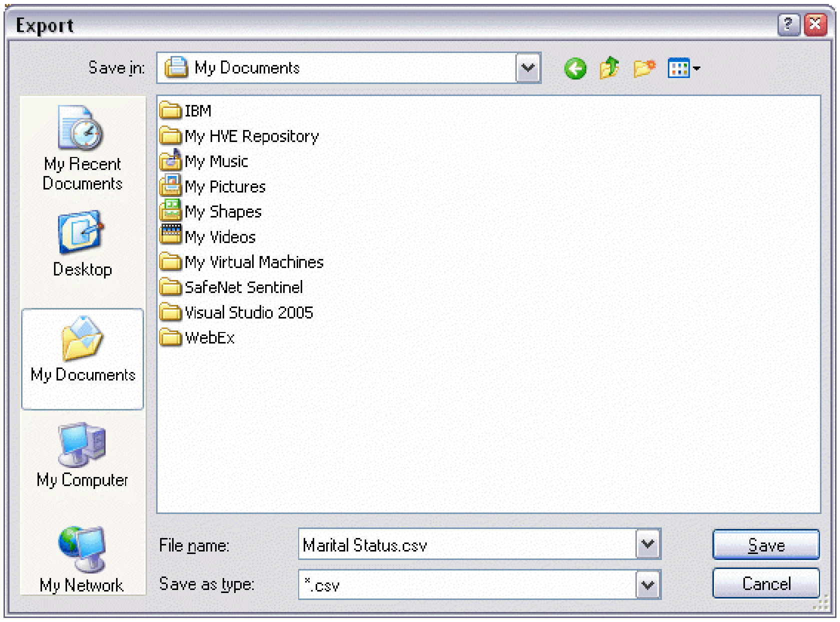
Index	Column Name	Type	Categories	Minimum	Maximum	Description
21	NUMBER_TRANS...	Integer				
20	NUMBER_PRODU...	Integer				
19	NUMBER_LOAN_...	Integer				
18	NUMBER_CALL_C...	Integer				
17	MONTHS_CUSTO...	Decimal				
16	MONTHS_CURRE...	Integer				
15	MEMBERS_WITHI...	Integer				
14	MARITAL_STAUS	String	S,U,M			
13	INCOME	Double				
12	HOUSEHOLD_DE...	Integer				
11	HAS_PENSION_P...	Boolean				
10	HAS_GOLD_CRE...	Boolean				
9	HAS_BAD_PAYME...	Boolean				
8	GENDER	String				
7	ENTY_ID	Integer				
6	DEBT_EQUITY_R...	Double				

Exporting Categorical Values

The Categorical Values dialog box allows you to export your defined categorical values to a *.csv* file. To export your defined categorical values:

▶ Click Export. The Export dialog box displays.

Figure 8-15
Enterprise View Export dialog box



Save in: My Documents

File name: Marital Status.csv

Save as type: *.csv

- ▶ Navigate to an appropriate save location.
- ▶ Enter a suitable filename for the *.csv* file.
- ▶ Click Save.

  Note: The export process saves all defined categorical values, regardless if they are all selected in the Categorical Values dialog box.
- ▶ Click OK to return to the Enterprise View Editor.

Removing Categorical Values

To remove existing categorical values:

- ▶ Select the appropriate values from the Categorical Values dialog box.
- ▶ Click Remove.
- ▶ Click OK to return to the Enterprise View Editor.

Minimum/Maximum

The minimum/maximum attributes function as user aids by providing defined limits for each column. For example, for the column *INCOME*, you could specify a minimum value of 20000 and a maximum value of 75000.

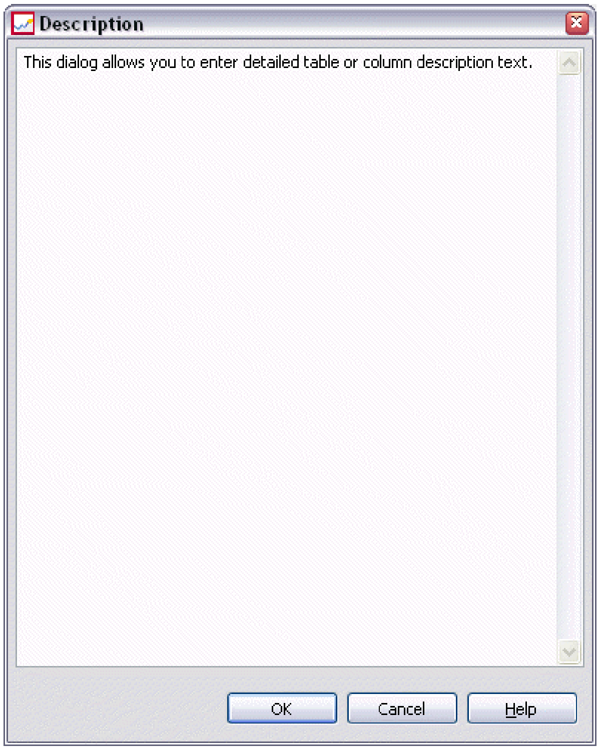
Both attributes can be edited by double-clicking in the desired cell and entering text manually.

Description

Table and column descriptions can be added or edited during the Add Table or Add Column tasks or by double-clicking within one of the *Description* column cells.

The Enterprise View Table/Column Description dialog box allows you to enter detailed table/column description text. The Enterprise View description is intended to provide high-level table and column information. The text defined here is also displayed (read-only) in the Application View Description dialog box.

Figure 8-16
Enterprise View Table/Column Description dialog box



This dialog allows you to enter detailed table or column description text.

Manage Keys

The Manage Keys dialog box provides options for viewing and defining keys for each Enterprise View table. A **key** is a set of columns that can be used to identify or access a particular row or rows. The key is identified in the description of a table, index, or referential constraint. The same column can be part of more than one key.

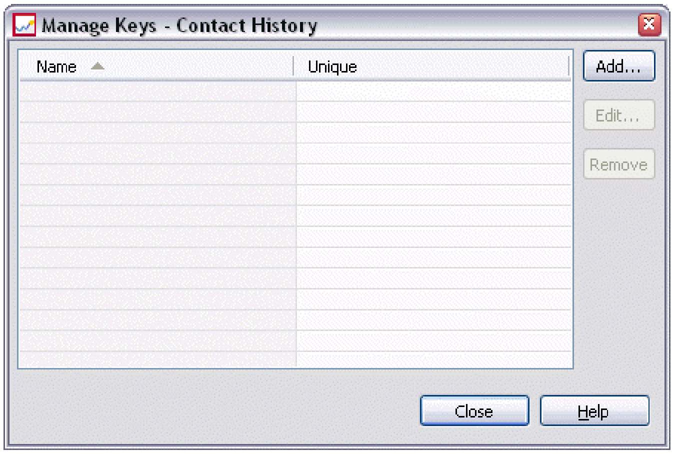
Note: A key can be based on more than one column. Rows in two tables match if the values of all columns in the table keys match each other, in the same order.

A unique key is a key that is constrained so that no two of its values are equal. The columns of a unique key cannot contain NULL values. For example, an employee number column can be defined as a unique key, because each value in the column identifies only one employee. No two employees can have the same employee number.

Assigning a Key to an Enterprise View Table

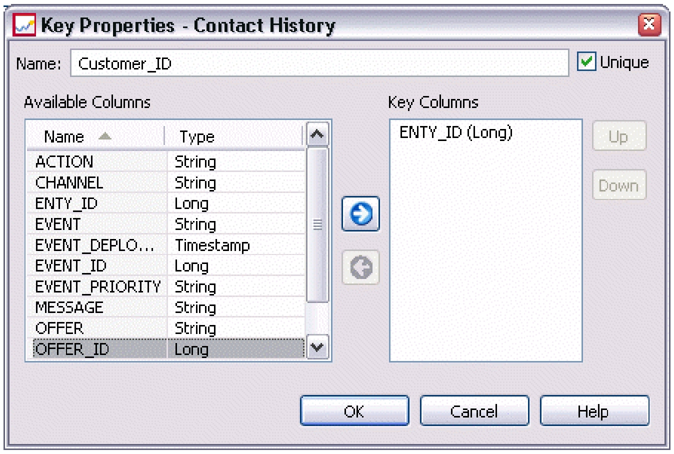
- ▶ Click Manage Keys in the Enterprise View Editor. The Manage Keys dialog box opens.

Figure 8-17
Manage Keys dialog box



- ▶ Click Add to define a new key. The Key Properties dialog box displays, allowing you to define or edit an existing key.

Figure 8-18
Key Properties dialog box



Name: Customer_ID ☑ Unique

Available Columns

Name	Type
ACTION	String
CHANNEL	String
ENTY_ID	Long
EVENT	String
EVENT_DEPLO...	Timestamp
EVENT_ID	Long
EVENT_PRIORITY	String
MESSAGE	String
OFFER	String
OFFER_ID	Long

Key Columns

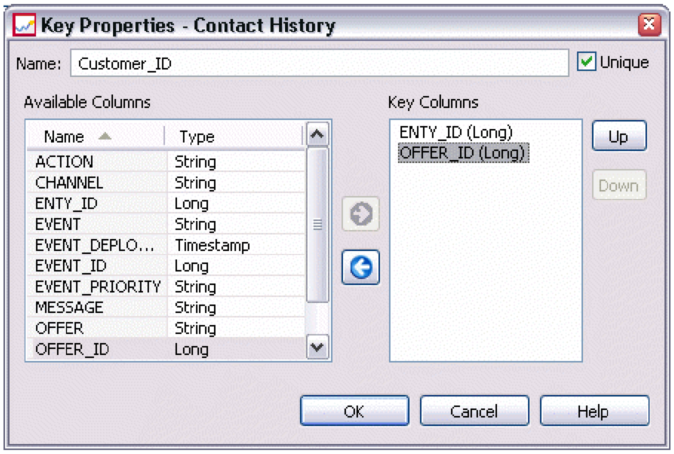
ENTY_ID (Long)

- ▶ Enter an appropriate key name in the Name field and click Unique if the key is unique. This name appears under the *Keys* column in the Enterprise View Editor and is the name a consuming application will use to determine the key.

- ▶ Select the appropriate columns from the Available Columns table and click the right arrow. The selected column names are displayed in the Key Columns list. Derived columns cannot be used as keys.

  You can remove existing columns from the Key Columns list by selecting the appropriate columns and clicking the left arrow.

Figure 8-19
Key Properties dialog box



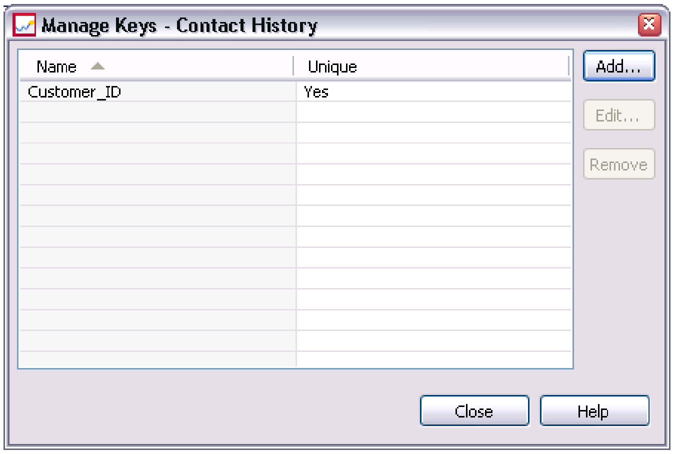
Name: Customer_ID ☑ Unique

Name	Type
ACTION	String
CHANNEL	String
ENTY_ID	Long
EVENT	String
EVENT_DEPLO...	Timestamp
EVENT_ID	Long
EVENT_PRIORITY	String
MESSAGE	String
OFFER	String
OFFER_ID	Long

ENTY_ID (Long)
OFFER_ID (Long)

- ▶ Click the Up and Down buttons to rearrange the key column order.
- ▶ Click OK to save the key definition. The Key Properties dialog box closes, and the key name is displayed in the Manage Keys dialog.

Figure 8-20
Manage Keys dialog box



Name	Unique
Customer_ID	Yes

- ▶ Click Close to return to the Enterprise View Editor. The selected table displays the number of currently defined keys under the *Keys* column.

Manage Links

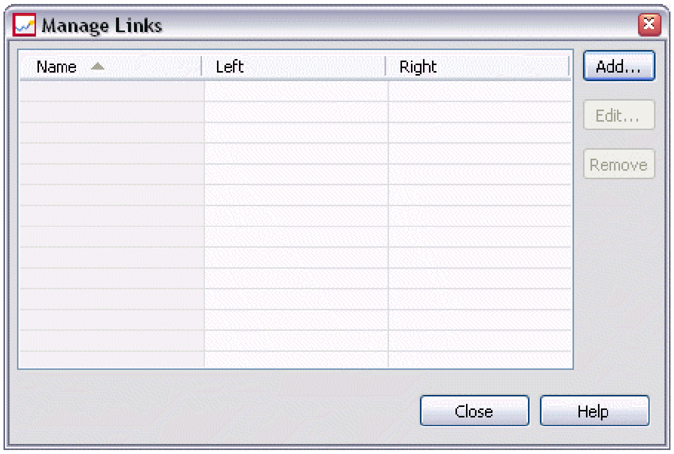
The Manage Links dialog box provides options for viewing and defining links between tables and data sets.

Links are similar in function to SQL joins in that they allow you to utilize data from more than one data source table or data set. Tables and data sets can be related to each other with keys. A key is a column that contains a value for each row. Unique keys are defined for columns that contain unique values for each row (customer identification numbers for example). Refer to the Manage Keys section for more information on keys.

Defining a Link Between Two Tables or Data Sets

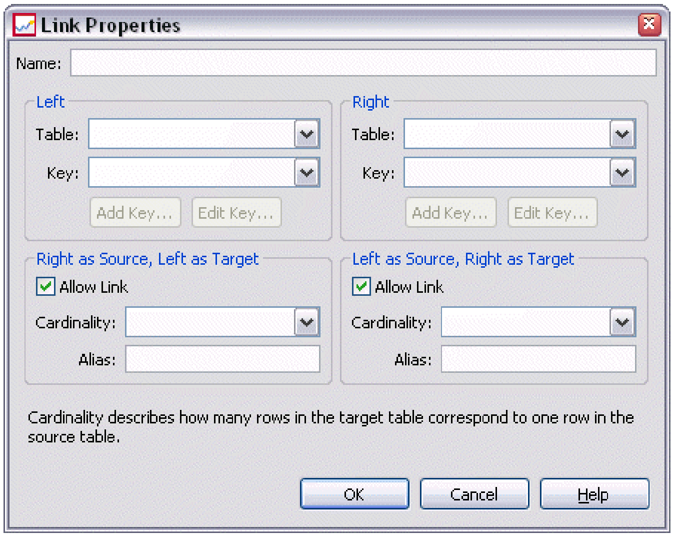
▶ Click Manage Links. The Manage Links dialog box opens.

Figure 8-21
Manage Links dialog box



▶ Click Add to define a new link. The Link Properties dialog box displays, allowing you to define a new link.

Figure 8-22
Link Properties dialog box



Right as Source, Left as Target: ☑ Allow Link

Left as Source, Right as Target: ☑ Allow Link

Cardinality describes how many rows in the target table correspond to one row in the source table.

▶ Enter an appropriate link name in the Name field. This name appears under the *Name* column in the Manage Links dialog and is the name a consuming application will use to determine the link.

- ▶ Select the appropriate table or data set and key under the Left section. The Tables/Data Set drop down menu provides a list of all defined tables or data sets. The Key drop-down menu provides a list of all keys currently defined for the selected table or data set.

- ▶ Select the appropriate table or data set and key under the Right section. The Tables/Data Set drop down menu provides a list of all defined tables or data sets. The Key drop-down menu provides a list of all keys currently defined for the selected table or data set.

  When matching on keys of the decimal type, the two keys must have the same scale. For example, the SQL Server *money* type is converted to a decimal with a scale of 4. A column of this type can only be linked to another column of the decimal type with a scale of 4.

  The selected keys must have the same number of matching column types. For example, if the Left table's key contains five INTEGER column types, the Right table's key must also contain five INTEGER column types. Refer to the Manage Keys section for more information on keys. If editing the keys used in a link results in incompatibilities between the keys, the link using those keys is removed from the system when you close the Link Properties dialog.

- ▶ Select the appropriate Allow Link check boxes.
  - ■ When both boxes are checked, the link maps freely between both tables/data sets.
  - ■ When only Right as Source, Left as Target is checked, a single row in the Right table/data set maps to a row or rows in the Left table/data set.
  - ■ When only Left as Source, Right as Target is checked, a single row in the Left table/data set maps to a row or rows in the Right table/data set.
  - ■ When neither checkbox is checked, the link is deactivated.

- ▶ Select the appropriate cardinality for both the Left and Right table/data set selections. The cardinality settings define the row relationships between the two selected tables/data sets. It is assumed you are starting from one row in the source table/data set, and defining how many rows in the target table/data set to map.

  For example, assume:

  - ■ Left table/data set is defined as CUSTOMERS.
  - ■ Right table/data set is defined as CITY.
  - ■ Left as Source, Right as Target is defined as Exactly One (1..1).
  - ■ Right as Source, Left as Target is defined as Zero or More (0..*).
  - ■ The Allow Link check box is selected for both tables/data sets.

  Given the above example, the following rules will be enforced:

  - ■ Left as Source, Right as Target. One customer maps to one city. Each customer must be registered in exactly one city.
  - ■ Right as Source, Left as Target. One city maps to any number of customers. Each city may have any number of customers, including none.

  The values displayed in the Cardinality drop-down menu will vary depending on whether or not the selected key in unique.

When the selected key in unique, you are presented with the following Cardinality options:

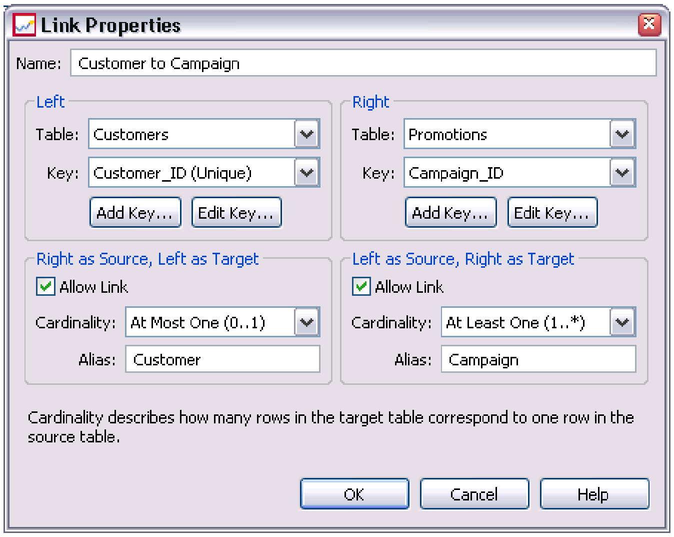
- At Most One (0..1)
- Exactly One (1..1)

When the selected key in not unique, you are presented with the following Cardinality options:

- Zero or More (0..*)
- At Most One (0..1)
- Exactly One (1..1)
- At Least One (1..*)

▶ Enter an appropriate alias for each link table/data set (optional). The alias defines a unique name for each link table/data set. Without the alias, the link name is used for each linked table/data set.

Figure 8-23
Link Properties dialog box



Name: Customer to Campaign

Left
Table: Customers
Key: Customer_ID (Unique)

Right
Table: Promotions
Key: Campaign_ID

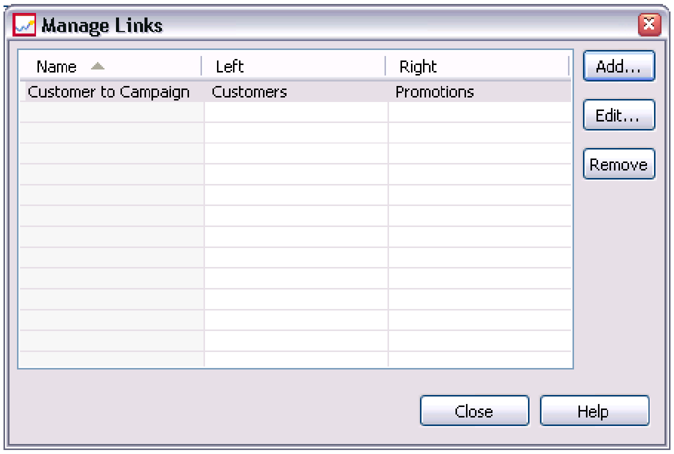
Right as Source, Left as Target
☑ Allow Link
Cardinality: At Most One (0..1)
Alias: Customer

Left as Source, Right as Target
☑ Allow Link
Cardinality: At Least One (1..*)
Alias: Campaign

Cardinality describes how many rows in the target table correspond to one row in the source table.

▶ Click OK to save the link definition. The Link Properties dialog box closes, and the link name is displayed in the Manage Links dialog.

Figure 8-24
Manage Links dialog box



Name	Left	Right
Customer to Campaign	Customers	Promotions

- ▶ Click Close to return to the editor.

Saving the Enterprise View

Saving the Enterprise View always creates a new version of the Enterprise View in the enterprise repository. It is not possible to overwrite previous versions. Object versions are handled by the built-in enterprise repository functionality.

To save a new version of the Enterprise View:

- ▶ Click the save icon from the Deployment Manager toolbar

or

- ▶ Choose:

File
  Save

All saved changes to the Enterprise View are logged and include the ID of the user who saved the changes, the Enterprise View version ID, and a timestamp reflecting when the changes were saved.

Application View

An Application View collects a subset of tables and columns from the Enterprise View that relate to a specific application. It captures additional information about how tables and columns are used in the context of an application (for example, call center or fraud detection). There can be more than one Application View associated with the Enterprise View.

The Application View serves two purposes. First, it provides a means to constrain the information displayed to a user in a tool or application based on the Enterprise View. For example, a user may be interested only in PredictiveCallCenter™ application information. By identifying which Enterprise View objects are used in the PredictiveCallCenter application, user interface logic can limit results to only those relevant objects. Second, the Application View provides a means for a system administrator or data expert to view the repository from the perspective of an application. This is especially useful in assessing the effect changes might have on a specific application.

Creating an Application View

- Right-click the *Content Repository* directory and choose:

  New
    Application View...

or

- Choose:

  File
    New
      Application View...

The New Application View dialog box opens, providing options for defining an Application View name and description. After entering an appropriate Application View name and description, click Finish to launch the Application View Editor.

Application View Editor

The Application View Editor allows the Data Expert to create, edit, review, and validate an Application View. An Application View is created from tables and columns defined in the Enterprise View definition. Consumers of an Application View, for example a PASW Modeler stream, can be linked to the Application View, facilitating impact analysis.

Adding Tables and Columns to the Application View

Each Application View references a specific Enterprise View version. The available tables and columns are limited to what the Enterprise View version contains.

To specify which tables and columns are included in the Application View:

- Select an appropriate table from the Application View Editor's table section. Once a table is selected, its corresponding columns display in the column section.
- Select columns by clicking the appropriate check boxes in the columns section.
- Repeat the above steps for each required table.

Removing Tables and Columns from the Application View

To remove tables and columns from the Application View:

- Select an appropriate table from the Application View Editor table section. Once a table is selected, its corresponding columns display in the column section.
- Remove columns by deselecting the appropriate check boxes in the columns section. If all columns for a particular table are deselected, the table is automatically deselected in the table section.
- Repeat the above steps for each required table.

Column Count

The **Column Count** column indicates how many columns are currently selected for each specified table. For example, when the CUSTOMER table lists a column count of 4/20, that means that 4 out of 20 possible columns are currently selected. The column count quickly lets you verify the column coverage for each table.

Sorting Columns

All columns can be sorted in ascending or descending order. To sort a column, click an appropriate column heading from the column section. The arrow displayed in the heading indicates whether the column is sorted in ascending or descending order.

Note: Sorting does not affect how columns are displayed for consuming applications. The index order dictates how columns are displayed for consuming applications. For more information, see the topic Index on p. 149..

Changing Application View Column Attributes

Column attributes can be changed by simply double-clicking in the desired column cell. You can also select multiple rows, right click, and apply the same attribute changes to all selected rows (for example, you can apply the same **Environment** to all selected rows).

Note: Changing row attributes via multi-row selection erases the Deployment Manager undo/redo history.

Not all Application View columns are user definable. For example, the Column Name and Type columns are defined in the Enterprise View Editor and cannot be modified in the Application View Editor.

- **Index**: The Index column represents the order in which Enterprise View columns will be presented to consuming applications. For more information, see the topic Index on p. 149.
- **Column Name**: Displays the current column name. This attribute cannot be modified in the Application View Editor. For more information, see the topic Column Name on p. 149.

- **Type**: Specifies the type of data stored in a column. Each column can store data consisting of only a single data type. Refer to Type for information regarding supported column types. This attribute cannot be modified in the Application View Editor.
- **Environment**: Displays the environments currently associated with the column. For more information, see the topic Environment on p. 150.
- **Direction**: Indicates wether the column is a predictor, a target, or both. For more information, see the topic Direction on p. 151.
- **Description**: Provides a user-defined column description. For more information, see the topic Description on p. 151.

Index

The column index order dictates how columns are displayed for consuming applications. In the following table, the columns will be displayed to consuming applications in the order *Income*, *State*, *Ownership*, *Mortgage*, and *Second Mortgage*.

Index	Column name
4	Mortgage
2	State
5	Second Mortgage
3	Ownership
1	Income

By default, the index order is determined by the order in which the columns were defined in the Enterprise View. The index order can be edited manually at the Enterprise View or the Application View level.

Note: When the index order is changed at the Application View level, it does not affect the original Enterprise View indexing order.

Column Name

The column name attribute displays the current column name inherited from the Enterprise View.

Type

The column type attribute is inherited from the Enterprise View and specifies the type of data stored in a column. Each column can store data consisting of only a single data type. For more information, see the topic Enterprise View Data Types on p. 207.

Table 8-2
Data types

Type	Description
Boolean	Used to store binary digits (0 or 1). These can be used to denote Boolean values, such as Yes/No, True/False, or On/Off.

Type	Description
Date	Values must have the form CCYY-MM-DD, where CCYY denotes the year, MM denotes the number of the month (January = 1, February = 2, and so on), and DD denotes the day (starting with 1). Optionally, the year can be preceded by + or –. Leading zeros for the year, month, and day are optional. Spaces are not allowed before or after the – separator.
Daytime	Values must have the form HH:MM:SS.xxx, where HH denotes the hours (ranging from 0 to 23), MM denotes the minutes (0 to 59), SS denotes the seconds (0 to 59), and xxx, the milliseconds (0 to 999). First, note that only the 24-hour representation is allowed. Furthermore, the seconds and milliseconds are optional. Leading zeros for the hours, minutes, and seconds are optional. Spaces are not allowed before or after the : separator.
Decimal	Used to store exact numbers. The scale and maximum precision can be configured. The precision denotes the maximum number of digits each value may consist of. The scale indicates how many of those digits may occur to the right of the decimal point (that is, the fractional part). The value of this parameter must be greater than or equal to zero and less than or equal to the precision.
Double	Represents signed, approximate numerical values. Double permits much larger numbers than Float.
Float	Represents signed, approximate numerical values. Double permits much larger numbers than Float.
Integer	Used for whole numbers (for example, 5 or 110).
Long	Used for character data of variable length up to two gigabytes.
String	Used to store a sequence of ISO-8859-Latin-1 characters. The maximum length of such a sequence is configurable. Furthermore, the string may be empty and cannot contain the newline (ASCII 10) or null (ASCII 0) characters.
Timestamp	Allows values of the form CCYY-MM-DDTHH:MM:SS.xxx in accordance with the ISO 8601 standard. It is easy to see that it is nothing more than the concatenation of the Date and the Time types, using a delimiter of *T* to indicate the beginning of the Time portion. Consequently, the rules for the Date and the Time types are also applicable to the Timestamp type.

Environment

The environment settings provide a means of identifying which particular columns should be associated with defined business segments. The business segments include:

- **Analytical.** Columns associated with an Analytic environment typically define the transaction data needed to run analytic and optimization tasks.
- **Operational.** Columns associated with an Operational environment are typically used to deploy the resultant analytics to operational channels.
- **Reporting.** Columns associated with an Reporting environment are typically used to provide the resultant analytics to reporting channels.

Columns can be defined as belonging to multiple environments. For example, the column CUST_ID can be defined to both the **Analytical** and **Operational** roles.

Note: By default, the Application View Editor lists all new columns as belonging to all three environments.

Direction

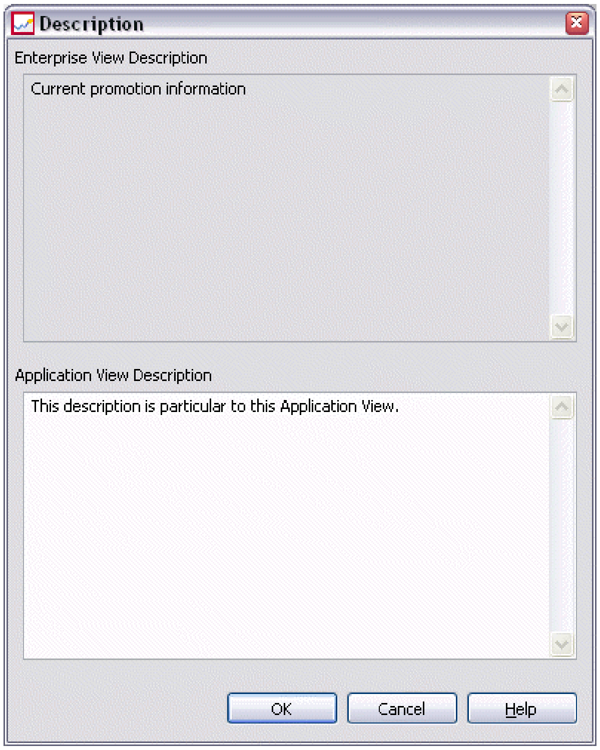
The direction setting specifies whether columns should be treated as predictors or targets. Predictors are used as an input to machine learning (predictor fields); targets are used as output for machine learning (predicted fields). The direction setting does not affect how columns are used or presented. The setting serves as a aid, allowing users to ascertain each column's intended role. PASW Modeler is one example of an application that utilizes the direction setting.

Description

Table and column descriptions can be added or edited by double-clicking inside one of the description column cells.

The Application View Description dialog displays any existing Enterprise View description information (read-only) and allows you to enter more detailed table/column description information that is specific to the Application View.

Figure 8-25
Enterprise View Table/ Column Description Dialog



Current promotion information

This description is particular to this Application View.

Saving the Application View

You can save a new Application View to the repository or modify and save an existing Application View. Saving an existing Application View always creates a new version of the Application View in the repository. It is not possible to overwrite previous versions. Object versions are handled by the built-in repository functionality.

To save a new version of the Application View:

▶ Click the save icon from the Deployment Manager toolbar

or

▶ Select

File
  Save or Save As...

All saved changes to the Application View are logged and include the ID of the user who saved the changes, the Application View version ID, and a timestamp reflecting when the changes were saved.

Changing the Reference Enterprise View Labeled Version

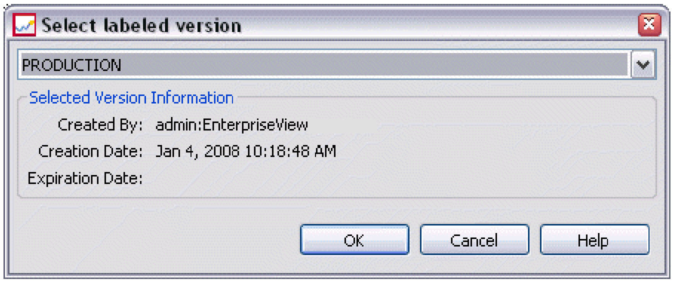
By default, a new Application View uses the Enterprise View version labeled **LATEST** for defining table and column information. An existing Application View will first attempt to match its label with an equivalent Enterprise View label. If no label match is found, the Application View editor defaults to the Enterprise View version with the **LATEST** label.

Note: It is possible to associate another label with the **LATEST** version. For example, you could apply the label **PRODUCTION** to the latest Enterprise View version. In this case the most recent version contains both the **LATEST** and **PRODUCTION** labels.

In some instances you may not want to use the most current Enterprise View version.

▶ Click Change to select a different Enterprise View labeled version on which the Application View is based.

Figure 8-26
Select Labeled Version Dialog



PRODUCTION

Selected Version Information

Created By: admin:EnterpriseView

Creation Date: Jan 4, 2008 10:18:48 AM

Expiration Date:

Note: Changing to a different labeled version is only effective for the current editing session. If you want to permanently change the reference Enterprise View version you must ensure the Application View version label matches the Enterprise View version label. For more information, see the topic Working With Labels in Chapter 5 on p. 46.

For more information, see the topic Working With Labels in Chapter 5 on p. 46.

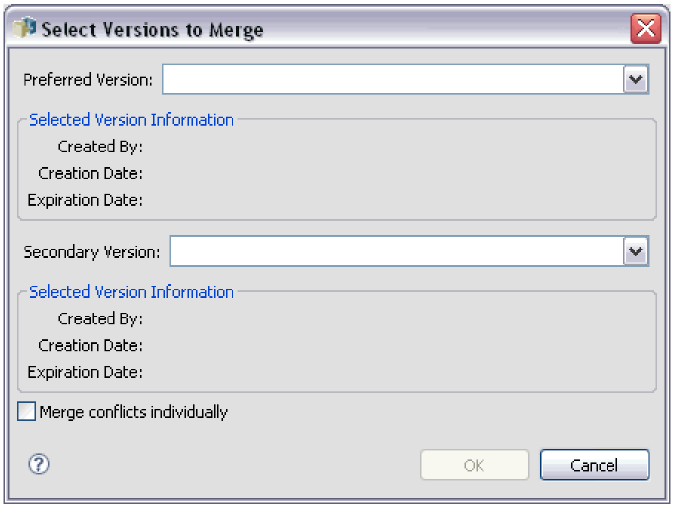
Merging View Versions

An Enterprise View or an Application View may have several different versions. Every saved change to an Enterprise View or Application View results in a new version of that view. Furthermore, when importing a view, all imported versions yield new versions of the corresponding target view.

Occasionally, information from two separate versions of a view is needed, such as a table from version *A* of a view and a table from version *B*. In this case, the versions can be merged into a new, composite version for the view. Note, however, that the merge involves two versions of the same view. Different views cannot be merged. For example, Application View*View1* cannot be merged with Application View*View2*, but version *A* of *View1* can be merged with version *B*.

To merge two versions of a view, right-click the view in the Content Explorer and select Merge Versions. The Select Versions to Merge dialog box appears.

Figure 8-27
Select Versions to Merge



Preferred version. Select the first version of the view to be merged. The tables, keys, links, and column information from the preferred version are preserved in the merged version, but table descriptions are not.

Secondary version. Select the version of the view to merge with the preferred version. Tables, keys, links, and columns from the secondary version are added to the preferred version. Merged tables contain everything from the preferred version with additional columns or metadata for existing columns from the secondary version. If a definition in the secondary version conflicts with a definition in the preferred version, the latter definition is used in the merged version by default. For example, if the preferred version defines the *Age* column of the *My_Table* table as a *String* and the secondary version defines *Age* as an *Integer*, the merged version defines *Age* as a *String*.

Merge conflicts individually. When selected, the resolution for each conflict occurring when merging the selected versions can be manually specified individually instead of allowing the preferred version to automatically take precedence. For more information, see the topic Resolve Conflicts on p. 154.

After selecting the two versions, click OK to perform the merge.

Usually, the merged version contains definitions from either one of the two versions being merged, depending on whether conflict resolution is automatic or manual. However, in some cases, the result is not a strict selection from one of the two versions.

- When merging categorical values of an Enterprise View column, the set of categories for the merged column is the case-insensitive union of the categories from the two selected versions. To specify one of the category sets instead of the union, merge conflicts individually.
- If the two selected versions each have a key with the same name on the same table but with different defined properties, the key from the secondary version is renamed.
- Links are updated to use any new key names. However if a link cannot be included due to a changed column type in one of the keys, the link is not included in the merged version.

Resolve Conflicts

The Resolve Conflicts dialog box identifies conflicts arising when merging two versions of an Enterprise View or an Application View and defines a resolution policy for each table containing conflicts.

Figure 8-28
Enterprise View Merge Conflicts

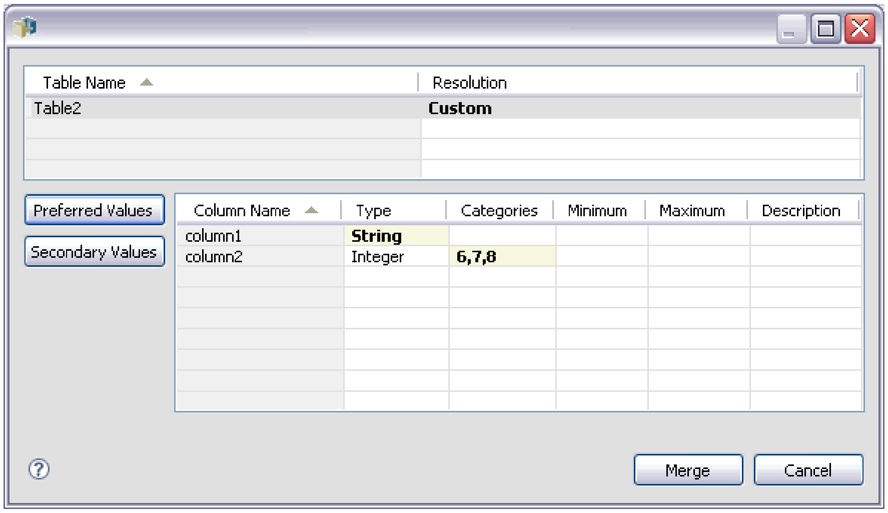


Table Name	Resolution
Table2	**Custom**

Column Name	Type	Categories	Minimum	Maximum	Description
column1	**String**				
column2	Integer	**6,7,8**			

The table list identifies the tables in the selected views that exhibit conflicts.

- **Table Name.** Name of the table resulting in conflicts.
- **Resolution.** Specifies the conflict resolution policy for the table. The policy can use the values from either selected version of the view or define custom settings. To define the policy, select the cell and identify the desired approach from the drop-down list.

For a selected table, the column list displays the properties for columns having conflicts, with the properties available depending on whether the merge involves an Enterprise View or an Application View. For resolution policies based on a specific view version, the property values cannot be modified as one version is simply overriding the other. However, for a custom resolution policy, the values for conflicting properties may be selected individually. The conflicting property value are highlighted to assist in identifying which values must be specified.

Preferred Values. For a custom resolution policy, sets all column properties to the values defined for the preferred version of the view.

Secondary Values. For a custom resolution policy, sets all column properties to the values defined for the secondary version of the view.

To specify a value for an individual column property in a custom resolution policy, select the cell containing the value and identify the desired value from the drop-down list.

After specifying the table resolution policies and any custom resolution values, click Merge to generate the new merged version of the view.

Data Provider Definition

The Data Provider Definition is used exclusively for retrieving data from relational data sources in non-real time applications. The Data Provider Definition manages the data at each stage by mapping the logical column definitions in the Application View to physical table columns in the customer database. This may involve a direct mapping of columns or SQL queries to select or derive columns and records as needed. For example, the historic data used to build the model may come from the warehouse, and the data that is being scored may come from the operational system. The Data Provider Definition also specifies the data source and credentials used to access data.

A Data Provider Definition links source data directly to an Application View and indirectly to Enterprise View tables and columns. A single Application View table may have more than one Data Provider Definition, each of which may point to multiple data sources.

Creating a Data Provider Definition

To create a Data Provider Definition:

▶ Right-click the Content Repository directory and select:

```
New
  Data Provider Definition...
```

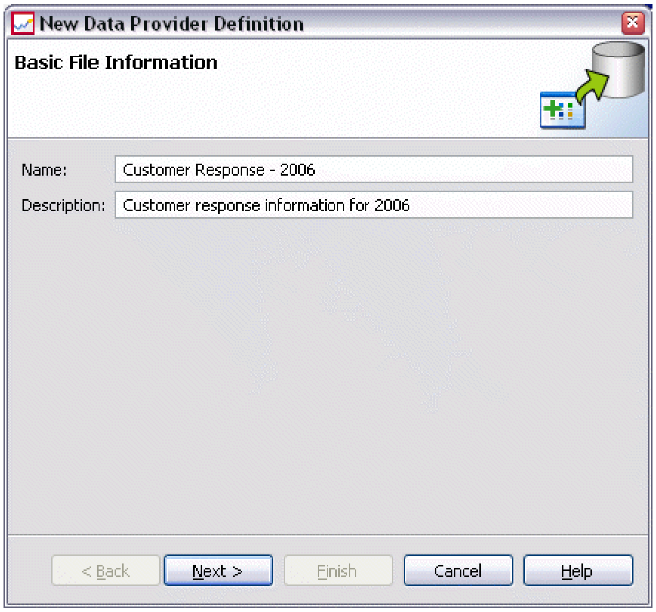
or

▶ Click the Content Repository folder and select:

```
File
  New
    Data Provider Definition...
```

The **New Data Provider Definition** wizard appears, providing options for defining a Data Provider Definition.

Figure 8-29
New Data Provider Definition Wizard



New Data Provider Definition

Basic File Information

Name: Customer Response - 2006

Description: Customer response information for 2006

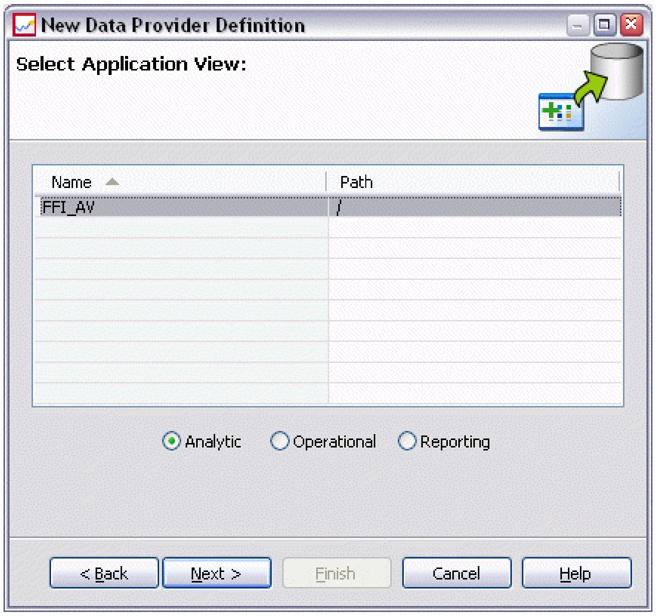
▶ After entering an appropriate Data Provider Definition name and description, click Next to Select Application View.

Note: The Name and Description fields are both limited to 256 characters.

Select Application View

The Select Application View dialog provides options for associating the Data Provider Definition with an appropriate Application View. The dialog only lists views that currently exist in the repository.

Figure 8-30
Select Application View Dialog



New Data Provider Definition

Select Application View:

Name	Path
FFI_AV	/

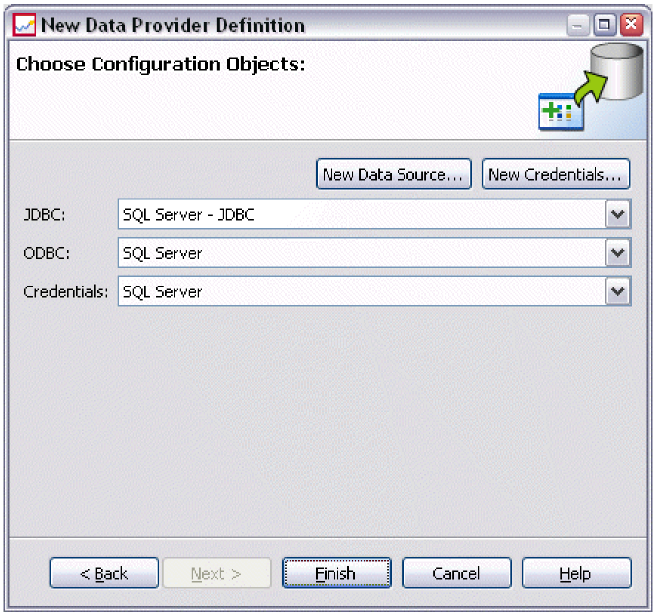
● Analytic ○ Operational ○ Reporting

- ▶ Select an Application View and select the appropriate environment (Analytic, Operational, or Reporting). For more information, see the topic Environment on p. 150.
- ▶ Click Next to display the Choose Configuration Objects dialog.

Choose Configuration Objects

The Choose Configuration Objects dialog provides options for selecting defined JDBC/ODBC data source definitions and server credentials. The dialog also provides options for creating new data source definitions and server credentials. Refer to Data Source Definitions and Credential Definitions for more information.

Figure 8-31
Choose Configuration Objects Dialog



JDBC: SQL Server - JDBC

ODBC: SQL Server

Credentials: SQL Server

- ▶ After selecting the appropriate data source definitions and credentials, click Finish. The new definition displays in the Data Provider Definition Editor. For more information, see the topic Data Provider Definition Editor on p. 158.

Data Provider Definition Editor

The Data Provider Definition Editor allows the Data Expert to create, edit, review, and validate a Data Provider Definition. A Data Provider Definition maps the column definitions in the Application View to physical tables in the customer database.

Changing the Reference Application View Labeled Version

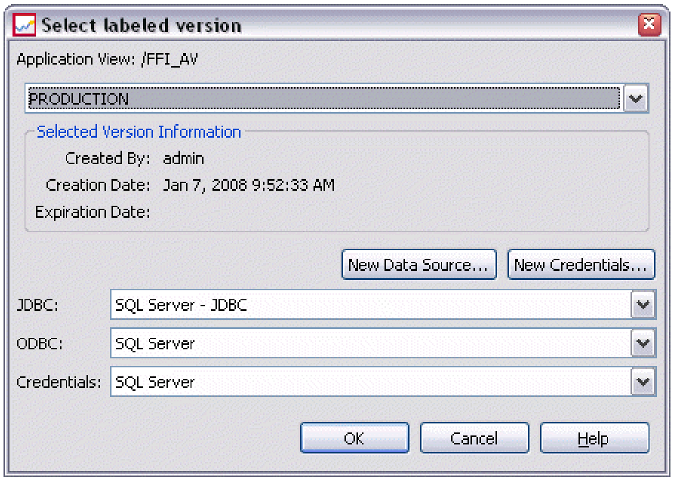
By default, a new Data Provider Definition uses the Application View version labeled **LATEST**. An existing Data Provider Definition will first attempt to match its label with an equivalent Application View label. If no label match is found, the Data Provider Definition editor defaults to the Application View version with the **LATEST** label.

Note: It is possible to associate another label with the **LATEST** version. For example, you could apply the label **PRODUCTION** to the latest Application View version. In this case the most recent version contains both the **LATEST** and **PRODUCTION** labels.

In some instances you may not want to use the most current Application View labeled version.

▶ Click Change to select a different, labeled Application View version.

Figure 8-32
Select Labeled Version Dialog



Select labeled version

Application View: /FFI_AV

PRODUCTION

Selected Version Information

Created By: admin

Creation Date: Jan 7, 2008 9:52:33 AM

Expiration Date:

JDBC: SQL Server - JDBC

ODBC: SQL Server

Credentials: SQL Server

For more information, see the topic Working With Labels in Chapter 5 on p. 46.

Mapping Column Definitions

The following steps explain the process of mapping the logical Application View tables and columns to physical data source tables and columns.

▶ Click on a table row that does not yet have mappings defined (the Mappings column is empty) to launch the Data Provider Table Specification dialog.

Note: You can also click inside a defined Mapping column cell. An ellipses button appears in the cell. Clicking the button launches the Data Provider Specification dialog.

The Data Provider Specification dialog provides options for selecting an existing data source table/view or manually requesting column and table information via an SQL statement.

▶ When you select Existing Table, the drop-down field becomes active, allowing you to select an existing data source table or view:

After the appropriate table/view is selected, you can verify column information by clicking Sample. The Sample Data dialog displays all columns from the selected table/view as well as a fifty-row sampling of actual data.

▶ Click OK to return to the Data Provider Definition Editor. All columns from the selected table/view now display in the Physical Columns section.

Note: Only columns from a single table are returned when the Existing Table option is selected.

The Data Provider Definition Editor will automatically map physical columns whose Column Name and Type match their logical column counterparts.

- ▶ When you select User Defined Select Statement, the edit box becomes active, allowing you to enter an SQL select statement. The select statement allows you to specify columns from more than one table/view or define various filtering attributes via a WHERE clause (WHERE GENDER = MALE, for example). The where clause information is not used in the Enterprise View, but is applied at run-time.

Note: The select statement must adhere to the syntax rules of the physical data source.

- ▶ After entering an appropriate select statement, click OK to return to the Data Provider Definition Editor. Only those columns defined in the select statement display, in the Physical Columns section, for the specified table(s).

The Data Provider Definition Editor will automatically map physical columns whose Column Name and Type match their logical column counterparts.

To Manually Map Columns

- ▶ Select a column name from the Physical Data Source Columns table.
- ▶ Select a corresponding column name from the Application View Columns table.
- ▶ Click Map. The Data Provider Definition Editor will map columns as long as their Type is the same.

Note: Alternatively you can map columns by dragging and dropping the Physical Data Source Columns into the Application View Columns table. Again, the Type must match between columns.

To Unmap Columns

- ▶ Select the appropriate columns from the Application View Columns table and click Unmap.

Note: Click Unmap All to remove the mappings for all columns.

Validating the Data Provider Definition

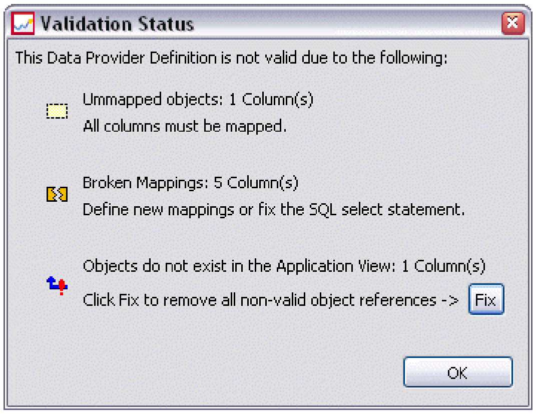
A Data Provider Definition is valid only when all tables and columns are properly mapped. The Data Provider Definition Editor provides the mapping status for each Application View table:

- A green check mark indicates that all logical table columns correctly mapped.
- A dotted rectangle indicates that not all logical table columns are mapped.
- A broken rectangle indicates that the logical table contains columns that were once mapped, but are now broken. Broken mapping errors typically occur when the SQL or data source, upon which the Application View's tables and columns are derived, is changed.
- A red exclamation with a blue arrow indicates that the logical column or table no longer exists in the Application View.

▶ Click Validate to check the mappings for all Application View tables. When all tables/columns are properly mapped, the Validation Status dialog indicates that the Data Provider Definition is valid.

When there is a problem with the column mappings, the Validation Status dialog will display one to three problems:

Figure 8-33
Validation Status Dialog for non-valid Data Provider Definition



Validation Status

This Data Provider Definition is not valid due to the following:

Ummapped objects: 1 Column(s)
All columns must be mapped.

Broken Mappings: 5 Column(s)
Define new mappings or fix the SQL select statement.

Objects do not exist in the Application View: 1 Column(s)
Click Fix to remove all non-valid object references ->

Unmapped Objects

Unmapped objects errors occurs when columns that exist in the Application View are not mapped in the Data Provider Definition. All Data Provider Definition column references must be mapped in order for the definition to be considered valid.

To resolve unmapped object errors:

- Ensure all Application View columns are properly mapped to the appropriate physical data source columns. For more information, see the topic Mapping Column Definitions on p. 159.

or

- Remove the unmapped column references from the associated Application View.

Broken Mappings

Broken mapping errors typically occur when the SQL or data source, upon which the Application View's tables and columns are derived, is changed. The Data Provider Definition mappings were correct at some point, but after changing the data source, the mappings are no longer valid.

To resolve broken mapping errors:

- Change the referenced data source back to the data source upon which the original mappings were established or ensure the new data source contains the same tables and columns as the previous data source.

Objects do not Exist in the Application View

Missing Application View object errors occur when columns are removed from the Application View after they have been mapped in the Data Provider Definition.

To resolve missing object errors:

- Restore the missing columns to the Application View. For more information, see the topic Adding Tables and Columns to the Application View on p. 147.

or

- Click Fix to automatically remove the column references from the Data Provider Definition.

Saving the Data Provider Definition

Saving the Data Provider Definition always creates in a new version of the Data Provider Definition in the enterprise repository. It is not possible to overwrite previous versions. Object versions are handled by the built-in enterprise repository functionality.

To save a new version of the Data Provider Definition:

▶ Click the save icon from the Deployment Manager toolbar

or

▶ Select

File
  Save or Save As...

All saved changes to the Data Provider Definition are logged and include the ID of the user who saved the changes, the Data Provider Definition version ID, and a timestamp reflecting when the changes were saved.

Data Provider Definition - Real Time

The Data Provider Definition - Real Time is used exclusively for real time interactions with applications. A real time Data Provider Definition links data sets directly to an Application View and indirectly to Enterprise View tables and columns. A single Application View table may have more than one associated real time Data Provider Definition, each of which may point to a different data source.

A data set represents a source of data. In its simplest form, a data set maps to a single data source table and abstracts the physical implementation of the data source. In addition to mapping to a data source table, a data set can map to other data sets. Using this mechanism, data from various data sources can be linked together. Refer to Manage Links on p. 142 for information on working with links.

The Data Provider Definition - Real Time retrieves data sets, one row at a time, from the following sources:

- JDBC data source defined in the Data Provider Definition - Real Time
- Application server defined data source
- User context data defined in the Data Provider Definition - Real Time
- User created data supplier (implemented externally)

Note: Additional data columns can be derived from existing columns.

Refer to Working with Data Source Definitions on p. 85 for more information.

Creating a Data Provider Definition - Real Time

To create a Data Provider Definition - Real Time:

▶ Right-click the Content Repository directory and select:

New
  Data Provider Definition - Real Time...

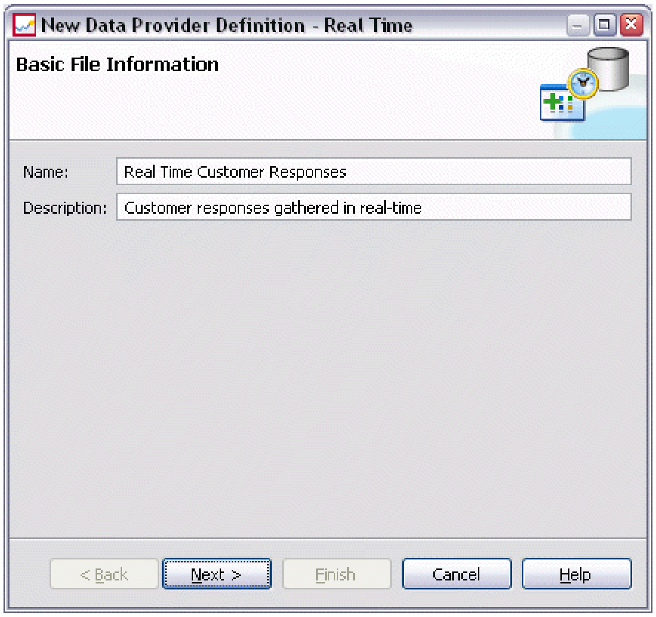
or

▶ Click the Content Repository folder and select:

File
  New
    Data Provider Definition - Real Time...

The **New Data Provider Definition - Real Time** wizard appears, providing options for defining a Data Provider Definition - Real Time.

Figure 8-34
New Data Provider Definition - Real Time Wizard



Name: Real Time Customer Responses

Description: Customer responses gathered in real-time

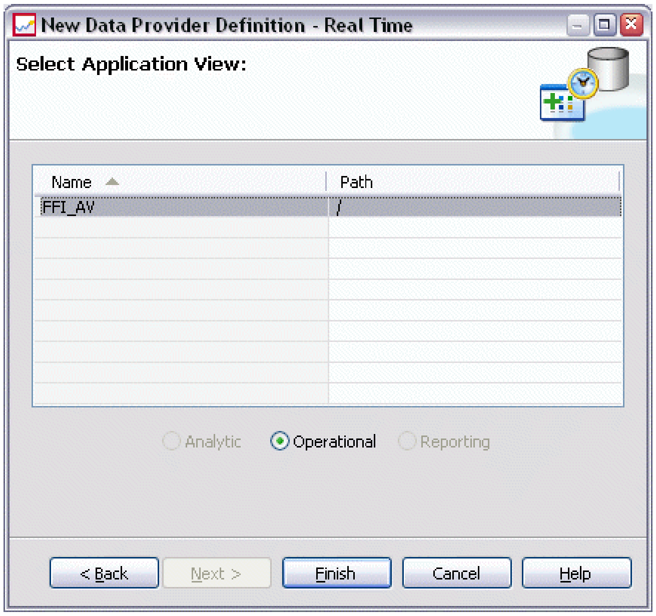
▶ After entering an appropriate Data Provider Definition - Real Time name and description, click Next to Select Application View.

Note: The Name and Description fields are both limited to 256 characters.

Select Application View

The Select Application View dialog provides options for associating the Data Provider Definition - Real Time with an appropriate Application View. The dialog only lists views that currently exist in the repository.

Figure 8-35
Select Application View Dialog



Name	Path
FFI_AV	/

○ Analytic ● Operational ○ Reporting

- ▶ Select an Application View.

  Note: Operational is the only available environment. For more information, see the topic Environment on p. 150.

- ▶ Click Finish. The Select Application View dialog closes and the Data Provider Definition - Real Time editor displays.

Data Provider Definition - Real Time Editor

The Data Provider Definition - Real Time Editor allows the Data Expert to create, edit, and review a Data Provider Definition designed for real-time interactions. A Data Provider Definition - Real Time maps the column definitions in a Application View to data sets derived from the following sources:

- JDBC data source defined in the Data Provider Definition - Real Time
- Application server defined data source
- User context data defined in the Data Provider Definition - Real Time
- User created data supplier (implemented externally)

Refer to Working with Data Source Definitions on p. 85 for more information.

The Data Provider Definition - Real Time Editor includes two tabs:

- **Mapping.** The Mapping tab displays the currently defined data set mappings and provides options for manually configuring mappings. For more information, see the topic Mapping Data Set Column Definitions on p. 166.
- **Data Sets.** The Data Sets tab displays the currently defined data sets and provides options for manually configuring data sets, managing links, and managing keys. For more information, see the topic Working with Data Sets on p. 168.

Mapping Data Set Column Definitions

The following steps explain the process of mapping the logical Application View tables and columns to real time data set tables and columns.

- ▶ Click on a table row that does not yet have mappings defined (the Mapping column is empty).
- ▶ Select an existing mapping from the drop-down menu in the selected table’s Mapping column.

Note: You will need to create a data set mapping if no mappings are currently defined. For more information, see the topic Working with Data Sets on p. 168.

The Data Set Columns table automatically populates with the column names and types defined for the selected data set mapping. The Data Provider Definition - Real Time editor automatically maps columns when both the Column Name and Type match.

Note: A data set can only be mapped to at most one Application View table.

To Manually Map Columns

- ▶ Select a column name from the Data Set Columns table.
- ▶ Select a corresponding column name from the Application View Columns table.
- ▶ Click Map. The Data Provider Definition - Real Time Editor will map columns as long as their Type is the same.

Note: Alternatively you can map columns by dragging and dropping the Data Set Columns into the Application View Columns table. Again, the Type must match between columns.

- ▶ Continue mapping until all columns in the Application View Columns table are properly mapped for each Application View table.

To Unmap Columns

- ▶ Select the appropriate columns from the Application View Columns table and click Unmap.

Note: Click Unmap All to remove the mappings for all columns.

Preview Data

You can select a table and verify column information by clicking Preview. The Preview Data dialog displays all columns from the selected table/view, and allows you to preview data for the selected data set, and add, edit, import, export, and remove context values for data sets of type context data.

▶ Click OK to return to the Data Provider Definition - Real Time Editor.

Preview Data

The Preview Data dialog provides options for previewing data for the selected data set, and adding, editing, importing, exporting, and removing context values for data sets of type context data.

Values for Key: The drop-down list provides available defined keys. Select an appropriate key. For some databases, the key type may impact the ability to preview the data.

- For SQL Server databases, the data cannot be previewed if the key is of type *text* or *ntext*.
- For DB2 databases, the data cannot be previewed if the key is of type *LONG VARCHAR*.
- For Netezza and Teradata databases, the data cannot be previewed if the key is of type *time* or *timestamp*.

Add: Opens the Value Editor dialog, allowing you to specify a new value for the selected key.

Edit: Opens the Value Editor dialog, allowing you to edit the selected key value.

Import: Opens the Import Values dialog, allowing you to import key values from a *.csv* file.

Export: Opens the Export Values dialog, allowing you to export key values to a *.csv* file.

Remove: Removes the selected key value.

Remove All: Removes all values for the selected key.

Context Values

The Context Values section allows you to add, edit, import, export or remove values for context data type data sources.

Add: Opens the Value Editor dialog, allowing you to specify a new value for the context data columns.

Edit: Opens the Value Editor dialog, allowing you to edit context data columns values.

Import: Opens the Import Values dialog, allowing you to import context data columns values from a *.csv* file.

Export: Opens the Export Values dialog, allowing you to export context data columns values to a *.csv* file.

Remove: Removes the selected context data column values.

Remove All: Removes all context data column values.

Get Data: Retrieves context data column values from the data source.

Working with Data Sets

The Data Sets tab provides options for defining, editing, and removing data sets.

A data set represents a source of data. In its simplest form, a data set maps to a single data source table and abstracts the physical implementation of the data source. In addition to mapping to a data source table, a data set can map to other data sets. Using this mechanism, data from various data sources can be linked together. Refer to Manage Links on p. 142 for information on working with links.

The Data Provider Definition - Real Time retrieves data sets, one row at a time, from the following sources:

- JDBC data source defined in the Data Provider Definition - Real Time
- Application server defined data source
- User context data defined in the Data Provider Definition - Real Time
- User created data supplier (implemented externally)

Note: Additional data columns can be derived from existing columns.

The system validates changes to the data source settings to determine if the modifications affect other definitions in the Data Provider Definition - Real Time. If the changes result in conflicts, such as missing necessary columns or incompatible data types, the modified data source cannot be applied to the data source. Other changes, such as the addition of columns, result in prompting for handling instructions. Refer to Working with Data Source Definitions on p. 85 for more information.

Data Sets

The Data Sets section provides options for defining data sets and working with data set links and keys.

Add.. Launches the Add Data set wizard, allowing you to define any number of data sets. For more information, see the topic Defining Data Sets on p. 169.

Manage Links. Opens the Manage Links dialog box, allowing you to create new data set links or work with existing links. For more information, see the topic Manage Links on p. 142.

Manage Keys. Opens the Manage Keys dialog box, allowing you to create new keys or work with existing keys. For more information, see the topic Manage Keys on p. 140.

Remove. Removes the currently selected data set.

Edit. Updates the data set to include new columns added to the underlying source and to swap one underlying source for another provided the new source provides the same set of base columns. Removing base columns is not allowed once they have been added. For more information, see the topic Editing Data Sets on p. 171.

Columns

The Columns section provides options for previewing data set column data and working with derived columns.

Preview. Launches the Preview Data dialog box, allowing you to preview data for the selected data set, and add, edit, import, export, and remove context values for data sets of type context data.

Add.. Launches the Create Derived Column expression editor, allowing you to manually create derived columns and join data sets. For more information, see the topic Expression Editor on p. 172.

Edit.. Launches the Edit Derived Column expression editor, allowing you to edit an existing derived column or data set join. For more information, see the topic Expression Editor on p. 172.

Remove. Removes a selected derived column from the Columns list.

Defining Data Sets

The following steps explain the process of defining a new data set.

▶ Click Add, under the Data Sets section, to launch the Add Data Set wizard.

The Add Data Set wizard's Select data source step allows you to define a JDBC, application server, or data service data source.

JDBC Data Source

The Data Provider Definition - Real Time Editor uses the JDBC metadata to automatically import the primary and foreign keys that are defined on the SQL table or view. Using this information, links are automatically created between data sets.

Note: The interaction server must be running in order to change anything related to the JDBC data source (selecting a different table for example).

▶ Select JDBC and click Next to display the Connection Settings step. For data sets that use a JDBC data source, the data set will map to a single SQL table (or view) within a SQL database.

Connection Settings for a JDBC Data Source

The Connection Settings dialog box provides options for selecting a JDBC data source provider and data source credentials. The dialog also provides options for creating new data sources and credentials. Refer to Selecting a Data Source Definition Type and Adding New Credentials for more information.

JDBC. Select an appropriate JDBC data source definition.

Credentials. Select an appropriate credential definition.

Connect. You must click Connect to verify the defined data source connection.

▶ Click Next to select tables and columns.

Select Tables and Columns for a JDBC Data Source

The Select Tables and Columns dialog box provides options for selecting an existing data source table and its associated column definitions.

Table Types... Click to select what to return from the physical data source.

- **System Table**—Returns data source system tables (for example, sysdevices and sysfiles).
- **Table**—Returns regular data source tables (for example, CUSTOMERS and ORDERS).
- **View**—Returns table views. Views are virtual tables whose contents are defined by a query. As with a real table, a view consists of a set of fields and records. The records and fields come from tables referenced in the query defining the view and are produced dynamically when the view is referenced.

Note: Other options may be available, depending on the data source configuration.

▶ Select the desired tables and columns and click Finish. You are returned to the Data Sets tab.

The newly defined data set, and its associated column definitions, are added to the Data Set table

Application Server Data Source

For data sets that use an Application Server data source, the data set will map to a single SQL table or view within a SQL database. You do not specify a credential when connecting to the data source. The credentials are defined in the application server.

The data source is available in the application server where the interaction server is running. As a result, the Data Provider Definition - Real Time editor connects to the interaction server in order to retrieve data source metadata. The Data Provider Definition - Real Time editor uses the metadata to automatically import the primary and foreign keys that are defined on the SQL table or view. Using this information, links are automatically created between data sets.

▶ Select Application Server and select the appropriate application server data source definition from the drop-down list.

Refer to Working with Data Source Definitions on p. 85 for more information.

▶ Click Next to display the Select Tables to Add step.

Data Service Data Source

For data sets that use a Data Services data source, the data set will map to a single table that is defined in the data service. The Data Provider Definition - Real Time editor retrieves data source metadata (columns, column types, and so on) using the information defined and stored as part of the repository object. No connection to an actual server is made.

Note: There can only be one Data Services data source of type **Context Data** per Data Provider Definition - Real Time.

▶ Select Data Service and select the appropriate real-time data service data source definition from the drop-down list.

Refer to Working with Data Source Definitions on p. 85 for more information.

▶ Click Next to display the Select Data Sets to Add step.

Select Data Sets to Add for a Data Service Data Source

The Add Data Set wizard's Select data source to add step allows you to select which data set to use from the previously selected real-time data service data source definition.

▶ Select the appropriate data sets and click Finish to return to the Data Sets tab.

The newly defined data set, and its associated column definitions, are added to the Data Set table

Editing Data Sets

To edit a data set:

1. Select the data set from the Data Sets section.
2. Click the Edit button.
3. Specify the data source settings. For more information, see the topicDefining Data Sets on p. 169. For JDBC Data Sources, specify the connection information.
4. Select the columns to include in the data set.

For JDBC and Application Server data sources, the Table Types button filters the list of available tables to selected types. Select a table from the list to view a list of the table columns. Each column will have one of the following states:

- **Missing.** The column appears in the data set being edited but the selected table does not contain a column having the same name.
- **Wrong type.** The column appears in both the data set and the selected table, but the types do not match.

- **Matched**. The column appears in both the data set and the selected table, and the type in each is identical. The column is selected for inclusion and cannot be unselected.
- **Additional.** The column appears only in the selected table but can be selected for inclusion in the data set.

If any columns have a state of *Missing* or *Wrong type*, the selected table cannot be used as the new data source.

After selecting columns, click the Finish button to commit the columns to the data set.

Saving the Data Provider Definition - Real Time

Saving the Data Provider Definition - Real Time always creates in a new version of the Data Provider Definition in the repository. It is not possible to overwrite previous versions. Object versions are handled by the built-in repository functionality.

To save a new version of the Data Provider Definition - Real Time:

▶ Click the save icon from the Deployment Manager toolbar

or

▶ Select

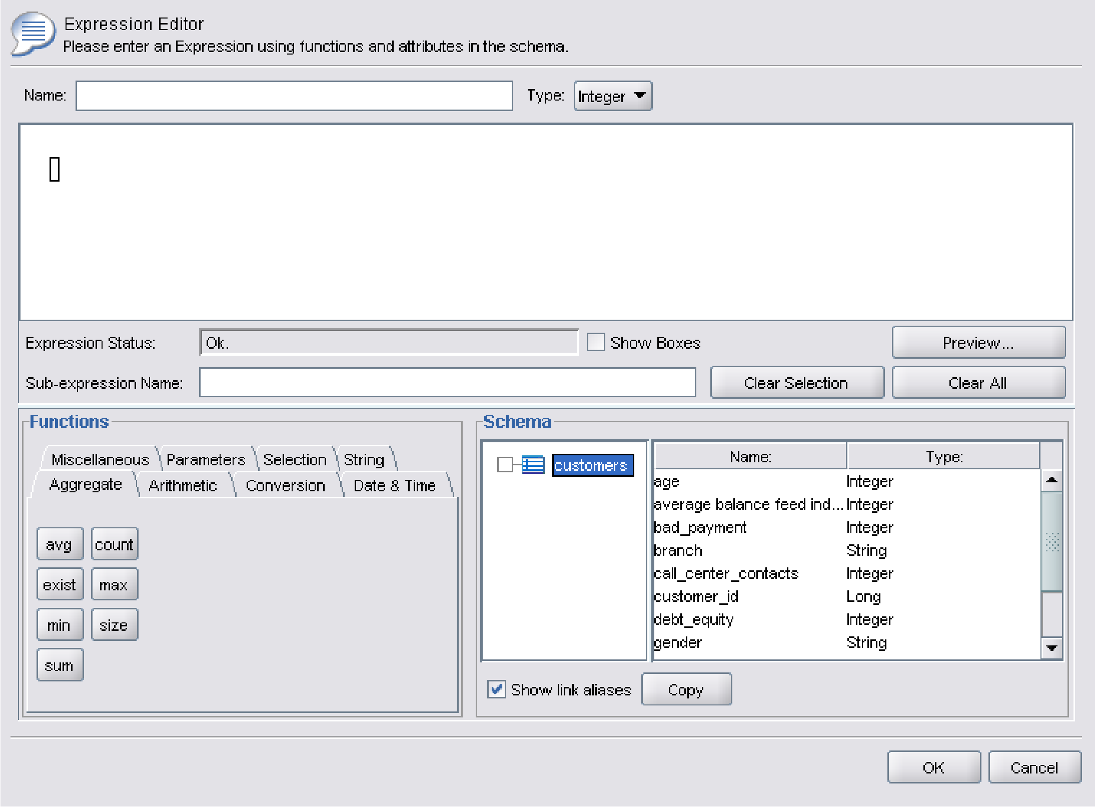
File
  Save or Save As...

All saved changes to the Data Provider Definition - Real Time are logged and include the ID of the user who saved the changes, the Data Provider Definition - Real Time version ID, and a timestamp reflecting when the changes were saved.

Expression Editor

The expression editor allows the creation of derived attributes for a Data Provider Definition used with the Enterprise View.

Figure 8-36
Expression Editor



Expression Editor
Please enter an Expression using functions and attributes in the schema.

Name:
Type: Integer

Expression Status: Ok.
□ Show Boxes

Sub-expression Name:

Name:	Type:
age	Integer
average balance feed ind...	Integer
bad_payment	Integer
branch	String
call_center_contacts	Integer
customer_id	Long
debt_equity	Integer
gender	String

☑ Show link aliases

The expression editor interface includes the following components:

- **Expression Panel.** Contains the expression canvas and controls for creating and editing expressions.
- **Schema Browser.** Allows you to browse through the schema and copy attributes to the expression.
- **Functions Panel.** Contains tabs with buttons for all available functions.

Expression Panel

The expression panel includes the following:

Name specifies the name of the expression. If creating a new expression, enter a unique name of your choice.

Type indicates the available return types based on the application. If creating a new expression, select a return type from the menu. The selected return type must match the return type of the expression created.

Expression Status indicates whether the syntax of the expression is valid. If the syntax is valid, a status message indicating the return type is displayed. If the syntax is not valid, an error message is displayed.

Show Boxes indicates whether sub-expressions should be outlined in boxes for easier viewing.

Sub-expression Name specifies the name of a sub-expression used in the expression.

Clear Selection deletes the selected item.

Clear All clears the expression canvas.

Preview displays the result of the expression.

Annotating Expressions

In the Sub-expression Name field, you can enter a name for a part of an expression. By giving one or more parts of an expression a useful name, complicated expressions can be made more easily readable because you can collapse and expand those named parts of expressions. The Show Boxes check box displays boxes in the expression canvas, illustrating which parts of the expression belong together.

Previewing an Expression

You can preview the result of an expression by clicking the Preview button. You can also preview the result of an expression or sub-expression by right-clicking it on the expression canvas and choosing Preview of selection. Always inspect the results of an expression before creating a new attribute.

Cutting, Copying, and Pasting an Expression

Expressions can be cut, copied, and pasted into a text editor or another expression editor session running on the same machine. Simply select the expression, right-click, and choose Cut or Copy from the pop-up menu. Then paste the expression into the desired application.

Note that when you paste an expression from expression editor into a text editor, that expression is displayed in XML format. To add the expression back into expression editor, you must copy the entire XML block and paste it onto the expression canvas. The expression editor converts the XML back into an easy-to-read string representation of the expression.

Functions Panel

Each tab on the functions panel contains a group of functions related to one another.

- The functions on the **Aggregate** tab can combine all the records of one table with the linked records from another table and return just one result, which can subsequently be used as the value for a new attribute in the first table. *Note:* Making use of aggregate functions in an expression can have a highly negative impact on the performance of its evaluation.
- The **Arithmetic** tab contains basic arithmetic functions you can use with numbers.
- The functions of the **Conversion** tab convert values from one data type to another.

- The **Date & Time** tab contains date and time functions.
- The **Miscellaneous** tab contains functions not offered on other tabs.
- The **Selection** tab contains functions to compare the values of expressions (like `or`, `between`, and `isnil`).
- The **String** tab contains string functions.

About Functions

SPSS Inc. functions perform many different kinds of operations. Functions are used to manipulate and prepare data. Thus, the raw data contained in tables (data sets) can be examined and extrapolated, providing more sophisticated information about customers and their attributes.

Before providing an overview of the different functions, this chapter explains:

- The different types of functions
- Data types associated with the functions
- Handling nil values
- Inspecting the results of an expression

Types of Functions

The expression editor includes two main types of functions: *basic* and *aggregate*.

Basic functions are functions used to derive results from the values of one or more attributes *from the same table*. For example, the string functions for string operations, the mathematical functions for mathematical operations, and the conversion functions to convert values into other types of values.

Aggregate functions are placed in the Aggregate tab. They can *combine* all records of one table with the linked records from another table and return one result. This result can be used as the value for a new attribute in the first table. For example, you can combine all records of the *PORDER* table that are linked to a specific record of the *Customer* table.

Data Types

The input and return types for each function are listed throughout this guide. You can also hover over a function in the expression canvas to display a popup message indicating the valid data types. If you create an expression where one or more input values is not of the appropriate data type, the Expression Status field displays an error message. Similarly, if the return type of the expression does not match the return type listed in the Type field, an error is displayed when you attempt to add the new expression.

Where **Any** is listed as the data type, the input or return can be any single value of any data type. Where **Any:1** is listed, the input or return can be one *or more* values of any data type.

The Conversion tab contains functions to convert values from one data type to another. For more information, see Chapter 8.

Handling Nil Values

The internal database of the server has a representation for unknown values. These are called **nil values** or **nils**.

When creating an expression, consider how nil values will be handled. The `ifnil` and `isnil` functions on the Selection tab are useful for testing for and/or excluding nil values. You can also use the `ifnil` function on the Miscellaneous tab to replace a nil value with another value.

Direct comparison (using =) of any value with nil will always give nil, even when the value being compared actually is nil. Therefore, selections such as `[Name] = str(nil)` will return the empty set. Always use `isnil([Name])` instead.

This section explains how the server deals with nil values in various situations.

Basic Functions

Most basic functions (row-wise operations) will result in nil if any of the values they operate on are nil. For example:

- The `isnil` function returns `true` if the argument is nil and `false` if not.
- The `ifnil` function returns the second argument if the first argument is nil, otherwise it returns the first argument.

Examples:

- `isnil(nil) → true`
- `isnil(3.0) → false`
- `[Total] = nil → nil` (direct comparison with nil always gives nil)
- `isnil([Total]) → false` (if `Total <> nil`) or `true` (if `Total = nil`)
- `ifnil(nil,9) → 9`
- `ifnil(8,9) → 8`
- `ifnil([Total],1000) → Total` (if `Total <> nil`) or `1000` (if `Total = nil`)

Certain functions on the Selection tab (`or, and, implies`) do not return nil under particular conditions. For example:

```
false and nil → false
```

In this case, the result will be `false` because one of the values is `false`. Similarly:

```
nil or false or true → true
```

Here, one of the values is `true`, so the result must be `true`.

When Function

When a nil is matched against the branches of a `when` functor, the `otherwise` branch will not be taken. This is because nil is an unknown value so it is not known which branch is the correct one to take. Instead, the result is nil.

The following example illustrates the *wrong* way to deal with nils; it will return nil when `[GENDER]` is nil, instead of `unknown` as was intended:

```
"female"       When:  ([GENDER] = "F")
"male"         When:  ([GENDER] = "M")
"unknown"      Otherwise.
```

This is the correct way to formulate this expression:

```
         "female"     When:  ([GENDER] = "F")
ifnil(   "male"       When:  ([GENDER] = "M")      , "unknown")
         str( nil)    Otherwise.
```

For more information about the `when` function, see Chapter 8.

Aggregate Functions

When using aggregate functions (column wise operation returning a unit value), if one of the values in the column is nil, the result will be nil.

Exceptions: The `size` and `exist` functions treat nil as a normal value.

Examples:

- `min([Total])` → `nil` (if one of the values = nil)
  → or min-value (if no value = nil)
  → in the Item table, `nil` will be returned
- `min(ifnil([Total],1000))` → in the Item table, min-value `52,10` will be returned because nil is replaced with 1000
- `size([Total]>=100)` → in the Item table, `7` will be returned, because nil is treated as a normal value (< 100)

Grouping

For grouping purposes, nils are considered equal and are therefore placed in their own group. This gives a better picture of the data than the alternate approach of considering every nil to be unique. For example:

`even_height(age,10)` → All nil values of `age` form a separate group.

Nils are also placed in their own group in the following situations:

- Finding the cardinality of a set using card
- Sampling an expression with unique tuples checked

Links

When records in a table are matched to records in another table according to the values of the attributes in their keys, nils are considered distinct and do not match with other nil values. Similarly, if any of the attributes in a multi-attribute key is nil, no match is made.

Data Upload

All values that do not belong to the domain return nil.

Examples:

- `flt("3.215") → 3.215`
- `flt("a23") → nil`
- `flt("23.4a") → nil`
- `flt("") → nil`

This also applies when null values are uploaded from a remote database—such values are uploaded as nil values.

Data Download

When data is downloaded to a database, nil values are returned null in the database. This means that attributes that are part of the key in the remote database cannot be nil (because keys in databases cannot be null).

Results of an Expression in the Expression Editor

Always inspect the results of an expression before creating a new attribute.

- Use the Preview button to inspect the entire expression, including all sub-expressions.
- If you want to inspect the contents of part of the expression, or of an attribute used in the expression, simply right-click the sub-expression or attribute and choose Preview of selection.

Aggregate Functions

The Aggregate tab can be used to deal with 1–*n* links. For example, the link between the tables *CUSTOMER* and *PORDER* is a 1–*n* link as 1 customer can have placed 0–*n* orders. When deriving new attributes for the *CUSTOMER* table, set-valued expressions must be reduced to a single value. This is what the aggregate functions do; they take a set of values as input and produce a single value as output.

Thus, the functions of the Aggregate tab can combine all the records of the order table that are linked to a specific *CUSTOMER_ID* and return just one result per customer. This result can be used as the value for a new attribute in the *Customer* table.

Restrictions

Aggregate functions allow you to add a restriction in a link to restrict the result of an aggregate expression. The result type of the restriction must be boolean.

Example: You are only interested in counting customer orders with *Commissionplan Y*. Using a restriction (in this case *Commissionplan = Y*) means that, instead of considering all the records of the *Porder* table that are linked to a specific *CUSTOMER_ID*, only the records for which the restriction is true will be considered.

Note: A link is a kind of restriction because, via a link, only the records linked to one specific record of another table will be considered (for example, the order records of a specific customer).

avg

The `avg` function returns the average of an expression.

Syntax

Input Type	Return Type
`avg(Integer)`	`Integer`
`avg(Long)`	`Long`
`avg(Float)`	`Float`
`avg(Double)`	`Double`

The return type of the `avg` function is the same as the return type of the expression.

Handling Nil Values

The `avg` function returns nil if the value of one of the attributes of the expression is nil.

Example:

$$\mathrm{avg}\begin{pmatrix}[1]\\ [14]\\ [5]\\ [7]\\ [18]\end{pmatrix} \to 9$$

count

The `count` function returns the number of records in a linked table that are linked to the current record (without inspecting the values of one of the attributes). The function returns 0 if there is no linked record.

Syntax

Input Type	Return Type
`count(Any)`	`Integer`

The return type is an integer (0, 1, 2, etc.).

Handling Nil Values

Nil values do not apply to the count function—the function does not inspect attribute values.

Example 1:

$$\text{count}\left(\begin{matrix}{[1]}\\ {[14]}\\ {[5]}\\ {[7]}\\ {[18]}\end{matrix}\right) \rightarrow 5$$

Example 2:

$$\text{count}\left(\begin{matrix}{[10],[11],[12]}\\ \ldots\\ {[1]}\\ {[20],[21],[22],[23],[24]}\end{matrix}\right) \rightarrow \begin{matrix}{[3]}\\ {[0]}\\ {[1]}\\ {[5]}\end{matrix}$$

exist

The `exist` function checks whether at least one of a set of booleans is true. The function returns false when there is no linked record.

Syntax

Input Type	**Return Type**
`exist(Boolean)`	`Boolean`

The return type is a bit with the values false or true.

Handling Nil Values

`exist{true, nil}`	`true`
`exist{false, nil}`	`false`
`exist{nil, nil}`	`false`

Example 1:

$$\text{exist}\begin{pmatrix}[\text{true}]\\ [\text{false}]\\ [\text{true}]\\ [\text{true}]\\ [\text{false}]\end{pmatrix}\rightarrow \text{true}$$

Example 2:

$$\text{exist}\begin{pmatrix}[\text{false}],[\text{false}],[\text{true}]\\ \\ [\text{false}]\\ [\text{true}],[\text{true}],[\text{true}],[\text{true}]\end{pmatrix}\rightarrow \begin{matrix}[\text{true}]\\ [\text{false}]\\ [\text{false}]\\ [\text{true}]\end{matrix}$$

max

The `max` function returns the maximum value of an expression. It returns nil if there is no linked record.

Syntax

Input Type	**Return Type**
`max(Any:1)`	`Any:1`

The return type of the `max` function is the same as the return type of the expression.

Handling Nil Values

The `max` function returns nil if the value of one of the attributes of the expression is nil.

Example 1:

$$\text{max}\begin{pmatrix}[1]\\ [14]\\ [5]\\ [7]\\ [18]\end{pmatrix}\rightarrow 18$$

Example 2:

$$\text{max}\begin{pmatrix}[10],[11],[12]\\ \ldots\\ [1]\\ [20],[21],[22],[23],[24]\end{pmatrix}\rightarrow \begin{matrix}[12]\\ [\text{nil}]\\ [1]\\ [24]\end{matrix}$$

min

The `min` function returns the minimum value of an expression. It returns nil if there is no linked record.

Syntax

Input Type	Return Type
`min(Any:1)`	`Any:1`

The return type of the `min` function is the same as the return type of the expression.

Handling Nil Values

The `min` function returns nil if the value of one of the attributes of the expression is nil.

Example 1:

$$\min\begin{pmatrix} {[1]} \\ {[14]} \\ {[5]} \\ {[7]} \\ {[18]} \end{pmatrix} \to 1$$

Example 2:

$$\min\begin{pmatrix} {[10],[11],[12]} \\ \cdots \\ {[1]} \\ {[20],[21],[22],[23],[24]} \end{pmatrix} \to \begin{matrix} {[10]} \\ {[nil]} \\ {[1]} \\ {[20]} \end{matrix}$$

size

The `size` function returns the number of true values of an expression (that is, the number of records for which the expression is known to be true). The function returns 0 if there is no linked record.

Syntax

Input Type	Return Type
`size(Boolean)`	`Integer`

Handling Nil Values

The `size` function considers nil to be a normal value and does not return nil if the value of one of the attributes of the expression is nil.

Example 1:

$$\text{size}\left(\begin{array}{c}[\text{true}]\\ [\text{false}]\\ [\text{true}]\\ [\text{true}]\\ [\text{false}]\end{array}\right)\rightarrow 3$$

Example 2:

$$\text{size}\left(\begin{array}{c}[\text{false}],[\text{false}],[\text{true}]\\ \\ [\text{false}]\\ [\text{true}],[\text{true}],[\text{true}],[\text{true}]\end{array}\right)\rightarrow\begin{array}{c}[1]\\ [0]\\ [0]\\ [4]\end{array}$$

sum

The `sum` function returns the sum of an expression. It returns 0 if there is no linked record.

Syntax

Input Type	Return Type
`sum(Integer)`	`Integer`
`sum(Long)`	`Long`
`sum(Float)`	`Float`
`sum(Double)`	`Double`

The return value is of the same type as those of the summarized attributes.

Handling Nil Values

The `sum` function returns nil when the value of one of the attributes of the expression is nil.

Example 1:

$$\text{sum}\left(\begin{array}{c}[1]\\ [14]\\ [5]\\ [7]\\ [18]\end{array}\right)\rightarrow 45$$

Example 2:

$$\text{sum}\left(\begin{array}{c}[10],[11],[12]\\ \ldots\\ [1]\\ [20],[21],[22],[23],[24]\end{array}\right)\rightarrow\begin{array}{c}[33]\\ [\text{nil}]\\ [1]\\ [110]\end{array}$$

Arithmetic Functions

The Arithmetic tab contains basic arithmetic functions that can be used with all types of numbers (for example, integers, longs, floats, doubles, and decimals).

Modulo

`%`

Modulo divides one integer by another and returns the remainder.

Syntax

Input Type	Return Type
`Integer % Integer`	`Integer`

Example:

`3 % 2 → 1`

Multiplication

*

The `*` function is a regular arithmetic operator and can be used with all numeric types. The return value has the same type as the expression.

Syntax

Input Type	Return Type
`Integer * Integer`	`Integer`
`Long * Long`	`Long`
`Float * Float`	`Float`
`Double * Double`	`Double`
`Decimal * Decimal`	`Decimal`

Example:

`12.13 * 4 → 48.52`

Addition

+

The `+` function is a regular arithmetic operator and can be used with all numeric types. The return value has the same type as the expression.

Syntax

Input Type	Return Type
`Integer + Integer`	`Integer`
`Long + Long`	`Long`
`Float + Float`	`Float`
`Double + Double`	`Double`
`Decimal + Decimal`	`Decimal`

Example:

`12.13 + 4` → `16.13`

Subtraction

-

The `-` function is a regular arithmetic operator and can be used with all numeric types. The return value has the same type as the expression.

Syntax

Input Type	Return Type
`Integer - Integer`	`Integer`
`Long - Long`	`Long`
`Float - Float`	`Float`
`Double - Double`	`Double`
`Decimal - Decimal`	`Decimal`

Example:

`12.13 - 4` → `8.13`

Negation

-

The `-` function is a regular arithmetic operator and can be used with all numeric types. The return value has the same type as the expression. `-()` is the negation operator.

Syntax

Input Type	Return Type
`-(Integer)`	`Integer`
`-(Long)`	`Long`
`-(Float)`	`Float`
`-(Double)`	`Double`
`-(Decimal)`	`Decimal`

Examples:

```
-(12.13)   →  -12.13
-(-15)     →  15
```

Division

```
/
```

The / function is a regular arithmetic operator and can be used with all numeric types. The return value has the same type as the expression.

Syntax

Input Type	Return Type
`Integer / Integer`	`Integer`
`Long / Long`	`Long`
`Float / Float`	`Float`
`Double / Double`	`Double`
`Decimal / Decimal`	`Decimal`

Examples:

```
12.13 / 4   →  3.0325
3 / 4       →  0
```

abs

The abs function returns the absolute value of an expression (a number without its sign). This function can be used with all numeric types. The return value has the same type as the expression.

Syntax

Input Type	Return Type
`abs(Integer)`	`Integer`
`abs(Long)`	`Long`
`abs(Float)`	`Float`
`abs(Double)`	`Double`
`abs(Decimal)`	`Decimal`

Examples:

```
abs(12.13)   →  12.13
abs(-15)     →  15
```

Conversion Functions

The functions of the Conversion tab convert a value of any type into the specified type.

Handling Nil Values

When one of the expressions has a value of nil or cannot be converted, the functions of the Conversion tab return nil.

bit

The `bit` function converts an integer or string into a boolean value (logical true or false). If the value starts with one of {0, *f*, *F*, *n*, *N*}, it returns false; all other values return true.

Syntax

Input Type	**Return Type**
`bit(Integer)`	`Boolean`
`bit(String)`	`Boolean`

Examples:

```
bit(0)   → false
bit("B") → true
```

date (Conversion)

The `date` function converts a string containing a date into a date. The string must have the format *yyyy-mm-dd*, where *yyyy* denotes the year, *mm* the month, and *dd* the day.

Syntax

Input Type	**Return Type**
`date(String)`	`Date`

Example:

```
date("2008-10-31") → 2008-10-31
```

dbl

The `dbl` function converts an expression into a double. This function is not applicable to string expressions containing non-numeric characters.

Note: The decimal separator must be a decimal point. If not, the expression is not valid.

Syntax

Input Type	Return Type
`dbl(Integer)`	`Double`
`dbl(Long)`	`Double`
`dbl(Float)`	`Double`
`dbl(String)`	`Double`
`dbl(Decimal)`	`Double`

Examples:

```
dbl(12.674567)   →   12.6746
dbl("90089")     →   90089
```

decimal

The `decimal` function converts a number of any type into a decimal. The decimal separator must be a decimal point. If not, the expression is not valid. The function is not applicable to string expressions containing non-numeric characters.

Syntax

Input Type	Return Type
`decimal(Integer)`	`Decimal`
`decimal(Long)`	`Decimal`
`decimal(Float)`	`Decimal`
`decimal(String)`	`Decimal`
`decimal(Double)`	`Decimal`

Examples:

```
decimal("90089.56")   →  90089.56
```

flt

The `flt` function converts an expression of any type into a float. The decimal separator must be a decimal point. If not, the expression is not valid. The function is not applicable to string expressions containing non-numeric characters.

Syntax

Input Type	Return Type
`flt(Integer)`	`Float`
`flt(Long)`	`Float`
`flt(Double)`	`Float`
`flt(String)`	`Float`
`flt(Decimal)`	`Float`

Examples:

```
flt(12)        →  12.0
flt("90089")   →  90089.0
```

int

The `int` function converts an expression of any type into an integer (whole number). The function is not applicable to string expressions containing non-numeric characters. The integer conversion of true returns 1. The integer conversion of false returns 0.

Syntax

Input Type	Return Type
`int(Boolean)`	`Integer`
`int(Long)`	`Integer`
`int(Float)`	`Integer`
`int(Double)`	`Integer`
`int(String)`	`Integer`
`int(Decimal)`	`Integer`

Examples:

```
int(12.67)  →  12
int(true)   →  1
```

lng

The `lng` function converts an expression of any type into a long (large whole number). The function is not applicable to string expressions containing non-numeric characters.

Syntax

Input Type	Return Type
`lng(Integer)`	`Long`
`lng(Float)`	`Long`
`lng(Double)`	`Long`
`lng(String)`	`Long`
`lng(Decimal)`	`Long`

Examples:

```
lng(12.67)     → 12
lng("325678")  → 325678
```

str

The `str` function converts an expression of any type into a string.

Syntax

Input Type	Return Type
`str(Any)`	`String`
`str(Decimal, String)`	`String`

Examples:

```
str(123)   →  "123"
str(true)  →  "true"
```

timestamp

The `timestamp` function converts a string containing a timestamp into a timestamp. The string must have the format *yyyy-mm-dd hh:mm:ss.fff*, where *yyyy* denotes the year, *mm* the month, *dd* the day, *hh* the hours, *mm* the minutes, *ss* the seconds, and *fff* the milliseconds.

Syntax

Input Type	Return Type
`timestamp(String)`	`Timestamp`

Example:

```
timestamp[("2004-04-18 09:23:45 987") → 2004-04-18 09:23:45 987
```

Date and Time Functions

The Date and Time tab contains the functionality to deal with dates and times. The tab relates to the `date`, `daytime`, and `timestamp` data types.

adddays

The `adddays` function adds a specified number of days to a specified date and returns a new date value. It takes two arguments: the original date, and the number of days to add to that date. When the second argument is negative, it is subtracted from the first argument.

Syntax

Input Type	Return Type
`adddays(Date, Integer)`	`Date`

Handling Nil Values

If one of the parameters (date and/or days) is equal to nil, then the result is equal to nil.

Example:

```
adddays(date(1972, 7, 14), 11030) → 2002-9-25
```

addmonths

The `addmonths` function adds a specified number of months to a specified date and returns a new date value. It takes two arguments: the original date, and the number of months to add to that date. When the second argument is negative, it is subtracted from the first argument.

Syntax

Input Type	Return Type
`addmonths(Date, Integer)`	`Date`

Handling Nil Values

If one of the parameters (date and/or days) is equal to nil, then the result is equal to nil.

Example:

```
addmonths(2001-04-18, 11) → 2002-03-11
```

addyears

The `addyears` function adds a specified number of years to a specified date and returns a new date value. It takes two arguments: the original date, and the number of years to add to that date. When the second argument is negative, it is subtracted from the first argument.

Syntax

Input Type	Return Type
`addyears(Date, Integer)`	`Date`

Handling Nil Values

If one of the parameters is equal to nil, then the result is equal to nil.

Example:

```
addyears(2001-04-18, 2) → 2003-04-18
```

date

The `date` function returns a date from a timestamp, or three integers representing the year, month, and day. In the latter case, the first argument is `year`, the second argument is `month`, and the third argument is `day`.

Syntax

Input Type	Return Type
`date(Timestamp)`	`Date`
`date(Integer, Integer, Integer)`	`Date`

Handling Nil Values

- If the `timestamp` is equal to nil, then the function returns nil.
- If the `year`, `month`, or `day` is equal to nil or out of range (for example, `day` is equal to 32), then the function returns nil.

Examples:

```
date(2001-04-18 09:23:15:965)    →      2001-04-18
date(2001, 04, 18)               →      2001-04-18
```

day

The `day` function returns the day of the month (value between 1 and 31) of a specified date.

Syntax

Input Type	Return Type
`day(Date)`	`Integer`

Handling Nil Values

If the `date` parameter is equal to nil, the function returns nil.

Example:

```
date(2004-04-18) → 18
```

dayofweek

The `dayofweek` function returns the day of the week (value between 1 and 7) of a specified date. Sunday is equal to 1, Monday is equal to 2, etc..

Syntax

Input Type	Return Type
`dayofweek(Date)`	`Integer`

Handling Nil Values

If the parameter `date` is equal to nil, the function returns nil.

Example:

`dayofweek(2001-04-18)` → `4(Wednesday`)

dayofyear

The `dayofyear` function returns the day of the year of a specified date. The first of January is equal to 1, the second of January is equal to 2, etc.

Syntax

Input Type	Return Type
`dayofyear(Date)`	`Integer`

Handling Nil Values

If the `date` parameter is equal to nil, the function returns nil.

Example:

`dayofyear(2001-04-18)` → `108`

diff

The `diff` function returns the number of days between two specified dates. If the first date is before the second date, the result is negative.

Syntax

Input Type	Return Type
`diff(Date, Date)`	`Integer`

Handling Nil Values

If one of the `date` parameters is equal to nil, the function returns nil.

Examples:

```
diff(2001-04-18, 2001-04-16)  →  2
diff(2001-04-18, 2001-04-20)  →  -2
```

month

The `month` function returns the month (value between 1 and 12) of a specified date.

Syntax

Input Type	Return Type
`month(Date)`	`Integer`

Handling Nil Values

If the `date` parameter is equal to nil, this function returns nil.

Example:

```
month(2001-04-18) → 4
```

weekofyear

The `weekofyear` function returns the week of the year from the date. A new week starts with Sunday.

Syntax

Input Type	Return Type
`weekofyear(Date)`	`Integer`

Handling Nil Values

If the `date` parameter is equal to nil, the function returns nil.

Example:

```
weekofyear(2001-04-18) → 16
```

year

The `year` function returns the year of a specified date.

Syntax

Input Type	Return Type
`year(Date)`	`Integer`

Handling Nil Values

If the `date` parameter is equal to nil, the function returns nil.

Example:

```
year(2001-04-18) → 2001
```

Miscellaneous Functions

The Miscellaneous tab contains functions that do not belong on other tabs.

Handling Nil Values

Where one of the expressions has value nil, the functions of the Miscellaneous tab return nil.

Exception: The `ifnil` function replaces nil with a new value.

ifnil

The `ifnil` function replaces each nil value with a new value. It takes two arguments: the original value of the expression, and the new value to replace each nil value. If the value of the expression is not nil, then the original value of the expression is returned in the new expression.

Note: The first argument cannot be a boolean.

Syntax

Input Type	Return Type
`ifnil(Any:1, Any:1)`	`Any:1`

Handling Nil Values

The `ifnil` function replaces nil with a new value.

Examples:

```
ifnil(“hello", str(nil))      →     "hello"
ifnil(str(nil), "see you")    →     "see you"
ifnil("goodbye", "see you")   →     "goodbye"
ifnil(str(nil), str(nil))     →    nil
```

when

The `when` function checks a specified condition and returns a value as appropriate. It takes three arguments: the condition to check, the value to return when the condition is true, and the value to return when the condition is false.

In the expression editor, the function `when` is displayed as value-when-true when the condition is met, and value-when-false otherwise.

You can nest a `when` function in another `when` function. If you press Enter after the condition, a new `when` function is added.

Syntax

```
Any:1 When:  Boolean

Any:1 Otherwise.
```

Handling Nil Values

The `when` function returns nil if the condition returns nil. For additional details about how the `when` function handles nil values, see Handling Nil Values in Chapter 8.

Example 1:

`"hello"`	When: true	→	`"hello”`
`"goodbye"`	Otherwise.		

Example 2:

`"hello"`	When: false	→	`"goodbye"`
`"goodbye"`	Otherwise.		

Example 3:

`"hello"`	When: false	→	`"see you"`
`"see you"`	When: true		
`"farewell"`	When: false		
`"goodbye"`	Otherwise.		

Selection Functions

The Selection tab contains functions for comparing the values of expressions.

Note: All operators in the Selection operator group return a boolean value.

Not Equal To

!=

The `!=` function compares two expressions and returns `true` if they are not equal.

Syntax

Input Type	Return Type
`Any:1 != Any:1`	`Boolean`

Handling Nil Values

The `=!` function returns nil if one of the expressions is nil.

Examples:

```
1 != 2    → true
1 != 1    → false
```

Less Than

<

The < function compares two expressions and returns `true` if the left expression is less than the right expression.

Syntax

Input Type	Return Type
`Any:1 < Any:1`	`Boolean`

Handling Nil Values

The < function returns nil if one of the expressions is nil.

Examples:

```
1 < 2  → true
1 < 1  → false
```

Less Than or Equal To

<=

The <= function compares two expressions and returns `true` if the left expression is less than or equal to the right expression.

Syntax

Input Type	Return Type
`Any:1 <= Any:1`	`Boolean`

Handling Nil Values

The <= function returns nil if one of the expressions is nil.

Examples:

```
1 <= 2   → true
2 <= 2   → true
3 <= 2   → false
```

Equal To

`=`

The = function compares two expressions and returns `true` if they are equal.

Syntax

Input Type	Return Type
`Any:1 = Any:1`	`Boolean`

Handling Nil Values

The = function returns nil if one of the expressions is nil.

Examples:

```
1 = 1   → true
1 = 2   → false
```

Greater Than

`>`

The > function compares two expressions and returns `true` if the left expression is greater than the right expression.

Syntax

Input Type	Return Type
`Any:1 > Any:1`	`Boolean`

Handling Nil Values

The > function returns nil if one of the expressions is nil.

Examples:

```
2 > 1   → true
1 > 1   → false
```

Greater Than or Equal To

```
>=
```

The >= function compares two expressions and returns `true` if the left expression is greater than or equal to the right expression.

Syntax

Input Type	Return Type
`Any:1 >= Any:1`	`Boolean`

Handling Nil Values

The >= function returns nil if one of the expressions is nil.

Examples:

```
1 >= 2  → false
2 >= 2  → true
2 >= 1  → true
```

and

The `and` function checks whether two boolean expressions are both true.

Syntax

Input Type	Return Type
`Boolean and Boolean`	`Boolean`

Handling Nil Values

If one of the boolean expressions returns nil, the result depends on the other boolean expression. If that expression returns true, the result is nil. If that expression returns false, the result is false.

Examples:

```
true and true  → true
true and nil   → nil
nil and true   → nil
false and nil  → false
nil and false  → false
```

between

The `between` function compares three expressions and determines whether the value of the first expression is at least equal to the value of the second expression, and at most equal to the value of the third expression. So the second expression serves as lower boundary and the third as upper boundary for the first expression.

Syntax

Input Type	Return Type
`between(Any:1, Any:1, Any:1)`	`Boolean`

Handling Nil Values

The `between` function returns nil when the first expression is nil. A nil value of the second expression is regarded as the absence of a lower boundary, while a nil value of the third expression is regarded as the absence of an upper boundary.

Examples:

```
between(3, 1, 3)   → true
between(3, 1, 2)   → false
```

implies

The `implies` function compares two expressions and returns:

- true if the first is false *or* the second is true
- false if the first is true *and* the second is false

Syntax

Input Type	Return Type
`Boolean implies Boolean`	`Boolean`

Handling Nil Values

The `implies` function returns nil if either expression is nil.

Examples:

```
true implies true    → true
true implies false   → false
nil implies false    → nil
```

in

The `in` function checks whether a specified search value is in a specified set of values. The return value is true if the value is in the set of values and false if it is not.

Syntax

Input Type	Return Type
`Any:1 in(Any:1, Any:1)`	`Boolean`

Handling Nil Values

The function `in` handles nil as a normal value.

Examples:

```
1 in {1,2}                  → true
"A" in {"B","C","D","E"}    → false
```

isnil

The `isnil` function checks whether a value is nil.

Syntax

Input Type	Return Type
`isnil(Any:1)`	`Boolean`

Handling Nil Values

The `isnil` function returns true if the value of the expression is nil and false if not.

Examples:

```
isnil(nil)         → true
isnil("Hello")     → false
```

not

The `not` function returns the negation of a boolean expression.

Syntax

Input Type	Return Type
`not(Boolean)`	`Boolean`

Handling Nil Values

The `not` function returns nil when the Boolean expression is nil.

Examples:

```
not(false)  → true
not(true)   → false
not(nil)    → nil
```

or

The `or` function checks whether at least one of two boolean expressions is true.

Syntax

Input Type	Return Type
`Boolean or Boolean`	`Boolean`

Handling Nil Values

If the `or` expression consists of true and a nil value, the function returns true. If the `or` expression contains false and a nil value, the result is nil.

Examples:

```
true or false    → true
false or false   → false
true or true     → true
true or nil      → true
```

String Functions

This tab contains string functions.

Handling Nil Values

If one of the expressions has value nil, the functions of the String tab return nil.

Concatenate

+ (concatenate)

The + function concatenates two strings.

Syntax

Input Type	Return Type
`String + Any`	`String`

Examples:

```
"child" + "ren"          → children
"hello " + "anna"        → hello anna
```

chrAt

The function `chrAt` returns the character at a specified position of a string. It takes two arguments: the string itself, and an integer representing the position of the character to return. The first character of the string is position 0. Using non-existent positions (smaller than zero or greater than the length minus one) returns a nil value

DEPRECATED: Non-ASCII characters become nil. Use: *string(s,index,1)*.

Syntax

Input Type	Return Type
`chrAt(String, Integer)`	`String`

Examples:

```
chrAt("hello world", 0)       → "h"
chrAt ("hello world", 1)      → "e"
chrAt ("hello world", -1)     → nil
```

endsWith

The `endswith` function checks whether a string ends with a specified value. It takes two arguments: the string to check, and the value to compare against the string.

Syntax

Input Type	Return Type
`endsWith(String, String)`	`Boolean`

Examples:

```
endsWith("hello", "lo")      → true
endsWith("hello", "la")      → false
endsWith("hello", "o")       → true
```

length

The `length` function returns an integer representing the length of a string.

Syntax

Input Type	Return Type
`length(String)`	`Integer`

Examples:

```
length("hello") → 5
length("")      → 0
```

ltrim

The `ltrim` function strips white space from the start of a string.

Syntax

Input Type	Return Type
`ltrim(String)`	`String`

Example:

```
ltrim(" hello")         → hello
```

rtrim

The `rtrim` function strips white space from the end of a string.

Syntax

Input Type	Return Type
`rtrim(String)`	`String`

Example:

```
rtrim("hello ")  → hello
```

search

The `search` function checks whether a string contains a specified value. It takes two arguments: the string itself, and the search value to run against the string. The return value is an integer containing the position of the first search value from the left side. If the string does not contain the specified value, the return value is –1.

The first character of the string is in position 0.

Syntax

Input Type	Return Type
`search(String, String)`	`Integer`
`search(String, Character)`	`Integer`

Examples:

```
search("hello, hello", "ll")    → 2
search("hello, hello", "l")     → 2
search("hello, hello", "s")     → -1
```

startsWith

The `startswith` function checks whether a string starts with a specified value. It takes two arguments: the string itself, and the value to compare against the string.

Syntax

Input Type	Return Type
`startsWith(String, String)`	`Boolean`

Examples:

```
startsWith("hello", "he")     → true
startsWith("hello", "ha")     → false
startsWith("hello", "h")      → true
```

string

The `string` function returns a substring containing characters from a string, starting at a specified position. It takes up to three arguments: the string itself, an integer representing the starting position of the substring, and optionally, an integer specifying the number of characters in the substring.

The first character of the string is in position 0. If the starting position has a negative value, the position is counted from the end of the string instead of the beginning.

Syntax

Input Type	Return Type
`string(String, Integer, Integer)`	`String`
`string(String, Integer)`	`String`

Examples:

```
string("hello", 1, 3)    → "ell"
string("hello", 1)       → "ello"
string("hello", -2)      → "lo"
```

toLower

The `toLower` function changes a string to lowercase.

Syntax

Input Type	Return Type
`toLower(String)`	`String`

Example:

```
toLower("HELLO")     → "hello"
```

toUpper

The `toUpper` function changes a string to uppercase.

Syntax

Input Type	Return Type
`toUpper(String)`	`String`

Example:

```
toUpper("hello")     → "HELLO"
```

Importing and Exporting Enterprise View Objects

The process of importing and exporting Enterprise View objects is similar to importing and exporting other repository objects. However, there are a few important considerations:

Enterprise View

- An imported Enterprise View is considered the master Enterprise View. This means that the imported Enterprise View's LATEST version will be the LATEST in the newly merged Enterprise View (imported LATEST version takes precedence over existing LATEST version). Any matching labels are retained with the imported Enterprise View versions. When viewing the Enterprise View properties, there can be multiple first versions (for example, when the imported and existing views do not share the same ID number). In cases of identical versions, the object timestamp can be used to differentiate between versions.

- The Enterprise View must be exported/imported prior to importing other Enterprise View objects. Enterprise View objects have dependencies on the Enterprise View, requiring that the appropriate Enterprise View is in place prior to importing.
- The Enterprise View and resource definitions must be exported/imported manually. Exporting the entire content repository will include the resource definitions and Enterprise View, but the import process will not include the resource definitions or Enterprise View.

Data Source Definitions

When importing other repository objects, matching Data Source Definitions must exist in the importing repository. The importing repository must contain the same Data Source Definitions that exist in the repository from which the objects were exported. Importing Enterprise View objects (Enterprise View, Application View, Data Provider Definition, Data Provider Definition - Real Time) does not require that matching Data Source Definitions exist in the importing repository.

Versions and labels

Although the LATEST label is applied to the LATEST version of the imported Enterprise View, if that version is changed, the new version number is applied to the importing repository Enterprise View.

Non-Enterprise View users

Non-Enterprise View user profiles can export objects (if they have the export action defined in role) but cannot import objects.

Refer to Export and Import Overview for general import/export information.

Enterprise View Data Types

The Enterprise View data types are an approximation of the SQL types defined by the ODBC and JDBC standards. There are a few data type mapping issues you should consider when defining a Data Provider Definition.

Data Provider Definition mappings will not allow a loss of precision. For example, you can map SQL_INTEGER to the Enterprise View data type Double, but you cannot map SQL_DOUBLE to the Enterprise View data type Integer since the number’s decimal portion would be lost.

The exceptions to the no loss of precision rule are the data types Date, Time, and Timestamp. Not all database vendors support separate data types for Date and Time. For instance Microsoft SQL Server does not support any separation of date from time and therefore only supports the data type Datetime (which is exposed as SQL_TIMESTAMP). Because of this, Enterprise View allows mapping SQL_TIMESTAMP fields to the Date, Time, and Timestamp fields. The preferred mapping for SQL_TIMESTAMP is the Enterprise View data type Timestamp. You must set the correct Enterprise View data type for columns imported from databases that do not separate date from time.

The following table describes which SQL data types can be mapped to Enterprise View data types. These mappings are exposed through the Data Provider Definition Editor.

Each SQL data type or subtype has one preferred mapping. In the case of SQL_DECIMAL and SQL_NUMERIC, the precision and scale are also considered (these data types have several subtypes, each with its own preferred mapping).

Table 8-3
Allowed Data Type Mappings

Enterprise View Data Type	SQL Data Type	Preferred Mapping?
String	SQL_CHAR	Y
	SQL_VARCHAR	Y
	SQL_LONGVARCHAR	Y
	SQL_WCHAR	Y
	SQL_WVARCHAR	Y
	SQL_WLONGVARCHAR	Y
Integer	SQL_INTEGER	Y
	SQL_SMALLINT	Y
	SQL_TINYINT	Y
	SQL_DECIMAL(1-9, 0)	Y
	SQL_DECIMAL(1 <= p-s <= 9, s < 0)	Y
	SQL_NUMERIC(1-9, 0)	Y
	SQL_NUMERIC(1 <= p-s <= 9, s < 0)	Y
Long	SQL_BIGINT	Y
	SQL_INTEGER	N
	SQL_SMALLINT	N
	SQL_TINYINT	N
	SQL_DECIMAL(1-9, 0)	N
	SQL_DECIMAL(10-18, 0)	Y
	SQL_DECIMAL(1 <= p-s <= 9, s < 0)	N
	SQL_DECIMAL(10 <= p-s <= 18, s < 0)	Y
	SQL_NUMERIC(1-9, 0)	N
	SQL_NUMERIC(10-18, 0)	Y
	SQL_NUMERIC(1 <= p-s <= 9, s < 0)	N
	SQL_NUMERIC(10 <= p-s <= 18, s < 0)	Y
Double	SQL_DOUBLE	Y
	SQL_FLOAT	Y
	SQL_REAL	N
	SQL_INTEGER	N
	SQL_SMALLINT	N
	SQL_TINYINT	N
	SQL_DECIMAL(1-15, 0)	N
	SQL_DECIMAL(1 <= p-s <= 15, s < 0)	N

Enterprise View Data Type	SQL Data Type	Preferred Mapping?
	SQL_DECIMAL(1-6, 1-38)	N
	SQL_DECIMAL(1-6, 39-308)	Y
	SQL_DECIMAL(7-15, 1-308)	Y
	SQL_NUMERIC(1-15, 0)	N
	SQL_NUMERIC(1 <= p-s <= 15, s < 0)	N
	SQL_NUMERIC(1-6, 1-38)	N
	SQL_NUMERIC(1-6, 39-308)	Y
	SQL_NUMERIC(7-15, 1-308)	Y
Float	SQL_REAL	Y
	SQL_SMALLINT	N
	SQL_TINYINT	N
	SQL_DECIMAL(1-6, 0)	N
	SQL_DECIMAL(1-6, 1-38)	Y
	SQL_DECIMAL(1 <= p-s <= 6, s < 0)	N
	SQL_NUMERIC(1-6, 0)	N
	SQL_NUMERIC(1-6, 1-38)	Y
	SQL_NUMERIC(1 <= p-s <= 6, s < 0)	N
Decimal	SQL_DECIMAL(16+, 1+)	Y
	SQL_DECIMAL(19+, 0)	Y
	SQL_DECIMAL(*, 309+)	Y
	SQL_DECIMAL(1-15, 0-308)	N
	SQL_DECIMAL(1 <= p-s <= 15, s < 0)	N
	SQL_NUMERIC(16+, 1+)	Y
	SQL_NUMERIC(19+, 0)	Y
	SQL_NUMERIC(*, 309+)	Y
	SQL_NUMERIC(1-15, 0-308)	N
	SQL_NUMERIC(1 <= p-s <= 15, s < 0)	N
	SQL_INTEGER	N
	SQL_SMALLINT	N
	SQL_TINYINT	N
	SQL_BIGINT	N
	SQL_DOUBLE	N
	SQL_FLOAT	N
	SQL_REAL	N
Boolean	SQL_BIT	Y
Date	SQL_DATE	Y
	SQL_TIMESTAMP	N
Time	SQL_TIME	Y
	SQL_TIMESTAMP	N

Enterprise View Data Type	SQL Data Type	Preferred Mapping?
Timestamp	SQL_TIMESTAMP	Y

Note: Enterprise View does not support SQL data types not specifically listed in the preceding table.

Many Enterprise View data types allow you to map multiple SQL data types. The SQL data type provided to the client application may change depending on which Data Provider Definition is in use. Client applications must dynamically handle data type changes and should not make assumptions about data types that are provided at run-time.

When you import physical table column definitions into the Enterprise View (through the Adding Columns to the Enterprise View process), a given SQL data type will map to only one Enterprise View data type. This mapping is determined by the SQL data type's preferred mapping.

Table 8-4
Preferred SQL Data Type Mappings

SQL Data Type	Preferred Enterprise View Data Type
SQL_CHAR	String
SQL_VARCHAR	String
SQL_LONGVARCHAR	String
SQL_WCHAR	String
SQL_WVARCHAR	String
SQL_WLONGVARCHAR	String
SQL_INTEGER	Integer
SQL_SMALLINT	Integer
SQL_TINYINT	Integer
SQL_BIGINT	Long
SQL_DOUBLE	Double
SQL_FLOAT	Double
SQL_REAL	Float
SQL_DECIMAL(1-9, 0)	Integer
SQL_DECIMAL(1 <= p-s <= 9, s < 0)	Integer
SQL_DECIMAL(10-18, 0)	Long
SQL_DECIMAL(10 <= p-s <= 18, s < 0)	Long
SQL_DECIMAL(1-6, 1-38)	Float
SQL_DECIMAL(1-6, 39-308)	Double
SQL_DECIMAL(7-15, 1-308)	Double
SQL_DECIMAL(16+, *)	Decimal
SQL_DECIMAL(*, 309+)	Decimal
SQL_NUMERIC(1-9, 0)	Integer
SQL_NUMERIC (1 <= p-s <= 9, s < 0)	Integer
SQL_NUMERIC (10-18, 0)	Long
SQL_NUMERIC (10 <= p-s <= 18, s < 0)	Long
SQL_NUMERIC (1-6, 1-38)	Float

SQL Data Type	Preferred Enterprise View Data Type
SQL_NUMERIC (1-6, 39-308)	Double
SQL_NUMERIC (7-15, 1-308)	Double
SQL_NUMERIC (16+, *)	Decimal
SQL_NUMERIC (*, 309+)	Decimal
SQL_BIT	Boolean
SQL_DATE	Date
SQL_TIME	Time
SQL_TIMESTAMP	Timestamp

Although a SQL data type may have a corresponding Enterprise View data type, the preferred mapping may not always be to the same Enterprise View data type. Precision and scale are also considered. This is due to the fact that vendors such as Oracle expose all of their non-floating point numbers as SQL_DECIMAL. Basing the preferred mappings at least partially on precision and scale is necessary to ensure that all Enterprise View data types can be imported from physical tables.

Note: The preferred mapping for SQL_DECIMAL is whichever Enterprise View data type best matches the scale and precision. Only in cases where there is no appropriate match to another data type will the preferred mapping for SQL_DECIMAL map to the data type Decimal. However, this does not mean that you cannot map any SQL_DECIMAL to the Enterprise View data type Decimal. On UNIX-based operating systems, SQL_BIGINT type is treated as numeric(19,0), which may result in loss of precision.

Chapter

9

Scoring

Scoring is the process of generating real-time values by supplying predictive models with input data. A scoring model is any artifact that can be used to produce output values given input data, such as a PMML file from PASW Statistics. In general, to use a model for generating scores:

1. Select a model to use for scoring from the repository.

2. Define a scoring configuration for the model.

3. Supply the configured model with data and generate scores.

The predictive model used for scoring can be defined using PASW Modeler streams, scenarios, or PMML generated from SPSS Inc. products. PASW Collaboration and Deployment Services supports PMML 3.2 and earlier versions. In addition, legacy markup (e.g. SPSS-ML) from older products may also be used for scoring. Note that some nodes in streams, such as the ADP node, must be trained before they can be used for scoring. For more information, see the PASW Modeler documentation.

Use the Deployment Manager to define model scoring configurations and to monitor the model scoring performance. Results generated from scoring can be viewed in the Deployment Portal or custom client applications.

Supported Scoring Functions and Models

Applying a predictive model to a set of data can produce a variety of scores, such as predicted values, predicted probabilities, and other values based on that model. The type of score produced is referred to as the **scoring function**. The following scoring functions are available:

PREDICT *Returns the predicted value of the target variable.*

STDDEV *Standard deviation.*

PROBABILITY *Probability associated with a particular category of a target variable.* Applies only to categorical variables. In the absence of the optional third parameter, *category*, this is the probability that the predicted category is the correct one for the target variable. If a particular category is specified, then this is the probability that the specified category is the correct one for the target variable.

CONFIDENCE *A probability measure associated with the predicted value of a categorical target variable.* Applies only to categorical variables.

NODEID *The terminal node number.* Applies only to tree models.

CUMHAZARD *Cumulative hazard value.* Applies only to Cox regression models.

NEIGHBOR	*The ID of the kth nearest neighbor.* Applies only to nearest neighbor models. In the absence of the optional third parameter, k, this is the ID of the nearest neighbor. The ID is the value of the case labels variable, if supplied, and otherwise the case number.
DISTANCE	*The distance to the kth nearest neighbor.* Applies only to nearest neighbor models. In the absence of the optional third parameter, k, this is the distance to the nearest neighbor. Depending on the model, either Euclidean or City Block distance will be used.

The following table lists the set of scoring functions available for each type of model that supports scoring. The function type denoted as `PROBABILITY (category)` refers to specification of a particular category (the optional third parameter) for the `PROBABILITY` function.

Table 9-1
Supported functions by model type

Model type	**Supported functions**
Tree (categorical target)	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`, `NODEID`
Tree (scale target)	`PREDICT`, `NODEID`
Boosted Tree (C5.0)	`PREDICT`, `CONFIDENCE`
Linear Regression	`PREDICT`, `STDDEV`
Binary Logistic Regression	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
Conditional Logistic Regression	`PREDICT`
Multinomial Logistic Regression	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
General Linear Model	`PREDICT`, `STDDEV`
Discriminant	`PREDICT`, `PROBABILITY`
TwoStep Cluster	`PREDICT`
K-Means Cluster	`PREDICT`, `CONFIDENCE`
Kohonen	`PREDICT`
Neural Net (categorical target)	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
Neural Net (scale target)	`PREDICT`
Naive Bayes	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
Anomaly Detection	`PREDICT`
Ruleset	`PREDICT`, `CONFIDENCE`
Generalized Linear Model (categorical target)	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
Generalized Linear Model (scale target)	`PREDICT`, `STDDEV`
Ordinal Multinomial Regression	`PREDICT`, `PROBABILITY`, `PROBABILITY (category)`, `CONFIDENCE`
Cox Regression	`PREDICT`, `CUMHAZARD`
Nearest Neighbor (scale target)	`PREDICT`, `NEIGHBOR`, `DISTANCE`
Nearest Neighbor (categorical target)	`PREDICT`, `PROBABILITY`, `NEIGHBOR`, `DISTANCE`

- For the Binary Logistic Regression, Multinomial Logistic Regression, and Naive Bayes models, the value returned by the `CONFIDENCE` function is identical to that returned by the `PROBABILITY` function.

- For the K-Means model, the value returned by the `CONFIDENCE` function is the least distance.
- For tree and ruleset models, the confidence can be interpreted as an adjusted probability of the predicted category and is always less than the value given by `PROBABILITY`. For these models, the confidence value is more reliable than the value given by `PROBABILITY`.
- For neural network models, the confidence provides a measure of whether the predicted category is much more likely than the second-best predicted category.
- For Ordinal Multinomial Regression and Generalized Linear Model, the `PROBABILITY` function is supported when the target variable is binary.
- For nearest neighbor models without a target variable, the available functions are `NEIGHBOR` and `DISTANCE`.

Scoring Configurations

Before a model can be used for scoring, supplemental information must be defined. For example, a PASW Modeler stream containing a Enterprise View source node needs an associated real-time data provider definition specifying how to retrieve the required data. Such information constitutes a **scoring configuration** for the model, and defines scoring parameters such as:

- identification information for the configuration itself
- identification information for the model used for scoring
- the data provider for the input
- settings for logging
- the order of the input attributes
- cache size used for scoring models

A single model may be used in a variety of scoring situations that require different scoring parameters. For example, scores may be based on a test data provider for internal purposes and on a different data provider for production usage. Alternatively, the information being logged as result of scoring may depend on the scoring situation. To allow a model to be used in differing scoring circumstances, any model may be associated with multiple scoring configurations.

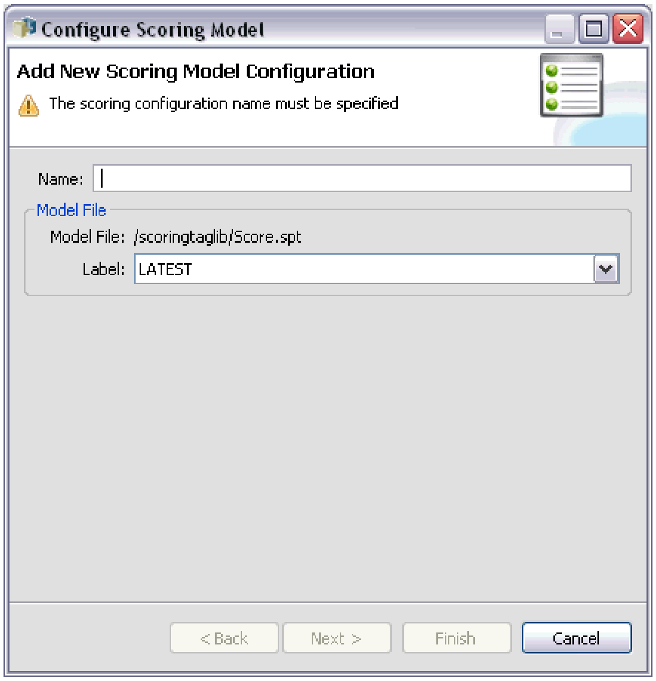
Scoring configurations can be suspended to temporarily prevent the processing of score requests. A suspended configuration must be reactivated before it can be used to generate scores.

To create a scoring configuration for a model, right-click the model to be configured for scoring in the Content Explorer and select Configure Scoring. The Scoring Configuration wizard appears.

New Scoring Model Configuration

The Add New Scoring Model Configuration page of the Scoring Configuration wizard identifies the model being used for scoring.

Figure 9-1
Model Identification



Configure Scoring Model

Add New Scoring Model Configuration

⚠ The scoring configuration name must be specified

Name:

Model File

Model File: /scoringtaglib/Score.spt

Label: LATEST

Name. The name for the scoring configuration. A model may have many scoring configurations, each identified by its unique name.

Model File. The name of the model file associated with the configuration.

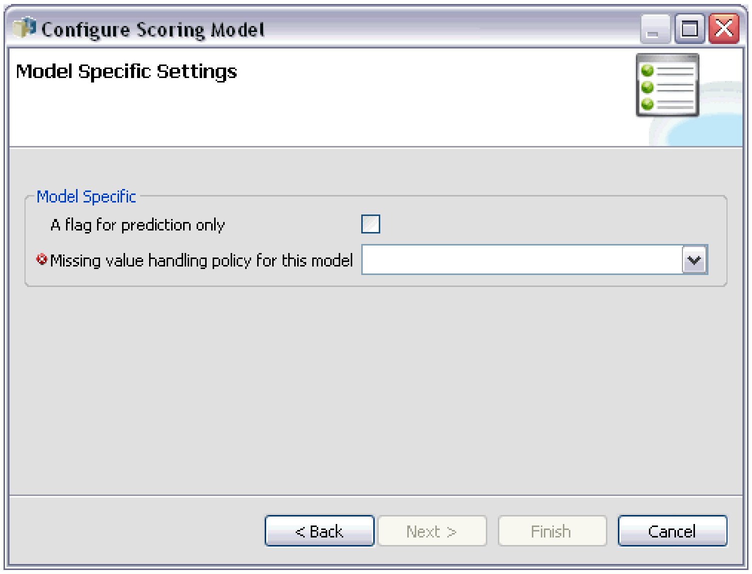
Label. The label identifying the version of the model file configured for scoring.

Click Next to specify additional settings.

Model Specific Settings

The Model Specific Settings page of the Scoring Configuration wizard defines settings parameters for the specific model being configured. The settings parameters available vary across different models.

Figure 9-2
Model Specific Settings



A flag for prediction only □

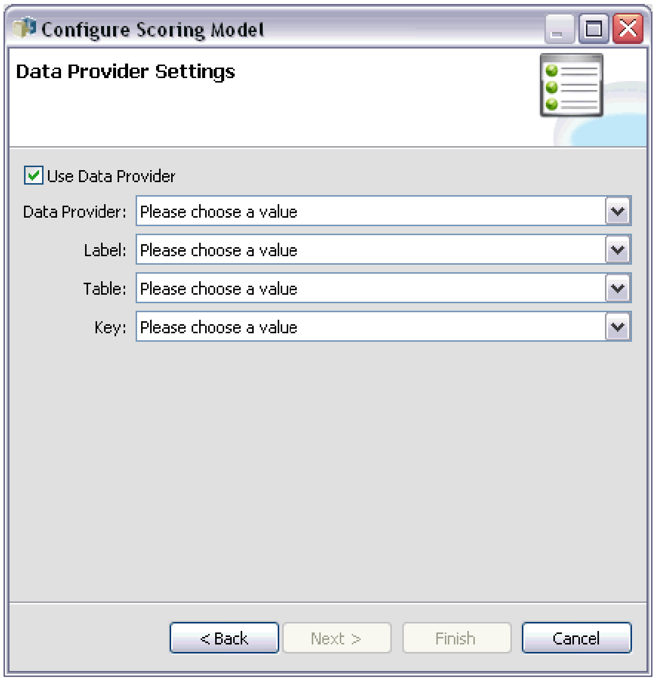
Missing value handling policy for this model

Specify values for the specific settings offered for the model. Click **Next** to specify additional settings.

Data Provider Settings

The Data Provider Settings page of the Scoring Configuration wizard defines the data used for scoring. If the model being configured for scoring uses Enterprise View data as a source, the real-time data provider definition for the source must be specified.

Figure 9-3
Data Provider Settings



Configure Scoring Model

Data Provider Settings

☑ Use Data Provider

Data Provider: Please choose a value

Label: Please choose a value

Table: Please choose a value

Key: Please choose a value

Use Data Provider. If selected, the configuration uses data from a real-time data provider definition.

Data Provider. Select the definition to use from the list of all available real-time data provider definitions currently specified within the system. To be an available data provider, the provider data set must specify a key that matches the key in the table in the Enterprise View.

Label. Select the label to use from the list of all labels available in the selected real-time data provider definition.

Table. Select the table to use from the list of all tables available in the selected combination of the real-time data provider definition and the label.

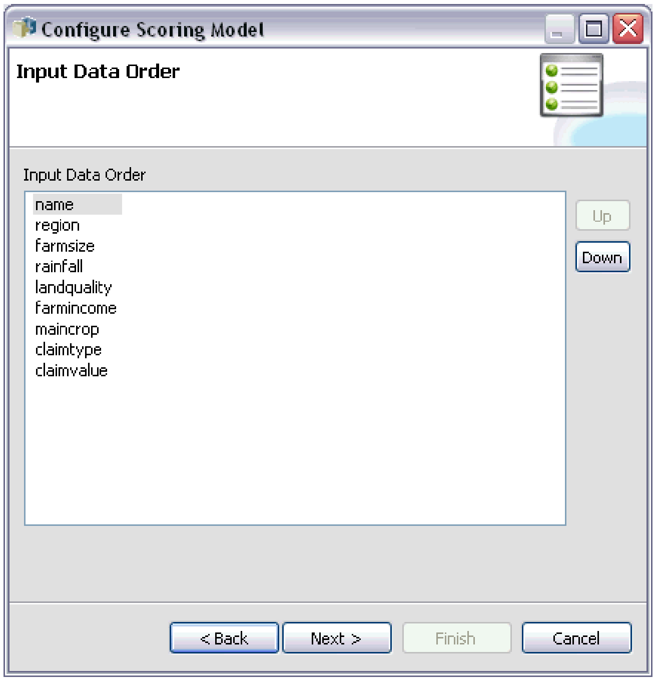
Key. Select the key to use from the list of all keys available as a result of the real-time data provider definition, label, and table specifications.

Click Next to specify additional settings.

Input Data Order

The Input Data Order page of the Scoring Configuration wizard allows control over the order of the input fields for scoring.

Figure 9-4
Input Data Order



name
region
farmsize
rainfall
landquality
farmincome
maincrop
claimtype
claimvalue

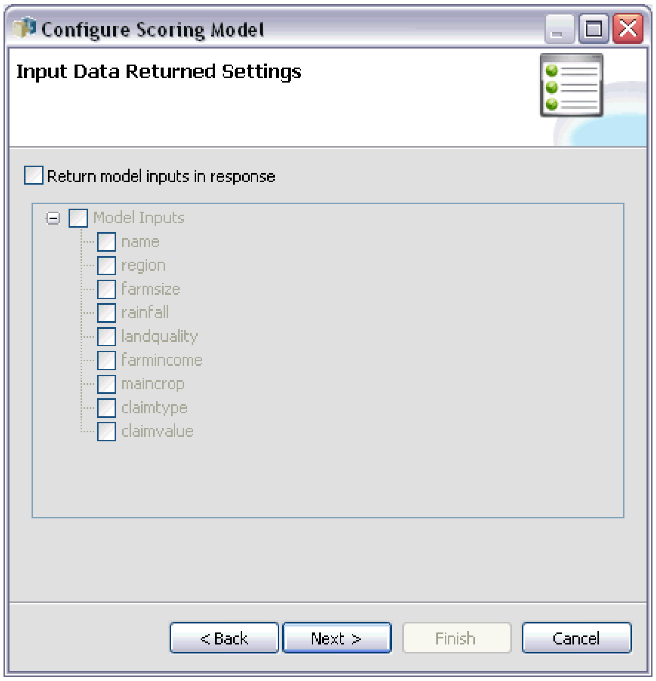
Use the Up and Down buttons to define the field order, which corresponds to the order of the attributes returned from PASW Collaboration and Deployment Services. In addition, if a client requests for scoring omit field names, the input values being passed are assumed to be in this order. Omitting the field names may streamline client/server communications for configurations involving a high volume of scores.

Click Next to specify additional settings.

Input Data Returned Settings

The Input Data Returned Settings page of the Scoring Configuration wizard allows control over the contents of the scoring response for inputs.

Figure 9-5
Model Inputs in Response



Return model inputs in response. If selected, the scoring response contains values for specified model inputs.

The model inputs that can be included with the returned scores appear in the list. Select the items to include in the response for this particular scoring configuration.

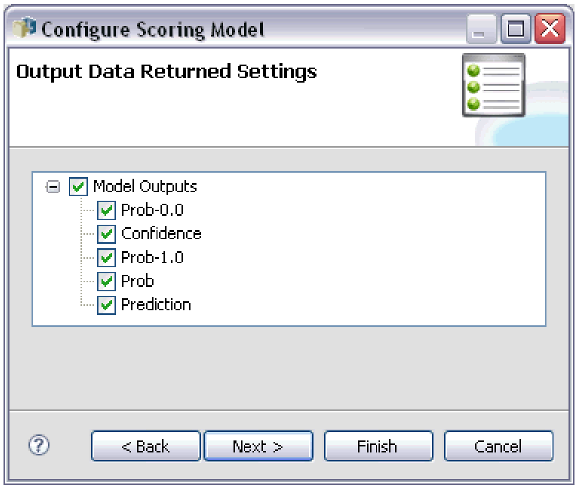
Click Next to specify additional settings.

Output Data Returned Settings

Scoring requests can produce a variety of scoring outputs, with the type of output being dependent on the model used for scoring. The scoring configuration allows the specification of a subset of outputs to be returned in response to a scoring request. For models that offer many scoring outputs, restricting the returned values to a small set can help to optimize scoring performance.

The Output Data Returned Settings page of the Scoring Configuration wizard allows control over the contents of the scoring response for outputs.

Figure 9-6
Model Outputs in Response



Configure Scoring Model

Output Data Returned Settings

- ☑ Model Outputs
  - ☑ Prob-0.0
  - ☑ Confidence
  - ☑ Prob-1.0
  - ☑ Prob
  - ☑ Prediction

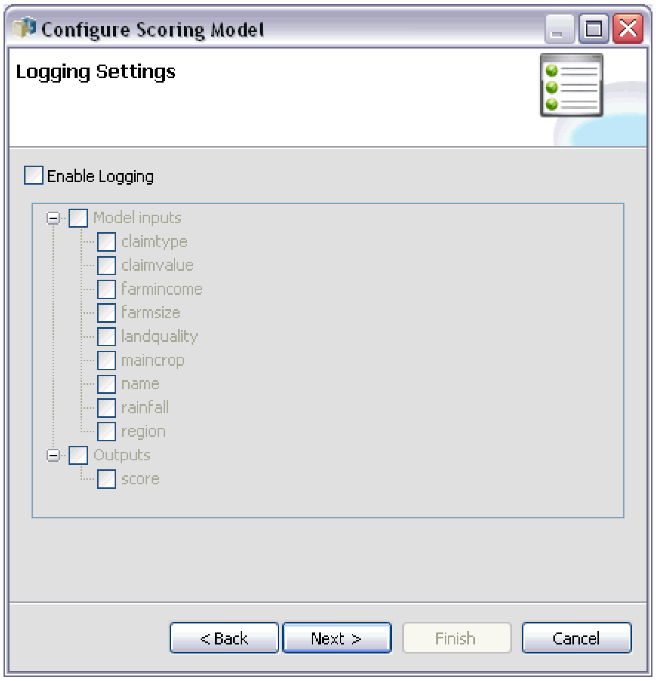
The model outputs that can be included in the response appear in the list. For information about specific items for PMML files, see Supported Scoring Functions and Models.

Select the outputs to include for this particular scoring configuration. At least one output item must be selected. Click Next to specify additional settings.

Logging Settings

The Logging Settings page of the Scoring Configuration wizard defines the content of audit logs for the scoring process.

Figure 9-7
Logging Settings



□ Enable Logging

- □ Model inputs
  - □ claimtype
  - □ claimvalue
  - □ farmincome
  - □ farmsize
  - □ landquality
  - □ maincrop
  - □ name
  - □ rainfall
  - □ region
- □ Outputs
  - □ score

To capture logging information, select Enable Logging. Measures that can be included in the logs for scoring appears in the list.

Model Inputs. The input fields used for scoring.

Model Outputs. Types of outputs from the model. The list of available outputs depends on the type of model being configured for scoring. For more information, see the topic Supported Scoring Functions and Models on p. 212.

Scoring Engine Metrics. Measurements that reflect the performance of the scoring configuration. Available metrics include:

- **Configuration Scores.** Total number of scores produced by a specific scoring configuration.
- **Configuration Uptime.** Amount of time, measured in seconds, that the scoring configuration has been available for scoring.
- **Average Latency.** Average amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Minimum Latency.** Smallest amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Maximum Latency.** Largest amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Cache Hits.** Number of successful attempts to retrieve data from the memory cache for a scoring configuration.

- **Cache Misses.** Number of failed attempts to retrieve data from the memory cache for a scoring configuration. Each failed attempt results in a new service call to retrieve the necessary data.
- **Score Elapsed Time.** Amount of time, measured in milliseconds, between the request for a score and the generation of the score.

Scoring Engine Properties. Characteristics of the scoring configuration itself. Entries include:

- **Scoring Configuration Serial.** Unique identifier of the scoring request for the configuration.
- **Scoring Configuration Name.** The name of the scoring configuration.
- **Model ID.**Repository identifier for the model file associated with the configuration.
- **Model Path.**Repository path for the model file associated with the configuration.
- **Model MIME type.** MIME type of the model file associated with the configuration.
- **Model Version Marker.** Marker identifying the version of the model file associated with the configuration.
- **Model Version Label.** Label identifying the version of the model file associated with the configuration.

If the configuration uses a data provider, additional scoring engine properties can be included in the scoring logs.

- **Enterprise View Version Marker.** Marker identifying the version of the Enterprise View associated with the data provider.
- **Application View Name.** The name of the Application View associated with the data provider.
- **Application View Path.** Repository path for the Application View associated with the data provider.
- **Application View ID.** Internal repository identifier for the Application View associated with the data provider.
- **Application View Version Marker.** Marker identifying the version of the Application View associated with the data provider.
- **Data Provider Definition Name.** The name of the data provider.
- **Data Provider Definition Path.** Repository path for the data provider associated with the configuration.
- **Data Provider Definition ID.** Internal repository identifier for the data provider.
- **Data Provider Definition Version Label.** Label identifying the version of the data provider.
- **Data Provider Definition Version Marker.** Marker identifying the version of the data provider.
- **Data Provider Definition Table.** Table used from the data provider.
- **Data Provider Definition Key.** Key used for the data provider.

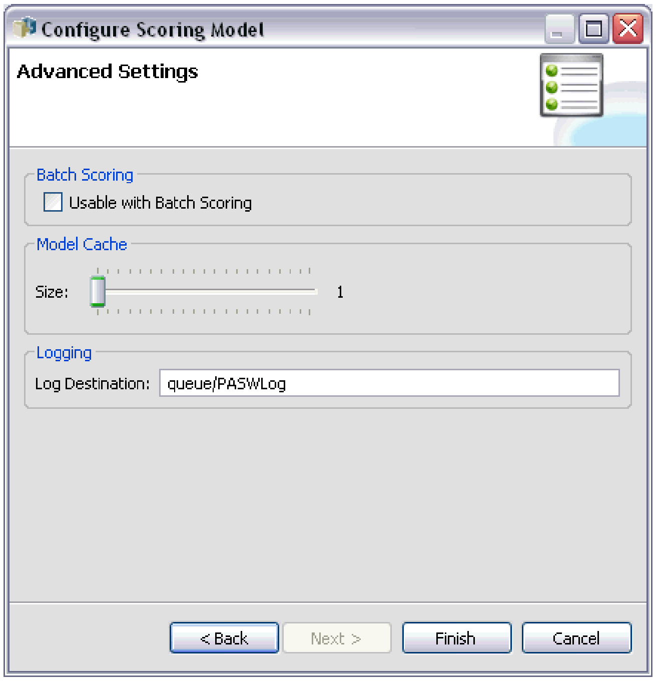
Select the items to include in the log for this particular scoring configuration. For each score request, values for all selected items will be entered into the scoring log.

Click Next to specify additional settings.

Advanced Settings

The Advanced Settings page of the Scoring Configuration wizard offers optional settings used to optimize the scoring process.

Figure 9-8
Advanced Settings



Configure Scoring Model

Advanced Settings

Batch Scoring

□ Usable with Batch Scoring

Model Cache

Size: 1

Logging

Log Destination: queue/PASWLog

Usable in Batch Scoring. If selected, the scoring model configuration can be used in batch scoring requests.

Model Cache Size. The size, in MB, of the cache used for models. To optimize performance, keep the cache size as low as possible. A value equivalent to the expected number of concurrent scoring requests for the configuration is recommended.

Log Destination. By default, PASW Collaboration and Deployment Services uses the Java Message Service (JMS) for logging scoring information to the *PASWLog* queue. If your environment is configured to use a custom message driven bean for logging, specify the log destination for that bean. Contact your administrator for the correct destination for your environment.

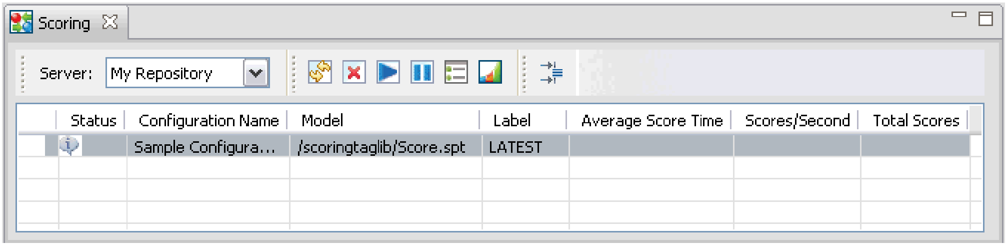
Click Finish to create a scoring configuration with the specified settings.

Scoring View

The Scoring View lists all of the scoring configurations currently defined in the system. From this view, configurations can be modified or deleted from the system. To access the Scoring View from the menus, select:

View
  Show View
    Scoring

Figure 9-9
Scoring View



Server: My Repository

	Status	Configuration Name	Model	Label	Average Score Time	Scores/Second	Total Scores
	ⓘ	Sample Configura...	/scoringtaglib/Score.spt	LATEST			

By default, configurations on the current server appear in the view. To access other servers, select the desired server from the Server drop-down list. If you are not logged into the selected server, the Login dialog box opens, prompting you to log into the server.

Each row in the view corresponds to a scoring configuration for the server. Properties shown for each configuration include:

- **Status.** The current status for the configuration.
- **Configuration Name.** The name of the scoring configuration.
- **Model.** The name of the model file associated with the configuration.
- **Label.** The label identifying the version of the model file configured for scoring.
- **Average Score Time.** The average time for a single score to be calculated.
- **Scores/Second** The number of scores produced per second.
- **Total Scores.** The total number of scores generated using the configuration.

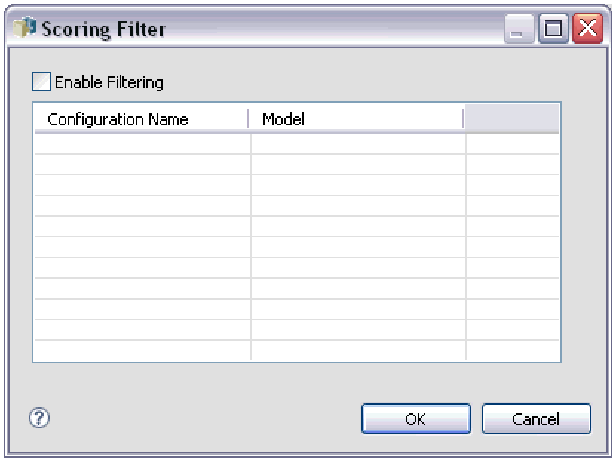
The configuration list can be refreshed to update the runtime scoring statistics. To refresh the view, click the Refresh icon.

The list can be sorted by any column in the view. Click a column heading to sort by that column. Click the heading again to reverse the sort direction.

Filtering the Scoring View

The Scoring view can be filtered to reduce the number of results that appear in the list. Once filtering is enabled, the specified filters persist throughout the Deployment Manager session until changed. In addition, filter settings persist across server connections. For example, if filtering is enabled and the server selection is changed from server A to server B, the filter settings established for server A persist for server B. To access view filters, click the Filters button. The Scoring Filters dialog box opens.

Figure 9-10
Scoring View Filters



Enable Filtering. If selected, the specified filters are applied to the view.

The view can be filtered by the configuration name. Select the configurations to view in the Scoring view.

Editing a Scoring Configuration

To edit a scoring configuration:

1. In the Scoring View, right-click the configuration to be modified.
2. Select Edit from the pop-up menu. The Scoring Configuration wizard appears.
3. Modify the configuration settings as needed.
4. Click Finish.

Suspending and Resuming Scoring Configurations

At times, it may be necessary to temporarily suspend a scoring configuration from processing requests. Suspending a configuration frees associated system resources and allows various maintenance tasks to be performed without affecting incoming score requests. Resuming a suspended configuration allows the configuration to begin processing scoring requests again.

To suspend a scoring configuration:

1. In the Scoring View, select the configuration to be suspended. To select multiple configurations, press the Ctrl key while selecting.
2. Right-click a selected configuration and select Suspend Configuration(s) from the pop-up menu. Alternatively, press the Suspend Configuration(s) button on the scoring view.

The status for the selected configurations updates accordingly.

To enable a suspended configuration to accept scoring requests, the configuration must be resumed. To resume a scoring configuration:

1. In the Scoring View, select the configuration to be resumed. To select multiple configurations, press the Ctrl key while selecting.
2. Right-click a selected configuration and select Resume Configuration(s) from the pop-up menu. Alternatively, press the Resume Configuration(s) button on the scoring view.

   The status for the selected configurations updates, indicating the configurations are active.

Deleting a Scoring Configuration

To delete a scoring configuration:

1. In the Scoring View, select the configuration to be deleted. To select multiple items, hold down the Ctrl key and select additional rows.
2. Click the Delete button. The Delete Confirmation dialog box appears.
3. Click OK. The configuration is removed from the system.

   Alternatively, right-click the configuration and select Delete from the pop-up menu.

   Deleting a scoring configuration does not remove the model associated with the configuration from the system. Furthermore, any other configurations using the model remain in the system until specifically deleted.

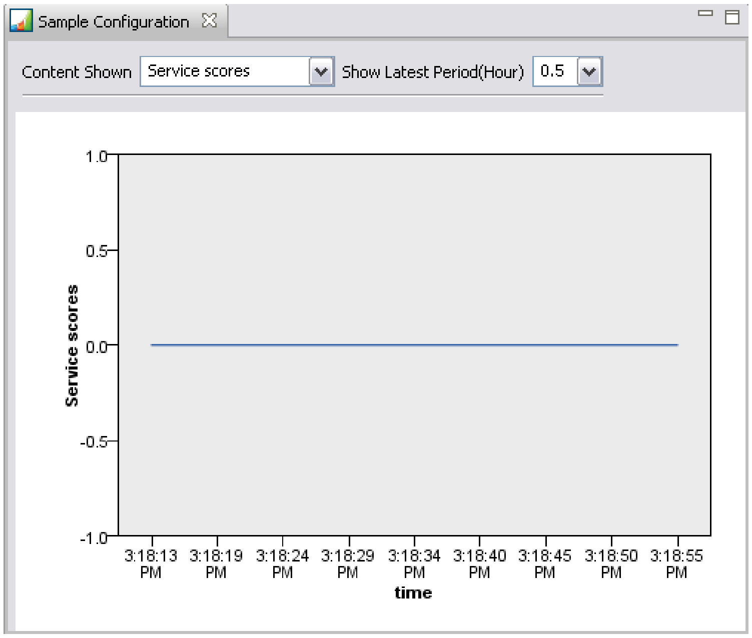
Scoring Graph View

The Scoring Graph view displays the scoring throughput for a selected scoring model. To access this view:

1. In the Scoring View, select the configuration to be viewed as a graph.
2. Click the Graphic View button.

   Alternatively, right-click the configuration and select Graphic View from the pop-up menu. The Scoring Graph view for the configuration appears.

Figure 9-11
Scoring Graph View



Content Shown: Service scores

Show Latest Period(Hour): 0.5

The appearance of the graph is defined by the following controls:

Content Shown. Select the measurements appearing in the graph.

Show Latest Period (Hour). Limits the graph to the most recent time period. Select the size of the period as the number of hours to show. For example, to show the past half-hour, specify a value of *0.5*. To show the past two hours, specify a value of *2*.

The graph in the view is regenerated at regular intervals, with the most recent information taking precedence over older results.

The Scoring Graph view displays the performance for a single selected configuration. To monitor multiple configurations, open multiple graph views.

Chapter
10

Jobs

What Is a Job?

A job is a container for a set of steps. Each step has parameters associated with it. Before you execute a step, you must embed it within a job. In order to generate results, a job must contain at least one step. (Although an empty job will run, it does not produce any results.) Jobs are scheduled and executed by an application server.

Job steps can be executed sequentially or conditionally. For example, you can make the second step in a job contingent upon the outcome of the first step in a job. Objects stored in the repository can provide input for a job step, and job results can be stored in the repository. For example, a job might include a data preparation step, which reads data from a data mart, followed by a step in which a PASW Modeler stream reads the prepared data and computes a propensity score based on the data.

Jobs are created in the Content Explorer and modified in the Job Editor. Jobs are stored in the database associated with the repository.

You can move jobs anywhere within the Content Repository. However, unlike other files, jobs are native and unique to the system. You must create and work with the job in the Deployment Manager.

Jobs are executed on a server. The content of the job determines the type of execution server required. For example, if a job contains a PASW Modeler stream, a PASW Modeler server is required. The server can reside on the same system as the PASW Collaboration and Deployment Services server. Alternatively, certain jobs can be executed on remote servers. For more information, see the topic Server Definitions in Chapter 6 on p. 96.

Versioning and Labeling Jobs

Previously, a job could only have one schedule associated with it. In this release, a job can be versioned, just like any other content object in the repository. In addition, a job can have many schedules associated with it. However, although a job can have many schedules, a schedule can contain only one job.

The following guidelines apply:

- Each time that changes are saved to a job, a new version of the job is created.
- Only a labeled version of a job can be scheduled.
- The system applies the *LATEST* label to the most recent version of a job. If a label is not explicitly supplied, the only option when scheduling a job is the version labeled *LATEST*.

Components of a Job

A job can contain any combination of the following components:

- Report files. For more information, see the topic Reporting Job Steps in Chapter 14 on p. 294.
- Visualization report files. For more information, see the topic Visualization Report Job Steps in Chapter 15 on p. 302.
- SAS syntax files. For more information, see the topic SAS® Job Steps in Chapter 16 on p. 309.
- General job steps. For more information, see the topic General Job Steps in Chapter 17 on p. 317.
- Message-based job steps. For more information, see the topic Message-Based Job Steps in Chapter 18 on p. 333.
- Notification job steps. For more information, see the topic Notification Job Steps in Chapter 19 on p. 335.

In addition, products that collaborate with PASW Collaboration and Deployment Services may offer additional components that can be included in a job, including:

- PASW Modeler streams.
- Scenario files.
- Champion challenger job steps.
- PASW Statistics syntax files.

To include these components, install the necessary adapters and plug-ins as described in the documentation for the product offering the additional functionality.

Prerequisite to Running Jobs

For certain types of jobs, you must establish server and credential definitions:

- Server definitions are required for PASW Modeler streams, PASW Statistics syntax, and SAS syntax job steps.
- Credential definitions are required for PASW Modeler stream job steps.

For more information, see the topic Resource Definitions in Chapter 6 on p. 82.

External File Dependencies

Job steps often refer to external sources, such as data files or databases. For those jobs to complete successfully, the external sources must be accessible to the repository. For example, consider the following PASW Statistics syntax that references the data file *Employee data.sav*:

```
GET FILE='C:\Program Files\data\Employee data.sav'.
GRAPH
 /BAR(SIMPLE)=MEAN(salary) BY jobcat .
```

If this syntax is used as an PASW Statistics job step, the data file must exist in the *C:\Program Files\data* directory on the repository computer. If not, the job will fail.

An alternative approach for external files involves storing them on a network location that the repository can access. This could involve mapping to a shared drive on a remote machine or using a UNC file reference.

Another method for dealing with external files involves storing the files in the repository. Use a General job step to extract them to a location necessary for subsequent steps. For example, if the repository contains *Employee data.sav*, a General job step can place it in the *C:\Program Files\data* directory for a subsequent PASW Statistics step.

Finally, if steps make use of databases, an ODBC source for the database must be defined on the repository computer.

Job Process Overview

Although the components of a job vary, the basic process for working with jobs consists of the following tasks:

1. Create a new job or open an existing job.
2. Add steps to the job.
3. Specify the relationship between job steps, if applicable.
4. Save the job.
5. Test the job by running it immediately (optional).
6. Schedule the job for execution or run the job immediately.
7. Specify e-mail notifications (optional).
8. Save changes to the job.
9. View job status (optional).
10. View output results from the job (optional).

Working with Jobs in the Content Explorer

Typically, the Content Explorer is the starting point and the end point for jobs in the system. When you first work with a job, you either create or select the job in the Content Explorer. After you have created and saved the job, you can access it again in the Content Explorer.

Creating New Jobs

Before you can schedule any step for execution, you must create a job. You can create an empty job and add steps to it later; however, to execute a job, your job must contain at least one step.

To make jobs easier to find, it is recommended that you have a designated place for storing them. For example, you can create a separate folder for jobs, or you can store jobs in the same folder as their respective model files. You must save a job within a folder. You cannot save a job within another object.

As you create jobs, it is important to note the following:

- By default, when you create a job, it is automatically saved in the repository. You need to explicitly save the job again after you add steps to it.
- If the Job option is disabled when you try to create a job, this means that you selected an item in the Content Explorer that is not a folder. For example, if you click on a PASW Modeler stream file, the New > Job option is disabled because you cannot store a job within a PASW Modeler stream.
- Although you can house jobs almost anywhere within the Content Explorer, you cannot store jobs within the Resource Definitions folder. For more information, see the topic Resource Definitions in Chapter 6 on p. 82.

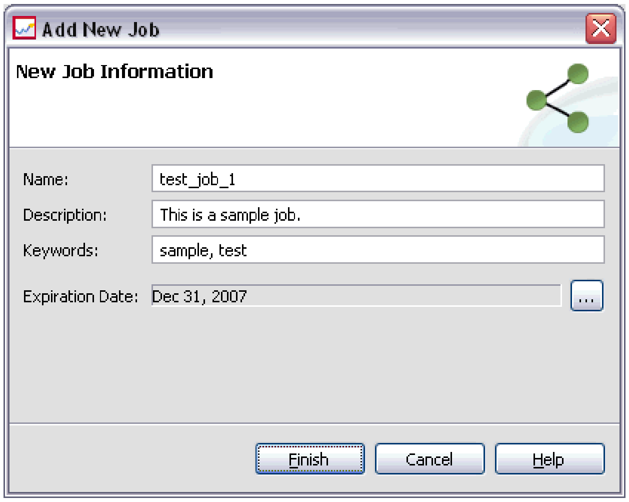
To create a new job:

1. In the Content Explorer, create a folder to house your job, if it does not already exist.
2. Select the folder in which you want to store your job.
3. From the File menu, choose:

   New
     Job

   The New Job Information dialog box opens.

   Figure 10-1
   Creating a new job

   Name: test_job_1
   Description: This is a sample job.
   Keywords: sample, test
   Expiration Date: Dec 31, 2007

4. In the New Job Information dialog box, provide the following information:

   Table 10-1
   New job information

Field	**Description**	**Required or optional**
Name	You must specify a unique name for your job. The name must be unique within the folder. For more information, see the topic Naming Conventions Within the System in Chapter 3 on p. 14.	Required

Field	Description	Required or optional
Description	This field contains a description of your job. Although this field is optional, it is recommended that you provide a detailed description for your job because the description can be searched. For example, if your description contains the phrase *propensity analysis*, and you search for that term, this job appears in the search results.	Optional
Keywords	The keywords that you specify here are used by the search facility. For more information, see the topic Searching in Chapter 4 on p. 22.	Optional
Expiration date	An expiration date can be specified for the job. For more information, see the topic Working with Expiration Dates and Expired Files in Chapter 5 on p. 50.	Optional

5. After you have specified the job details, click Finish. The new job appears in the folder that you selected previously in the Content Explorer and as a tab in the Job Editor.

   You are now ready to add steps to your job. For more information, see the topic Adding Steps to a Job on p. 236.

Opening an Existing Job

To open an existing job:

▶ In the Content Explorer, navigate to the job and double-click on it. The steps within the job appear in the Job Editor.

Viewing Job Properties

Properties for jobs are similar to properties for other objects in the Content Explorer. For more information, see the topic Viewing Object Properties in Chapter 5 on p. 34.

Working with the Job Editor

The Job Editor, which is located in the upper right pane, is where you modify jobs. You can perform the following tasks:

- Add steps to a job.
- Specify parameters for a job step.
- Establish relationships between steps.
- Schedule a job for execution.
- Specify notifications.
- View job status.

The Job Editor is divided into the following sections:

Job canvas. The job canvas provides a visual representation of your job. You add steps to your job in the job canvas. You can drag and drop files from the Content Explorer and select job step types from the Job Steps list. In addition, you can establish relationships between items within a job.

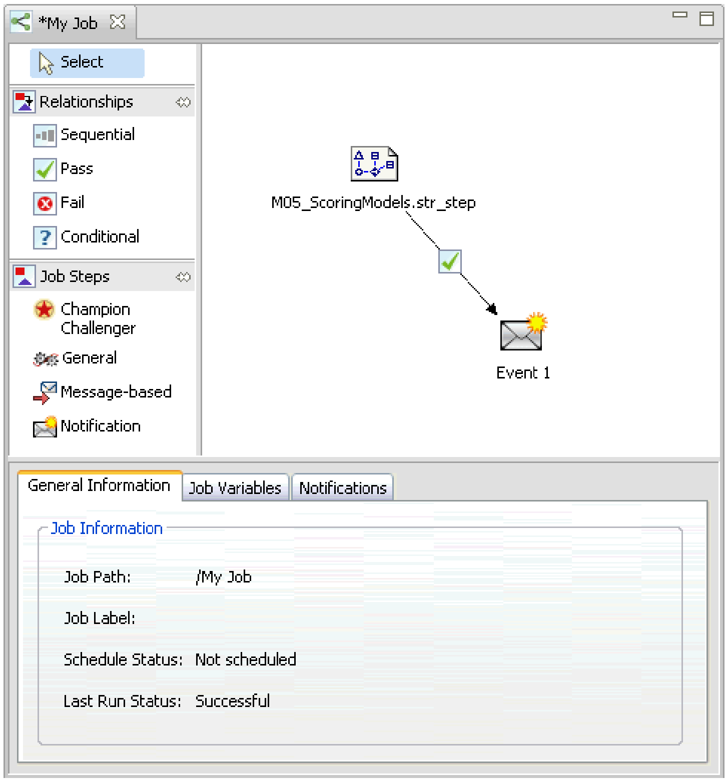
Job palette. The job palette is divided into two main sections:

- **Relationships.** Using the tools in the Relationships section, you can establish relationships between the steps in your job. For more information, see the topic Specifying Relationships in a Job on p. 237.
- **Job Steps.** In the Job Steps section, you can add several types of job steps to your job—for example, champion challenger, general, message-based, or notification job steps.

Job properties section. This section is divided into the following tabs:

- **General Information.** Describes general attributes associated with the job. For more information, see the topic General Information Tab on p. 234.
- **Job Variables.** Defines variables for the job whose values can be passed to steps within the job. For more information, see the topic Job Variables on p. 234.
- **Notifications.** Allows you to specify e-mail notifications for job failure and job success. For more information, see the topic Job Success and Failure Notifications in Chapter 13 on p. 278.

Figure 10-2
Job Editor



Job Path: /My Job

Job Label:

Schedule Status: Not scheduled

Last Run Status: Successful

The General Information, Job Variables, and Notifications tabs apply to the entire job. Each step within a job has its own tabs associated with it. When you click on a job step, the job processing tabs change accordingly.

General Information Tab

When you first open a job, the General Information tab appears by default. You cannot modify any items within the General Information tab. The information on this tab is generated and updated by the application, based on information that is provided at other points in the job creation process.

The General Information tab contains the following details about the job:

Job Path. This field specifies the directory location for the job.

Job Label. This field contains the label that has been applied to the job. If no label has been applied, this field remains blank.

Schedule Status. This field describes whether or not the job has already been scheduled. Valid values are *Scheduled* or *Not scheduled*.

Last Run Status. This field describes whether the job ran successfully or failed the last time it was run.

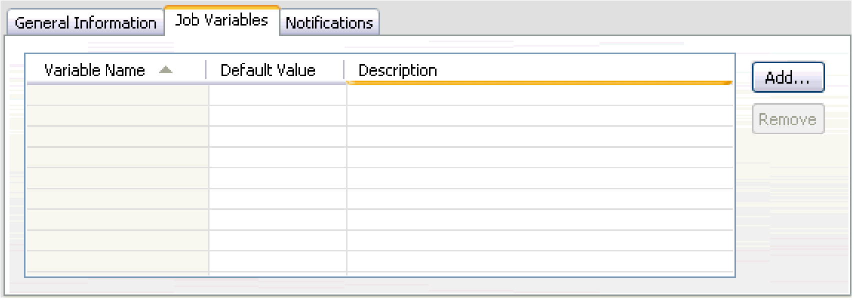
Job Variables

Job variables define parameters whose values can be passed to any step within the job. Using variables, any job can be used as an iterative consumer, in which values external to the job can be used to control job processing. Values for the variables can be defined:

- when initiating the job
- in schedules associated with the job
- in other jobs executing before the job

The Job Variables tab for a job displays a table identifying variables defined for a job.

Figure 10-3
Job variables



Variable Name ▲	Default Value	Description

Variable Name. Lists the names of variables defined for the job.

Default Value. Identifies the default value for each job variable. If a variable has no specified default value and a value has not been assigned in some other fashion, the user is prompted for a value during job execution.

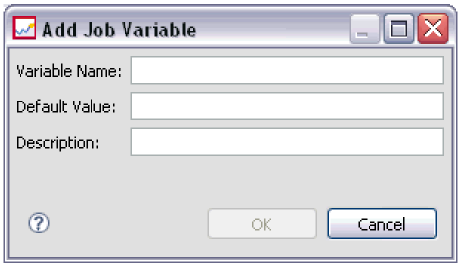
Description. Lists informative text about each variable typically used to aid in identifying the variables.

Job variables can be used in any job step field that supports entry field content assist. In addition to system properties, the list of available variables appearing when typing a $ in those fields includes all variables defined at the job level.

Adding Variables to a Job

The Add Job Variable dialog box creates new job variables. To define a new variable for a job, click Add on the Job Variables tab for an open job.

Figure 10-4
Add job variable dialog box



Add Job Variable

Variable Name:

Default Value:

Description:

1. Type a unique name for the variable. No two variables for the same job can have the same name.
2. Type the value of the variable to use as the default.
3. Type a description for the variable.
4. Click OK.

The new variable appears in the variable list for the job.

Editing Existing Job Variables

To modify an existing variable for a job:

1. Open the job.
2. Select the Job Variables tab.
3. In the variables table, select the cell containing the value to be changed. The variable name cannot be modified.
4. Modify the value as desired.
5. Press the Enter key.

The updated information appears for the variable in the list.

Removing Variables from a Job

To remove an existing variable from a job:

1. Open the job.
2. Select the Job Variables tab.
3. Select the variable to remove.
4. Click Remove.

The variable is removed from the job and no longer appears in the variable list.

Adding Steps to a Job

After you have created or selected a job, you are ready to add steps to it. For more information, see the topic Components of a Job on p. 229.

To add steps to a job:

1. Select the job. The job appears in the Job Editor.
2. Select the item that you want to add to the job. You have the following options:
   - **Add a file.** From the Content Explorer, click on the file you want to add, and then drag and drop the file into the job canvas. Alternatively, you can right-click on the file and select Add to Job.
   - **Add a job step.** From the Job Steps section of the job palette, select the type of job step that you want to add. The option you select will be highlighted. Then click anywhere in the job canvas. By default, the system numbers steps in the order in which they appear. Thus, subsequent job steps appear as Event 2, Event 3, and so on. You can rename a step at any time. The number has no effect upon the order in which steps are executed.
3. If you want to add additional files or commands, return to step 2.
4. If you have two or more steps in your job, you can either:
   - Run job steps concurrently. For more information, see Executing Job Steps Concurrently below.
   - Establish relationships between the steps. For more information, see Specifying Relationships in a Job below.

Many job steps reference repository resources that may have multiple versions. These steps reference a particular version of the resource by label. You must have adequate authority to the label selected for the job step to execute the job. If your authority does not permit referencing the job step resource by the selected label when the job runs, the system will be unable to find the resource and the execution will fail. For more information, see the topic Label Security in Chapter 5 on p. 56.

Executing Job Steps Concurrently

If you create a job with multiple steps but you do not specify a relationship between the steps, the system will execute the steps concurrently. Thus, the steps are not contingent upon one another.

Specifying Relationships in a Job

When you add steps to a job, you have the option of establishing a relationship between the steps. The relationships that you specify determine the sequence of steps in the job and the conditions under which each step is executed.

Relationships are established by relationship connectors. When you connect two steps, there is an antecedent step and a consequent step. The antecedent step is the starting point for the relationship. This step, as its name suggests, is executed first. The consequent step is contingent upon the completion of the antecedent step. It is important to note that completion does not always mean successful completion. For example, if you specify a fail relationship between two steps, the consequent step is executed only if the antecedent step fails to complete successfully.

For example, suppose you have four steps in your job: A, B, C, and D. A simple scenario for the four steps would be to execute them sequentially—that is, execute A first, then B, then C, and then D. In another case, you might want to execute step A and then execute step B only if step A succeeds. If step A fails, you might want to execute step C. In both cases, you would want to generate a report in step D.

If you cancel a job, the relationships that you establish will not be followed. For example, suppose that you have a job with steps A and B with a sequential relationship between them. If you run the job and cancel it while step A is running, step B will not execute. This behavior holds true for all relationship types, including a fail relationship.

When you create a job, you can establish the following relationships between steps in the job:

Table 10-2
Relationship connectors and their corresponding icons

Relationship connector	Icon	Conditional expressions allowed?
Sequential	▮	No
Pass	✓	No
Fail	⊗	No
Conditional	?	Yes. In order to use a conditional connector, you must specify at least one argument.

Although the relationships that you can establish between steps vary, the process for establishing a relationship is the same.

To specify a relationship between two steps in a job:

1. Open an existing job. The job appears in the Job Editor.

2. From the palette, select the relationship you want to establish. The relationship is highlighted on the palette.
3. In the job canvas, click on the antecedent step in the sequence and then click on the consequent step in the sequence. When the mouse is released, a directional arrow connects the two steps. The icon in the center of the arrow describes the relationship.
4. To establish another relationship within the job, return to step 2.
5. Save your changes.

Sequential Connector

The sequential connector is always executed. If you connect two steps using the sequential connector, the consequent step is executed immediately after the antecedent step is completed. Unlike pass, fail, or conditional relationships, there are no conditions associated with a sequential connector. The only criterion for the execution of the consequent step is the completion of the antecedent step.

Pass Connector

The pass connector is executed if the antecedent step is executed successfully. If the antecedent step is completed successfully, the system proceeds to the consequent step. If the antecedent step fails to complete successfully, the system will not execute the consequent step.

Fail Connector

The fail connector is executed if the antecedent step does not execute successfully. If the antecedent step is executed successfully, the consequent step in a fail relationship is not executed. For example, the consequent step in a fail relationship may generate a report indicating that the antecedent step failed.

Conditional Connector

The conditional connector is executed if the antecedent step satisfies the condition(s) you specify. For conditional relationships, you must specify an expression that contains the condition. This is the only relationship connector in which you can pass arguments to the system.

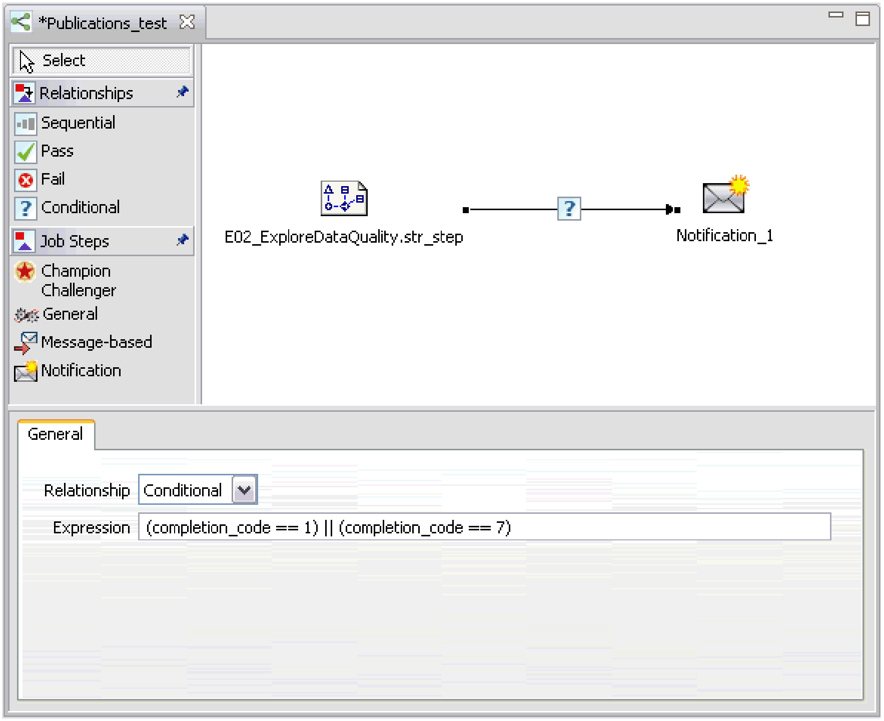
The Conditional Connector Expression can be any script the system can evaluate, but there are two special parameters that relate to the consequent step:

Completion_code. The `completion_code` is an integer value. When the `completion_code` expression is used, it must be lowercase. The job step interprets the meaning of the completion code. For the General job step, the completion code is the executable return code. For example, for most Windows commands, `File Not Found = 2`.

Success. The `success` parameter is a Boolean, where `true` indicates that the job step execution was successful. When the `success` expression is used, it must be lowercase.

You can compare and combine these two parameters with the standard scripting operators.

Figure 10-5
Expression in a conditional connector



Relationship: Conditional

Expression: (completion_code == 1) || (completion_code == 7)

You can compare and combine parameters, using standard scripting parameters. For example,

- &&
- ||
- ==
- !=
- <
- >
- !

Warning Expressions in Conditional Connectors

A special case of using conditional job step connectors is specifying a relationship based on the warning code returned by the antecedent step. The following job step types support warning codes:

- General
- PASW Modeler
- Scenario
- PASW Statistics
- SAS

To specify a warning-code-based relationship:

1. Connect the steps with a conditional connector.
2. Enter `warning==true` as the conditional expression. When the `warning` expression is used, it must be lowercase.
3. Open the parent job step and override the default warning code on the General tab if necessary.

   Once the antecedent step is completed, the system will evaluate the conditional expression and proceed to the next step if `true` is returned.

Viewing General Relationship Properties and Modifying Relationships

Each object in the job canvas has properties associated with it. The relationship between steps is described on the General tab. At times, you may want to modify the relationship between steps in a job.

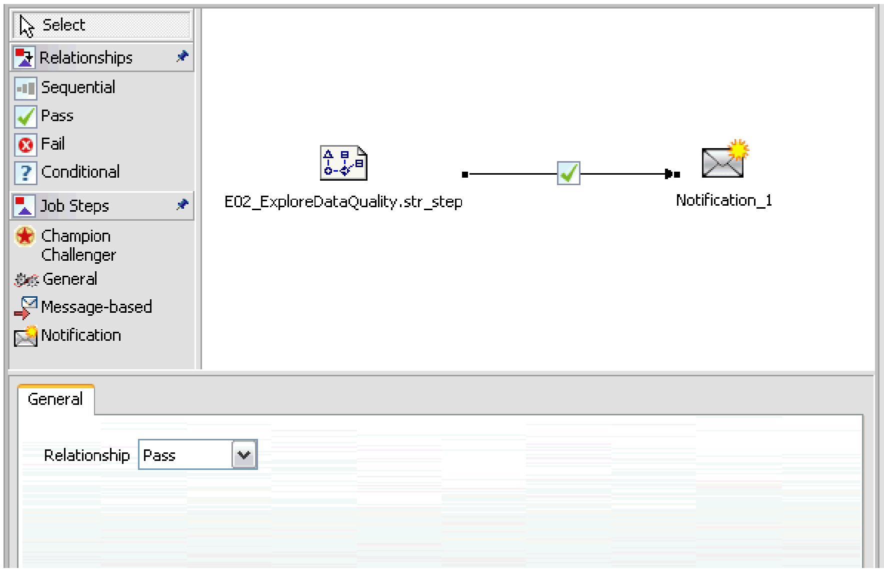
Viewing relationship properties. To view the properties for a relationship:

- ▶ In the Job Editor, click on the relationship icon. Two small black boxes appear at each end of the arrow to indicate that the connector has been selected. The General tab appears in the job properties section.

Modifying relationship properties on the General tab. To modify a relationship:

1. Select a new relationship from the Relationship drop-down list. The connector icon changes to reflect the new relationship.
2. If the new relationship requires that you provide parameters, you must specify the parameters. For example, if you change a pass connector to a conditional one, you must specify the parameters that describe the condition. For more information, see the topicConditional Connector on p. 238.
3. Save your changes.

Figure 10-6
General relationship properties



Relationship: Pass

Modifying relationship properties in the job canvas. Alternatively, you can modify relationships within the job canvas. To do so:

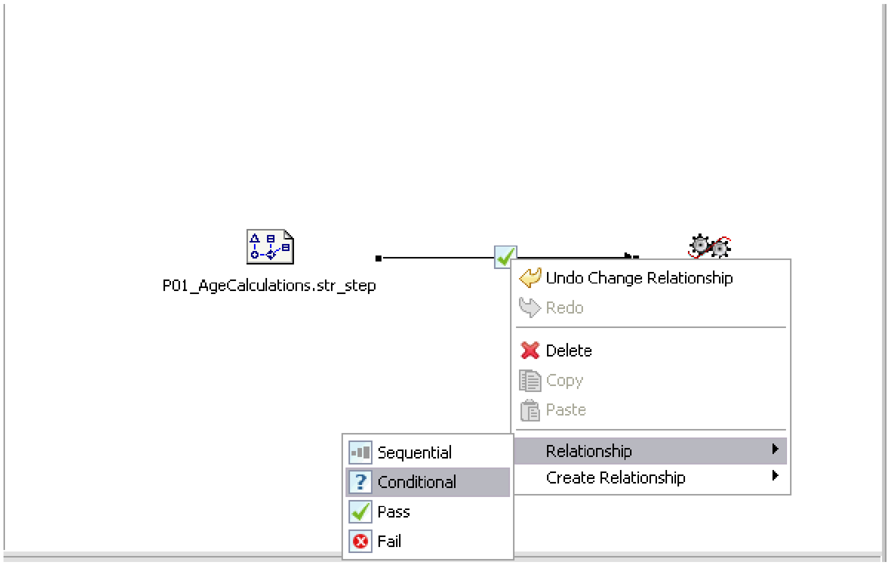
1. In the Job Editor, right-click on the arrow connecting the two steps whose relationship you want to modify. Two small black boxes appear at each end of the arrow to indicate that the connector has been selected.

2. Select Relationship.

3. Choose the new relationship you want to establish.

4. Save your changes.

Figure 10-7
Modifying a relationship in the job canvas



Deleting Relationships in a Job

At times, you may want to delete relationships between steps in a job. If you delete a step from a job, any relationships that were established between the deleted step and other steps in the job are automatically deleted.

If you simply want to modify a relationship, you do not need to delete it. For more information, see the topic Viewing General Relationship Properties and Modifying Relationships on p. 240.

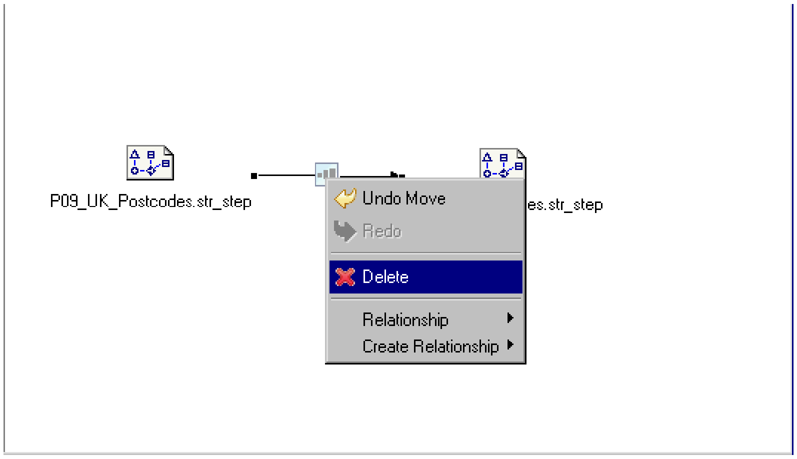
To delete a relationship between two steps:

1. In the Job Editor, right-click on the connector icon. Two small black boxes appear at each end of the arrow to indicate that the connector has been selected.

2. Select Delete. The relationship is deleted.

3. Save your changes.

Figure 10-8
Deleting a relationship within a job



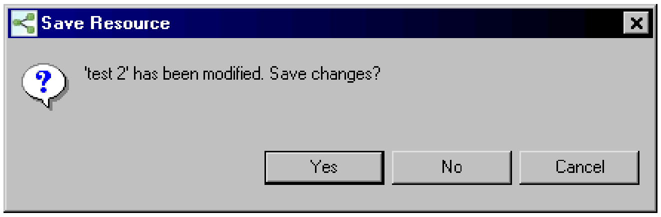
Saving Jobs

It is recommended that you save changes to your jobs regularly. Changes that you make while working with jobs are not committed to the system until you save the job. In the Job Editor, a job containing unsaved changes is preceded by an asterisk in its name.

If you attempt to close a job before you have saved your changes, the system prompts you to save changes. The dialog box that opens depends on how you chose to close the job.

Closing a job in the Job Editor. If you attempt to close a job with unsaved changes in the Job Editor, a Save Resource dialog box opens.

Figure 10-9
Saving changes to a job when exiting the Job Editor



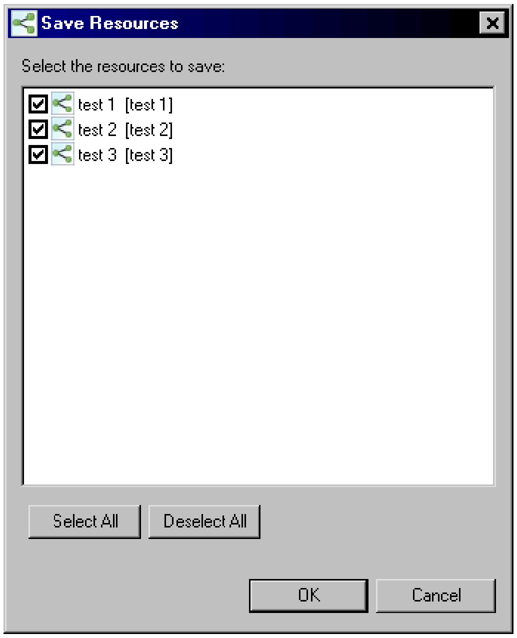
'test 2' has been modified. Save changes?

You have the following options in this dialog box:

- To save your changes and close the job, click Yes.
- To disregard your changes and close the job, click No.
- To exit the dialog box and return to the job, click Cancel. The Job Editor reopens.

Exiting the system. If you attempt to exit the system and you have unsaved jobs in the Job Editor, a Save Resources dialog box opens. This dialog box provides a list of the jobs with unsaved changes in them.

Figure 10-10
Saving changes to a job when exiting the system



Select the resources to save:

☑ test 1 [test 1]
☑ test 2 [test 2]
☑ test 3 [test 3]

To save the changes:

1. Select the check boxes next to the jobs with changes you want to save. To expedite the process, click Select All to save changes in all of the jobs in the list, or click Deselect All to discard changes in all of the jobs in the list.

2. Click OK.

Exiting the System Without Saving Changes

To exit the system without saving changes to any of the jobs in the list:

▶ Click Cancel.

Changes to Files Within a Job

It is important to note that if any changes are made to the files in a job—for example, a PASW Modeler stream (*.str*) or a scenario file (*.scn*), any job that contains the file is affected. When changes are made to the file, a new version of the file is saved to the repository. However, the job that contains the file is not automatically updated with the modified file. To incorporate the file updates into the affected job:

1. Reopen the job. When the job is reopened, an asterisk appears beside the job name in the job canvas, indicating that the job contains unsaved changes.

2. Resave the job.

Job Step Results

Most types of Deployment Manager job steps generate results in the form of output files. The files can be saved in the repository or in the file system running the application that generated them (for example, a system running PASW Statistics server). Certain types of job steps can generate multiple output files. Most types of job steps allow you to select output formats, such as HTML, text, or PDF.

Job step output settings are displayed on the Results tab of the Job Editor for PASW Statistics, SAS, PASW Modeler, and reporting job steps and on the Output Files tab for general job steps. The Results tab and available options are different for different job step types. , , *Reporting Job Steps* on p. 298, *SAS® Job Steps* on p. 311, and *General Job Steps* on p. 320.

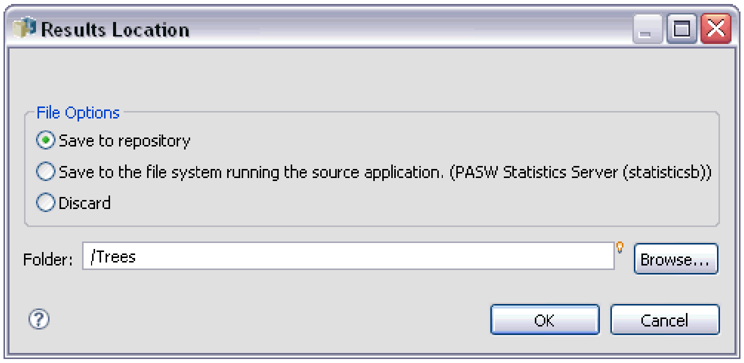
Regardless of the job step type, the following output file settings can be modified:

- Output file location
- Output file permissions
- Output file metadata

Output File Location

Click the ellipsis button in a selected Location cell to define the location for the results.

Figure 10-11
Results properties location



Results Location

File Options

● Save to repository

○ Save to the file system running the source application. (PASW Statistics Server (statisticsb))

○ Discard

Folder: /Trees

Save to repository. Saves output files to the repository in the specified folder. Click Browse to select the folder in which to save the output. If the files already exists, new versions are saved.

Save to the application server file system. Saves output files to the application server running the repository. In the Folder field, type the name of the application server folder in which to save the file.

Discard. Job output is discarded.

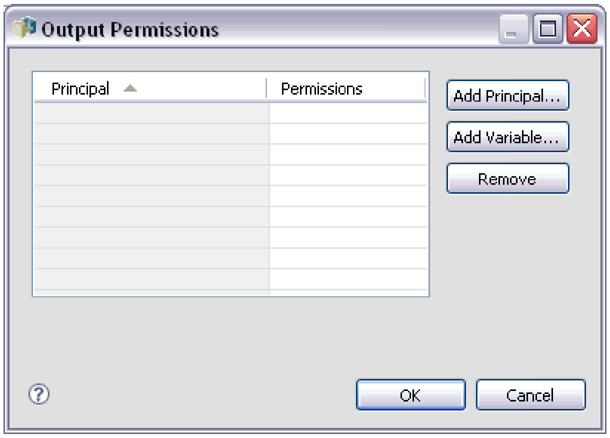
Note: For Champion Challenger steps, the *Discard* option in this dialog is always disabled.

Output File Permissions

When a job is run, any output generated by the job is owned by the user that ran the job. Output file permissions for additional users or groups can be set in the Output Permissions dialog box. This dialog can be used to:

- Add a new principal. For more information, see the topic Adding New Users or Groups in Chapter 5 on p. 42.
- Add a principal using a variable. For more information, see the topic Add Principal By Variable on p. 247.
- Delete an existing principal. For more information, see the topic Deleting a User or Group from the Permissions List in Chapter 5 on p. 46.
- Modify permissions for an existing principal.

Figure 10-12
Output File Permissions dialog box



Accessing the Output Permissions dialog box varies by the job step type.

- **PASW Statistics, SAS, and PASW Modeler job step output files.** In the Results tab, click the ellipsis in the Permissions column next to the file entry.
- **Reporting job step output files.** Click Browse next to the Permissions field.
- **General job step output files.** On the Output Files tab, click the ellipsis in the Permissions column next to the file entry.

To modify permissions for an existing user or group:

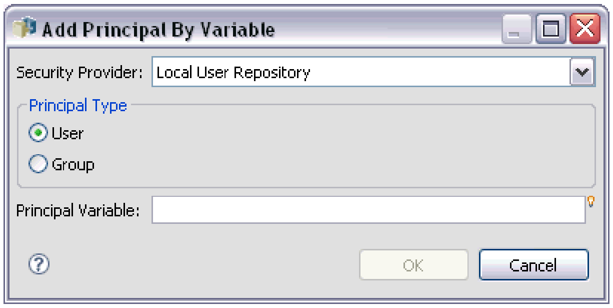
1. Select the principal from the table.
2. In the Permissions column, click the drop-down arrow and select a new permissions level from the list.
3. Click OK.

Add Principal By Variable

Job step output file permissions can be assigned dynamically by adding users and groups to the permissions list as iterative variable values. In such cases, the variables defined within the step are used to retrieve the recipient address value every time the step is executed. For more information, see the topic Type for Reporting Steps in Chapter 14 on p. 296.

To add a user or group to the permissions list, in the Output Permissions dialog box click Add Variable. The Add Principal By Variable dialog box opens.

Figure 10-13
Add Principal By Variable dialog box



Add Principal By Variable

Security Provider: Local User Repository

Principal Type

● User

○ Group

Principal Variable:

1. Select the security provider for authenticating the principal defined by the iterative variable.

2. Select User or Group as the principal type.

3. Enter the variable in the Principal Variable field. Use field content assist if necessary. For more information, see the topicEntry Field Content Assist in Chapter 3 on p. 12.

Output File Metadata

Deployment Manager provides the capability to specify metadata properties for PASW Statistics, PASW Modeler, SAS, and reporting job step output files similar to the way properties are specified for repository objects. To specify properties for a job step output file:

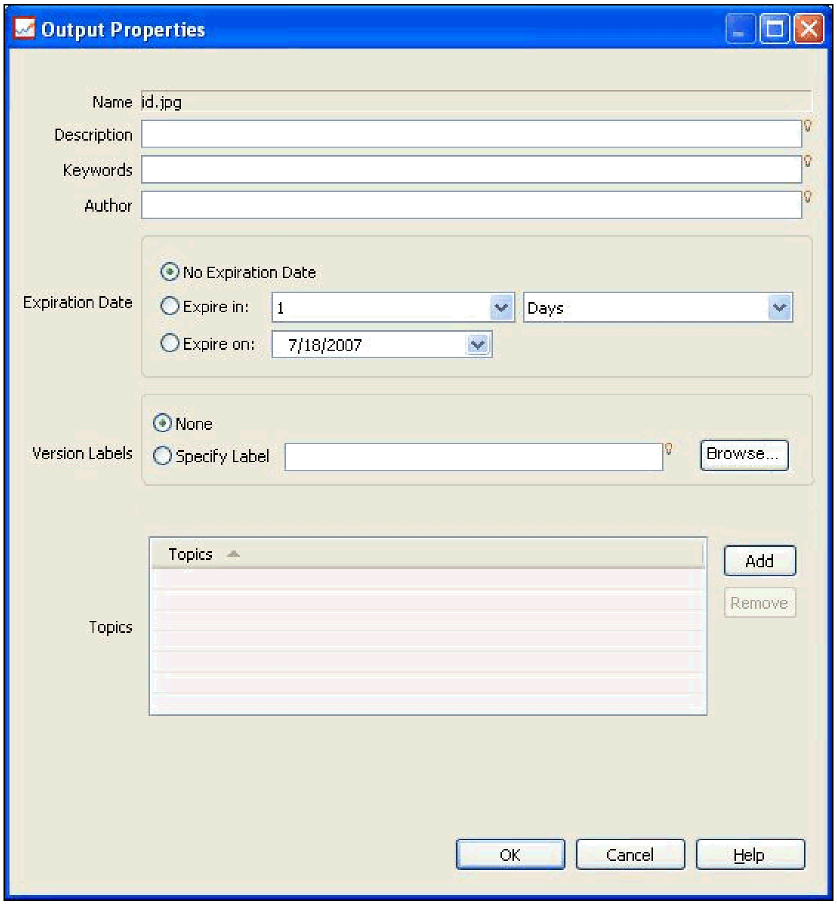
1. Click the step in an open job.

2. Click the Results tab.

3. In the list of output files, click in the Properties column next to the file. The ellipsis button is displayed. Disregard this step for reporting job steps.

4. Click the ellipsis button. The Output Properties dialog box opens. For reporting steps, click Browse next to the Metadata field.

Figure 10-14
Output Properties dialog box



Name: id.jpg

Description:

Keywords:

Author:

Expiration Date:
● No Expiration Date
○ Expire in: 1 Days
○ Expire on: 7/18/2007

Version Labels:
● None
○ Specify Label

Topics:

Topics

5. Provide the following information.

- **Description.** A user-defined description of the output file.
- **Keywords.** The metadata assigned to the output file for the purposes of content search.
- **Author.** A string identifying the author of the output file.

Note: The dialog box may also display custom properties if such properties have been defined for the server. For more information, see the topic Creating Custom Properties in Chapter 5 on p. 66.

- **Expiration Date.** The date after which the file is no longer active. By default, no expiration date is set for the job output file. You can specify a time interval from the current date for file expiration (for example, one year) or enter a specific date.
- **Version Labels.** A user-defined label for the file. By default, no label is assigned to the file. To specify a new label or select from a list of existing labels, click Browse. The Edit Version Labels dialog appears. For more information, see the topic Editing the Version Label in Chapter 5 on p. 48. If your label permissions change after defining output labels, labels may appear in the list that can no longer be applied. Job execution will fail due to an inability to assign a label. The label that cannot be assigned must be removed from the output properties

or your permissions to the label will need to be changed. For more information, see the topic Label Security in Chapter 5 on p. 56.

- **Topics.** The topics assigned to the file. Click Add to open the Add Topic dialog box and select topics. To remove a topic, select the topic in the list and click Remove. For more information, see the topic Working with Topics in Chapter 5 on p. 72.

6. To save your changes, click OK.

   Note: Fields marked with a light bulb icon support the insertion of variables to the property at run time. For example, you can append the date to an output file/folder name at run time. For burst reporting job step types, you can also insert burst variables. See Entry Field Content Assist on p. 12.

Chapter

11

Running Jobs

Running a job results in all of its job steps executing according to their relationships defined in the job. Job execution can be initiated on-demand in response to a user request or scheduled to occur in the future in response to a specified event. The job status and job history are displayed in the history view for jobs run through schedules and jobs run immediately. For more information, see the topic Job History View in Chapter 12 on p. 263. Existing time-based and message-based schedules are listed in the job schedule view. For more information, see the topic Job Schedule View in Chapter 12 on p. 262.

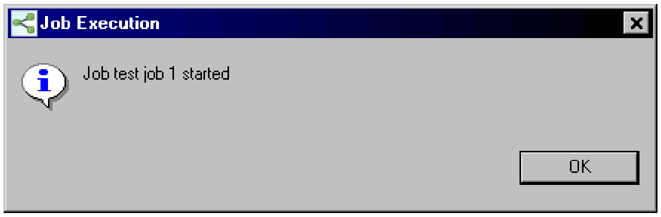
On-demand Job Execution

At times, you may want to run a job immediately after you have created or modified it. This process, which is typically used to test a job, is launched manually. To run a job immediately:

1. In the Content Explorer, select an existing job.

2. From the Tools menu, select Run Job Now. The Job Execution dialog box opens. Alternatively, you can select the Run Job Now button from the toolbar.

   Figure 11-1
   Job Execution dialog box



Job test job 1 started

3. Click OK.

The execution uses the specified default values for any variables in the job. In addition, any notifications associated with the job and its steps will be processed and delivered. To specify alternative variable values or disable notifications for the execution, use Run with Options.

It is not necessary to open a job in order to run it. You only need to select the job. If you want to view the contents of the job, double-click the job to open the Job Editor. Job execution can also be initiated from the Job Editor and the Job History view.For more information, see the topic Job History View in Chapter 12 on p. 263.

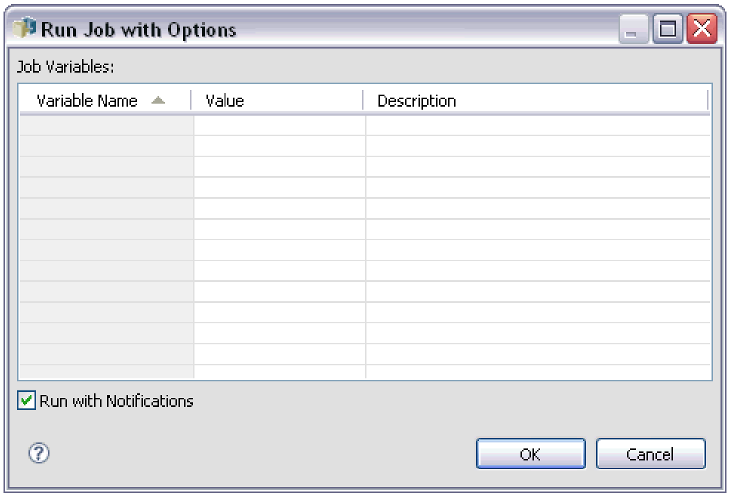
Specifying On-demand Execution Options

The default variable values and notification treatment can be overridden by specifying execution options. For example, it may be desirable to turn off notifications when testing jobs. To specify options when running a job on-demand:

1. In the Content Explorer, select an existing job.For more information, see the topicOpening an Existing Job in Chapter 10 on p. 232.

2. From the Tools menu, select Run Job with Options. The Run Job with Options dialog box opens.

   Figure 11-2
   Run Job with Options dialog box



3. Modify variable values as needed by clicking the value to be changed and typing the new value.

4. Specify whether or not the job and job step notifications should be delivered as a result of the execution.

5. Click OK.

   The job executes with the specified options.

Scheduled Job Execution

A job often needs to execute multiple times in response to future events. For example, a report created by a job may be needed every Friday. To define the execution pattern, the job must be associated with a schedule specifying the execution settings. A schedule can be either time-based or message-based.

- A time-based schedule relies on the occurrence of a specified time or date to initiate job execution. For example, a time-based schedule may run a job at 5:00 PM every Thursday.
- A message-based schedule relies on the receipt of a JMS message to trigger job execution. In this case, the job executes every time a specified message domain receives a message.

Given that a scheduled job executes at some point in the future, the schedule must use credentials defined in the system. Jobs cannot be scheduled using single sign-on (SSO) credentials because a user might not be logged in when the job executes.

Creating Schedules

To create a new schedule for a job, right-click on a job in the Content Explorer and select:

New Schedule
  Time-Based

or

New Schedule
  Message-Based

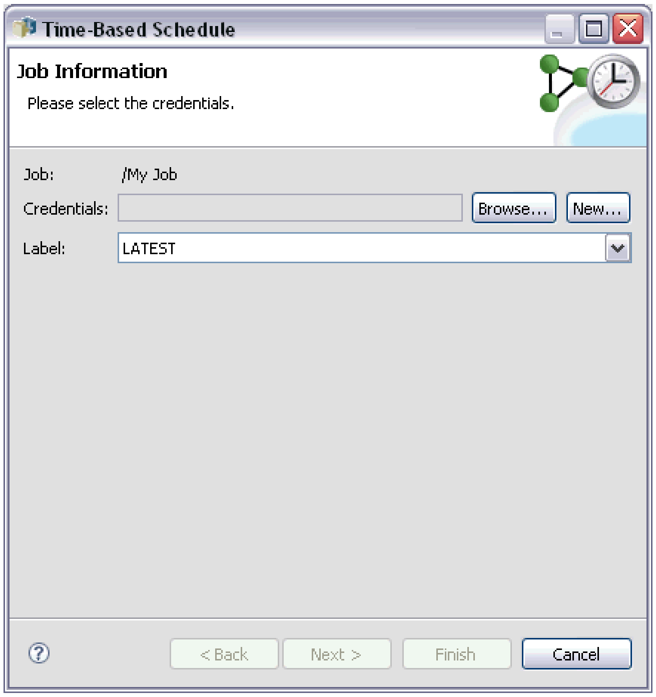
The Job Schedule wizard opens. Schedules for jobs must specify the following information:

- settings for the job being scheduled
- settings for the schedule that depend on whether the schedule is time-based or message-based
- values for job variables

Job Information

The Job Information page of the Schedule wizard identifies the version of the job associated with the schedule and the credentials used for job execution.

Figure 11-3
Scheduled Job Information



Job Information
Please select the credentials.

Job: /My Job
Credentials:
Label: LATEST

Job. The name and repository path for the job associated with the schedule.

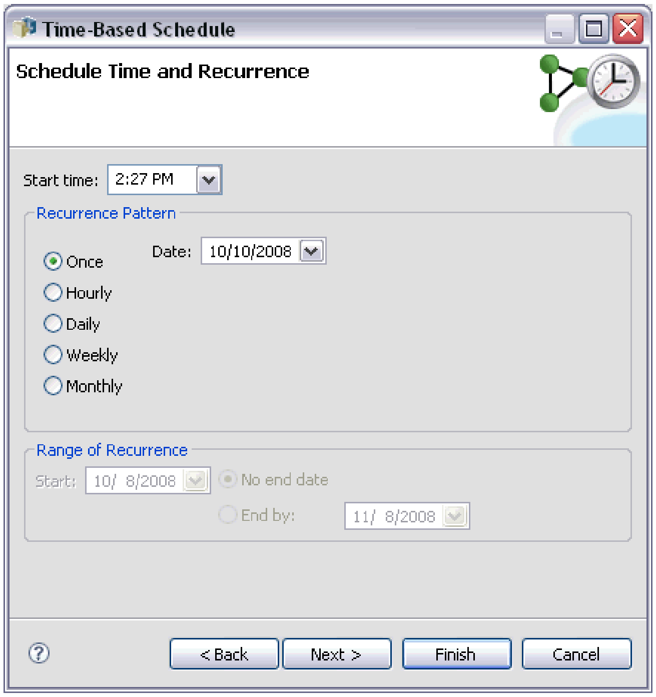
Credentials. Credentials, which specify permission levels, determine the user under which the scheduled job runs. Specifically, the system verifies credentials when writing output and saving files to the repository. Click Browse to select a credential from those defined in the system. To create a new credential for the scheduled job, click New. The specification of a credential for a job schedule is required.

Label. The label for the version of the job being scheduled. Select the desired value from the list of existing labels. The user under which the job runs must have adequate authority to the selected label when the schedule triggers job execution. If the user authority does not permit referencing the job by the selected label when the schedule triggers, the system will be unable to find the job and the execution will fail. For more information, see the topic Label Security in Chapter 5 on p. 56.

Time-Based Schedule Settings

For time-based schedules, the Schedule Time and Recurrence page of the Schedule wizard defines the time of day and frequency at which the job runs, as well as the duration of the schedule.

Figure 11-4
Time-Based Schedule Settings



Time-Based Schedule

Schedule Time and Recurrence

Start time: 2:27 PM

Recurrence Pattern

● Once Date: 10/10/2008
○ Hourly
○ Daily
○ Weekly
○ Monthly

Range of Recurrence

Start: 10/ 8/2008 ● No end date
○ End by: 11/ 8/2008

Start Time. Select the time of day at which you want to run the job from the drop-down list. To schedule a job at a time that is not available from the drop-down list (for example, 5:45 AM), type the value in the Start time field.

Recurrence Pattern. The recurrence pattern defines how often the job runs. Select the pattern and frequency from the following options:

- **Once.** The job runs one time only. From the Date drop-down list, select the date on which you want the job to run.
- **Hourly.** The job runs at hourly intervals. Specify the schedule frequency using the Recur every <value> hours field. For example, a value of 2 results in the job running every two hours.
- **Daily.** The job runs regularly at daily intervals. Specify the schedule frequency using the Recur every <value> days field. For example, a value of 3 results in the job running on every third day.
- **Weekly.** The job runs at weekly intervals on the specified day. Specify the schedule frequency using the Recur every <value> weeks field. For example, a value of 4 results in the job running on every four weeks on the designated day.
- **Monthly.** The job runs at monthly intervals on the specified day of the month. Specify the schedule frequency using the Recur every <value> months field. For example, a value of 4 results in the job running every fourth month on the designated day.

Range of recurrence. The range of recurrence is the duration for which the job runs and consists of two parts:

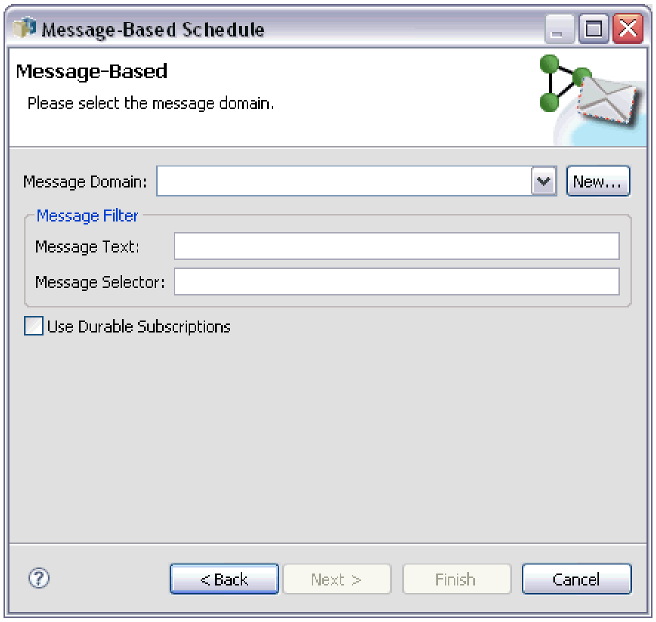
- **Start date.** The date for the initial execution of the job.
- **End date.** The date for the final execution of the job. Select No end date to have the job run indefinitely according to the specified schedule settings.

Click Next to specify values for variables used in the scheduled job.

Message-Based Schedule Settings

A message-based schedule is triggered by an outside event signalled by a JMS (Java Messaging Service) message. For example, when PASW Collaboration and Deployment Services job relies on the input from a third-party application, the application must send PASW Collaboration and Deployment Services a JMS message when the input file is ready for processing. For message-based schedules, the Message-Based page of the Schedule wizard defines the message domain and filter for the schedule.

Figure 11-5
Message-Based Schedule Settings



Message-Based Schedule

Message-Based

Please select the message domain.

Message Domain:

Message Filter

Message Text:

Message Selector:

☐ Use Durable Subscriptions

Message Domain. The message domain identifies the JMS topic to which to subscribe. Select the desired domain from the list or click New to create a new message domain.For more information, see the topic Message Domains in Chapter 6 on p. 94.

Message Filter. Optional values that must be satisfied by a message for schedule activation to occur. Filters can be based on the text and the header of the message.

- **Message Text.** For JMS text messages, text that must be included in the message to activate the schedule.
- **Message Selector.** Optional selector text of contents of the message header. For example, you can indicate that the message header must contain the ID of a specific repository resource (ResourceID=<resource ID>) or specific custom properties (NewsType='Sports' or NewsType='Business').

Use Durable Subscriptions. The option to save the messages specified for the schedule while PASW Collaboration and Deployment Services is not actively listening on the messaging service, so that the saved message can be retrieved later when the system starts listening.

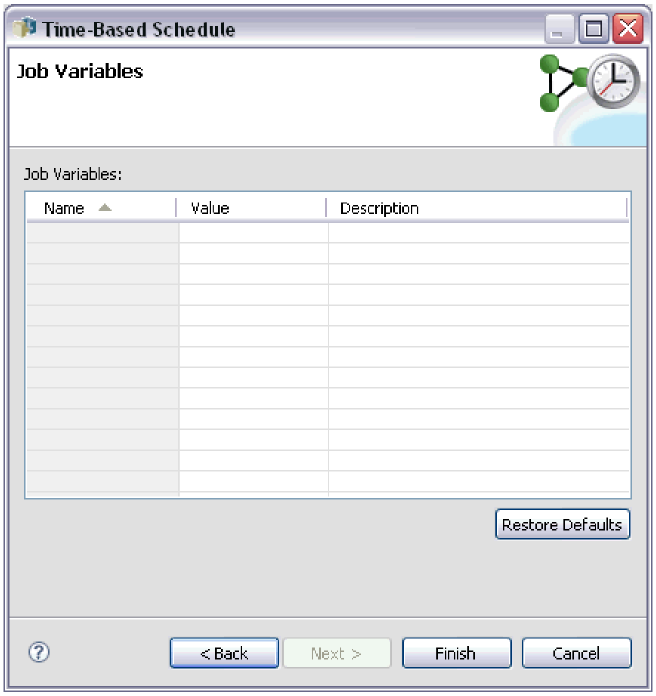
Note: Job processing may depend on several different external events. The initial event that triggers the start of the execution is specified in the message-based schedule. If processing requires subsequent events to take place, they must be specified by message-based job steps. For more information, see the topic Message-Based Job Steps in Chapter 18 on p. 333.

Click Next to specify values for variables used in the scheduled job.

Job Variables for Schedules

The Job Variables page for the Schedule wizard specifies values for the variables defined for the job associated with the schedule.

Figure 11-6
Job Variables for Schedules



The variables table contains the following columns:

- **Name.** The names of the existing job variables.
- **Value.** The value currently assigned to each variable. These are typically the default values defined for the variables.
- **Description.** Informative text about each variable.

When scheduling a job, you may want to use alternative values for one or more variables. To change a variable value:

1. Click the value to be changed.
2. Type the new value for the variable.
3. Press the Enter key.

The existing value updates to the new value. To return the variables to their default values, click Restore Defaults.

Click Finish to create the schedule for the job.

Mapping JMS Header Properties to Job Variables

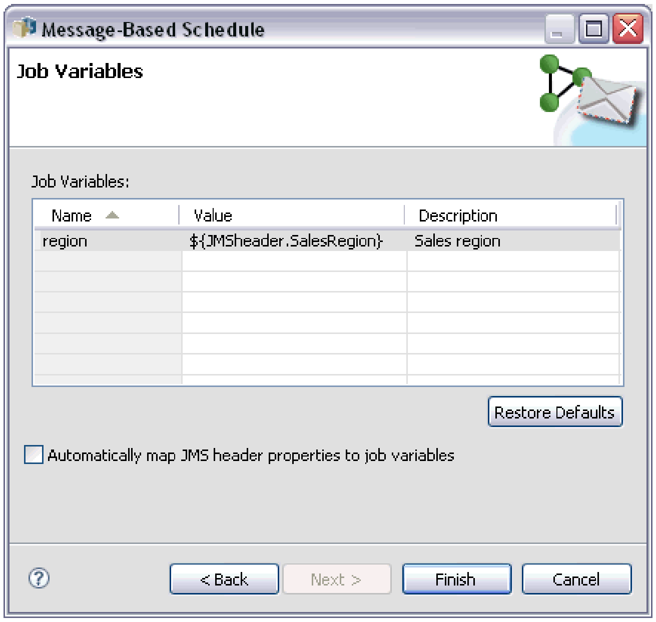
For message-based schedules, the JMS message may contain variables in the header that can be mapped to the values for job variables. This mapping may be manual or automatic, as defined by the Job Variables page of the Schedule wizard.

In manual mapping, the header properties from the message are assigned to specific job variables. Message header variables are referenced using the following syntax:

```
${JMSHeader.propertyName}
```

where *propertyName* corresponds to the name of the property in the message header. For example, suppose the header for the message associated with the schedule includes the property *SalesRegion* and the job being scheduled includes the job variable *region*. The value of *SalesRegion* is assigned to *region* by specifying *${JMSHeader.SalesRegion}* as the value for *region*.

Figure 11-7
Mapping Header Variables



Message-Based Schedule

Job Variables

Job Variables:

Name	Value	Description
region	${JMSheader.SalesRegion}	Sales region

☐ Automatically map JMS header properties to job variables

In automatic mapping, on the other hand, the values for header properties in the JMS message are automatically used for any job variables that have the same name as the header property. Values defined in the schedule for the matching job variables are replaced with values from the message. Job variables that do not match a header property use the values specified in the schedule. If any of those values reference a header property that is unavailable, the name of the value is used. To use an empty string instead of the value name, use **quiet reference notation** by inserting an exclamation point between the dollar sign and the property name:

```
$!{JMSHeader.propertyName}
```

Header properties in the message that do not match a job variable are unused by the schedule. To enable automatic mapping, select the Automatically map JMS header properties to job variables option.

Editing Schedules

A list of schedules defined in the system can be accessed using the Job Schedule view. For more information, see the topic Job Schedule View in Chapter 12 on p. 262.

To edit an existing job schedule, right-click on the schedule in the Job Schedule view and select Edit Schedule. Alternatively, you can click the *Edit Schedule* icon.

The Schedule wizard opens, showing the settings for the selected schedule. Modify the values as needed and click Finish to save the updated schedule.

Reactivating a Dormant Schedule

If a version label of a job is removed, or if that version of the job is deleted, the schedule associated with the labeled version becomes dormant. Thus, the dormant schedule cannot be used until it is reassociated with a valid labeled job version.

If a schedule becomes dormant, the following message appears:

The scheduled label, <label name>, for this job does not exist anymore. You must select a different label, or re-apply the originally scheduled label to the job.

<label name> represents the version label that was removed or the job version that was deleted.

The resolution for a dormant schedule depends upon how the schedule became dormant.

- **Deleted job version.** If the job version was deleted, a different job version must be selected for scheduling.
- **Deleted version label.** If the label for the job version was removed, the label needs to be reapplied to the job version.

Deleting Schedules

To delete an existing job schedule, right-click on the schedule in the Job Schedule view and select Delete Schedule(s). Alternatively, you can click the *Delete Schedule(s)* icon.

The job associated with the schedule remains in the system. However, the job will no longer execute in accordance with the deleted schedule.

Message-Based Processing Example

Message-based scheduling functionality of PASW Collaboration and Deployment Services can be used to trigger processing by repository events as well as by third-party applications. For example, a job can be configured to be rerun when the PASW Modeler stream used in one of the job steps is updated. The procedure involves the following steps:

▶ Using Deployment Manager, create a JMS message domain. For more information, see the topic Message Domains in Chapter 6 on p. 94.

- Set up a message-based schedule for the job using the message domain. Note that the JMS message selector must indicate the resource ID of the PASW Modeler stream as in the following example:

  ```
  ResourceID=<resource ID>
  ```

  The repository resource ID of the PASW Modeler stream can be found in the object properties. For more information, see the topic Working With Object Properties in Chapter 5 on p. 34.

- Set up a notification for the PASW Modeler stream based on the JMS subscriber you have defined. For more information, see the topic JMS Subscriber in Chapter 13 on p. 286.

- To test the message-based schedule, the stream must be opened in PASW Modeler, modified, and stored in repository. If everything has been set up correctly, the schedule will trigger the job.

Chapter

12

Monitoring Status

In Deployment Manager, the status of a job can be analyzed via a number of job summary views. Information is organized in a table and is designed to provide an at-a-glance summary for jobs in the repository. The following views are available:

- Job Schedule
- Job History
- Model Management
- Predictors

Although individual tables contain specialized information, all job summary views are accessed in a similar way.

In addition, the status of servers and server clusters can also be monitored. For more information, see the topic Server Status View on p. 275.

Accessing Status Views

In the following section, the <View Type> refers to the specialized view. Examples include:

- *Job Schedule*.
- *Job History* .
- *Model Management*.
- *Server Status*.

Job summary views can be accessed using any of the following methods:

Toolbar. This option launches a blank job summary view. To access the job schedule view from the toolbar:

▶ From the View menu, select:

Show View
  <View Type>

Content Explorer. This option launches a populated job summary view. To access the job summary view from the Content Explorer:

▶ Right-click on a job and select Show <View Type>.

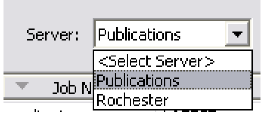
Selecting a Server in Status Views

By default, jobs on the current server appear in the job list.

To view jobs for other servers:

▶ From the Server drop-down list, select a server.

Figure 12-1
Selecting a server



Server: Publications

The job list is updated with jobs on the selected server.

Note: If you are not logged into the server that was selected, the *Login to repository* dialog box opens, prompting you to log into the server.

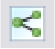
Opening a Job in the Job Editor

At times, when a job summary table is open (e.g., the job schedule or the job history), it may be useful to open the job in the Job Editor and view the job contents in conjunction with the job table.

To open a job in the Job Editor:

1. Select the job in the table.
2. Click the Open job in editor icon. The job opens in the canvas of the Job Editor.

Figure 12-2
Open job in editor icon



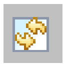
Refreshing Status Views

When a job is run, the corresponding job tables (e.g., job schedule and job history) are not refreshed automatically. You must refresh job tables manually.

To refresh the job table:

1. Click the Refresh icon. The job table displays the updated status.

Figure 12-3
Refresh job status icon



Reordering Items in Status Views

By default, items in the job summary tables (e.g., job schedule and job history) are organized in chronological order. Thus, the most recent item appears at the bottom of the table. The order of items in the job summary tables can by reorganized by column.

To reorder items:

1. Click in the header row of the column that you want to reorder.
2. Click on the triangle to the left of the row title. The table is reordered.

Deleting a Job from Status Views

To delete an item from a job summary table (e.g., a job schedule or a job history):

1. Select the item in the table. To select multiple items, hold down the Ctrl key and select additional rows.
2. Click the Delete icon. The Delete Confirmation dialog box opens.

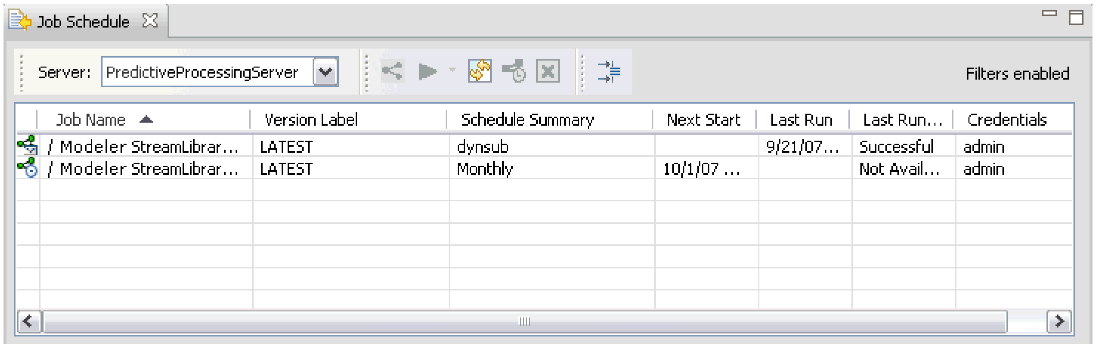
Figure 12-4
Delete icon

3. Click OK. The item is removed from the job schedule table.

Job Schedule View

A scheduled job is run automatically at a designated date and time or when a JMS message is received from an outside application.

Figure 12-5
Job schedule table



Server: PredictiveProcessingServer

Job Name	Version Label	Schedule Summary	Next Start	Last Run	Last Run...	Credentials
/ Modeler StreamLibrar...	LATEST	dynsub		9/21/07...	Successful	admin
/ Modeler StreamLibrar...	LATEST	Monthly	10/1/07 ...		Not Avail...	admin

Job schedules are described in the Job Schedule table, which contains the following summary information:

Job Name. The name of the job.

Version Label. The label that has been applied to the job. If no label has been applied to the job, this field is blank.

Schedule Summary. The frequency with which the job is scheduled to run. Valid values are *Once*, *Hourly*, *Daily*, *Weekly*, and *Monthly*.

Next Start. The next date and time at which the job is scheduled to run.

Last Run. The date and time when the job was last run.

Last Run Status. The status of the job when it was last run. Valid values include *SUCCESS* or *FAILED*. A cancelled job is recorded as a failure.

Credential. The credential under which the scheduled job will run.

The job schedule type (message-based or time-based) is specified by the icon in the column before the job name.

The first column indicates any problems with a schedule, such as a reference to a message domain that has been deleted, by displaying a warning icon. Right-click a problematic schedule and select Show Error Messages to view any errors associated with the schedule.

Job History View

Each time a job is run, the action is recorded in the job history, which provides status information about the job and its corresponding steps. Specifically, the *Name* column describes the overall job. When the + is expanded, information about the individual job steps within a job appears.

Information in the job history table cannot be modified because this table reflects information obtained from other components in the system. Job information is provided on a per server basis. For more information, see the topic Selecting a Server in Status Views on p. 261.

In addition, filters can be applied to the job history to reduce the number of jobs that appear in the list. For more information, see the topic Job History Filters on p. 273.

Working With the Job History Table

The job history table contains the following information:

Name. The name of the job or step.

Version. The version label that has been applied to the job. If no version label was explicitly applied in the job schedule, then the *LATEST* label is used by default.

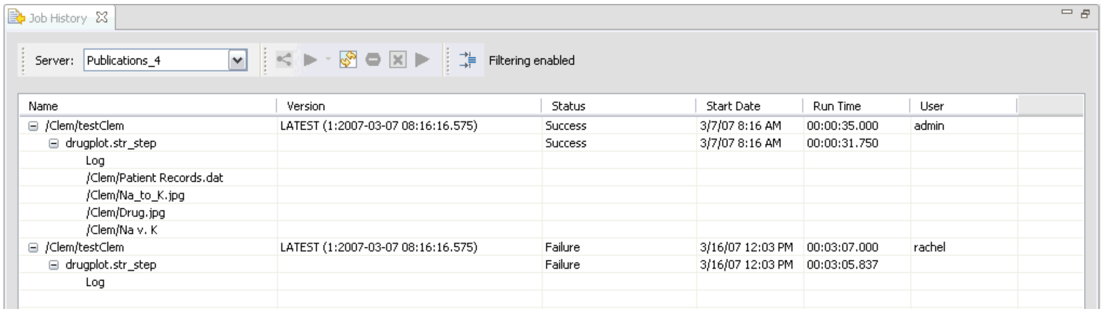
Status. The current status of the job or step. Valid values include *Success*, *Running* or *Failed*. A cancelled job is recorded as a failure. The status of the individual job steps that comprise each job, as well as any corresponding logs, appear underneath each job. To expand the job history list for a specific job, click the + next to the job.

Start date. The date and time at which the job or step was started.

Run time. The time taken to run the job or step. It is important to note that a value in this field does not necessarily imply success. The *Status* column describes whether or not the job ran successfully.

User. The user who last scheduled the job.

Figure 12-6
Job History table



Name	Version	Status	Start Date	Run Time	User
⊟ /Clem/testClem	LATEST (1:2007-03-07 08:16:16.575)	Success	3/7/07 8:16 AM	00:00:35.000	admin
⊟ drugplot.str_step		Success	3/7/07 8:16 AM	00:00:31.750	
Log					
/Clem/Patient Records.dat					
/Clem/Na_to_K.jpg					
/Clem/Drug.jpg					
/Clem/Na v. K					
⊟ /Clem/testClem	LATEST (1:2007-03-07 08:16:16.575)	Failure	3/16/07 12:03 PM	00:03:07.000	rachel
⊟ drugplot.str_step		Failure	3/16/07 12:03 PM	00:03:05.837	
Log					

Blank Cells in the Job History View

If no label has been applied to a job, the corresponding *Label* field is blank. For the *Status*, *Start Time*, and *Run Time* fields, a blank cell indicates that the job has not been run yet.

Canceling a Job

You can cancel a job while it is running. It is important to note that canceling a job is not the same as deleting a job. Canceling only stops the job from running. Canceling does not remove a job from the content repository.

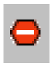
If you cancel a job that contains relationships, the relationships will not be followed. For example, suppose your job contains step A and step B, which are joined by a sequential connector. If you cancel the job while step A is running, step B will not be executed. This process applies to all relationship connectors, including the fail connector.

To cancel a job:

1. In the Job History table, select the job that you want to cancel. The job status of the selected job should be *Running*.

2. Click the Stop selected running job icon.

Figure 12-7
Cancel job icon



Viewing Job Results

If results are available, separate rows appear under the job steps in the table. Each row contains the path in which the results are stored. For certain applications, such as PASW Modeler, double-clicking on the results path launches the PASW Modeler client and displays the results.

The system can open any result file that is written back to the repository. If the file type has a corresponding editor in the Deployment Manager (e.g., HMTL, text, image files), the system opens the file within the Deployment Manager. If a file path is preceded by the machine name, the

Deployment Manager does not have a editor to display the file and will call upon the operating system to open the file.

Viewing Job Logs

The job history table includes logs for the overall job as well as for steps within the job. These logs are system-generated logs and cannot be modified. Note that some types of steps do not produce logs.

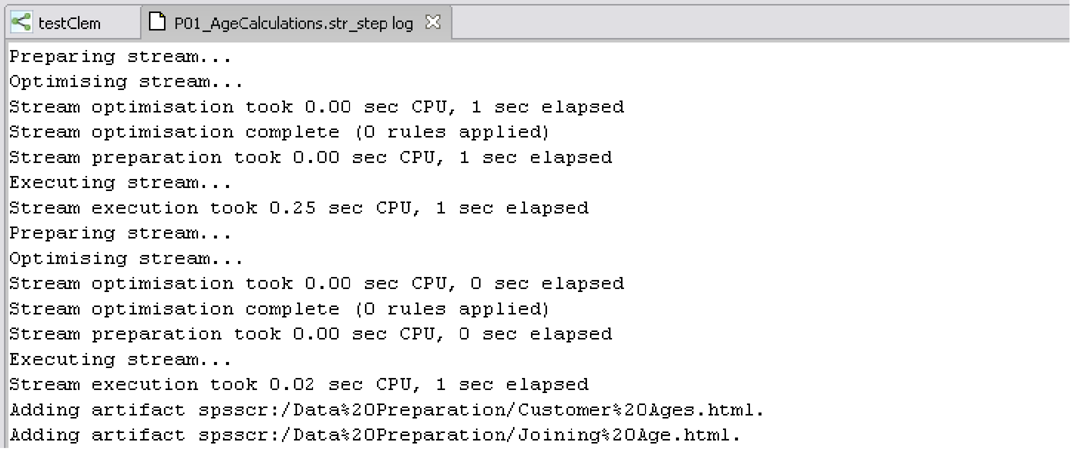
In the table, the overall log appears under a subheading corresponding to the name of the job. This log contains information relevant to the entire job, such as whether or not residual artifacts have been deleted. The values used for any job variables also appear in this log.

If a log is available for a job step, it too can be accessed from the job history table. Job step logs appear under a subheading corresponding to the job step name and pertain to the corresponding step only. If the job contains multiple steps that produce logs, the logs appear under each step in the table. These are not logs for the entire job, and they are not logs for the system.

To view a log, double-click Log in the Name column for the cell that corresponds to the log that you want to view. The log appears in a separate editor.

For example, a sample log for a PASW Modeler stream might look like this:

Figure 12-8
PASW Modeler log example



```
Preparing stream...
Optimising stream...
Stream optimisation took 0.00 sec CPU, 1 sec elapsed
Stream optimisation complete (0 rules applied)
Stream preparation took 0.00 sec CPU, 1 sec elapsed
Executing stream...
Stream execution took 0.25 sec CPU, 1 sec elapsed
Preparing stream...
Optimising stream...
Stream optimisation took 0.00 sec CPU, 0 sec elapsed
Stream optimisation complete (0 rules applied)
Stream preparation took 0.00 sec CPU, 0 sec elapsed
Executing stream...
Stream execution took 0.02 sec CPU, 1 sec elapsed
Adding artifact spsscr:/Data%20Preparation/Customer%20Ages.html.
Adding artifact spsscr:/Data%20Preparation/Joining%20Age.html.
```

Model Management Views

A summary of model results is provided in the model management views. Similar to the job history and job schedule views, the model management tables provide information about scenario analysis results. The following views are available:

- Model evaluation
- Champion challenger

The information provided in each view, which may vary by model type, is not modifiable. In addition, filters can be used to narrow the results of a model management view even further. For more information, see the topic Model Management Filters on p. 275.

To select a specific model management view:

1. From the View menu select:

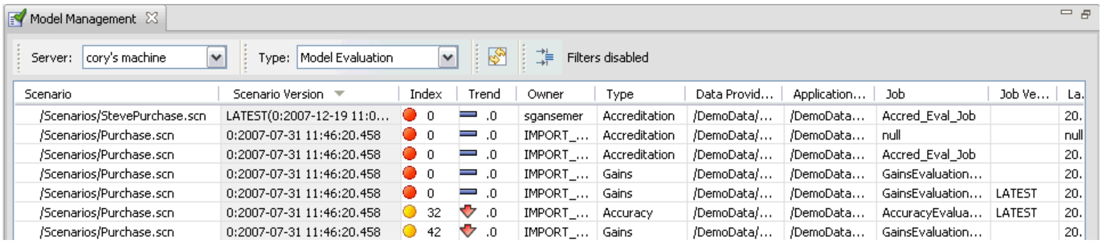
   Show View
     Model Management

   The Model Management tab appears.

2. From the Server drop-down list, select the server name.

3. From the Type drop-down list, select the view type. Options include model evaluation or champion challenger.

Figure 12-9
Model evaluation view



Server: cory's machine
Type: Model Evaluation

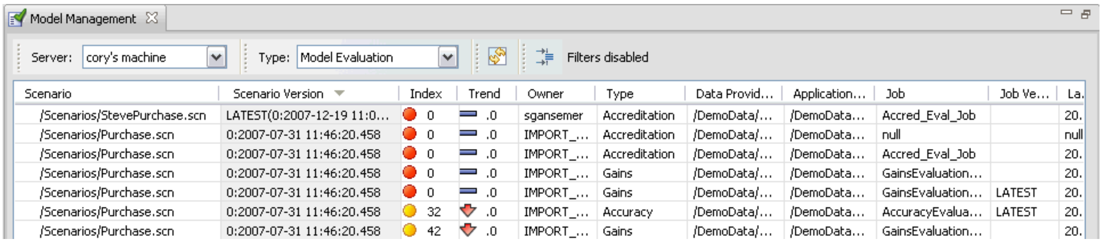
Scenario	Scenario Version	Index	Trend	Owner	Type	Data Provid...	Application...	Job	Job Ve...	La.
/Scenarios/StevePurchase.scn	LATEST(0:2007-12-19 11:0...	● 0	— .0	sgansemer	Accreditation	/DemoData/...	/DemoData...	Accred_Eval_Job		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Accreditation	/DemoData/...	/DemoData...	null		null
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Accreditation	/DemoData/...	/DemoData...	Accred_Eval_Job		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...	LATEST	20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 32	▼ .0	IMPORT_...	Accuracy	/DemoData/...	/DemoData...	AccuracyEvalua...	LATEST	20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 42	▼ .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...		20.

Note: The model management views present information about scenarios stored in the repository. To store scenarios in the repository, the PASW Collaboration and Deployment Services environment must include PASW Modeler adapters. For information on installing the adapters, see the PASW Modeler documentation.

Model Evaluation View

The model evaluation view describes results for model evaluation scenarios. This view is designed to provide an at-a-glance summary of the overall performance of the scenario. For example, this view describes whether the scenario is trending upward, downward, or has remained the same. Jobs may appear multiple times in the model evaluation view. One entry appears for each labeled version of a job that contains a model evaluation scenario.

Figure 12-10
Model evaluation view



Model Management
Filters disabled

Server: cory's machine
Type: Model Evaluation

Scenario	Scenario Version	Index	Trend	Owner	Type	Data Provid...	Application...	Job	Job Ve...	La.
/Scenarios/StevePurchase.scn	LATEST(0:2007-12-19 11:0...	● 0	— .0	sgansemer	Accreditation	/DemoData/...	/DemoData...	Accred_Eval_Job		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Accreditation	/DemoData/...	/DemoData...	null		null
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Accreditation	/DemoData/...	/DemoData...	Accred_Eval_Job		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...		20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 0	— .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...	LATEST	20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 32	▼ .0	IMPORT_...	Accuracy	/DemoData/...	/DemoData...	AccuracyEvalua...	LATEST	20.
/Scenarios/Purchase.scn	0:2007-07-31 11:46:20.458	● 42	▼ .0	IMPORT_...	Gains	/DemoData/...	/DemoData...	GainsEvaluation...		20.

The Model Evaluation view contains the following information.

Scenario. The name and path of the scenario used to perform the analysis.

Scenario Version. The version of the scenario used to generate the results.

Index. A percentage value. The colored circle next to the index value corresponds to the thresholds for performance (e.g., good, better, bad) that were specified in the General tab for the scenario job step. For example, a red circle indicates that the index value falls into the lowest-performing range.

Trend. The percentage change in the model. An arrow indicates whether the trend is up or down. If the trend is 0.00, a horizontal bar appears, which represents no change. The first time the scenario is run, the trend field is blank.

Author. The author of the scenario.

Type. The type of analysis. Examples include *gains* and *accuracy*.

Data provider definition. The data provider definition (DPD) associated with the scenario.

Application view. The application view associated with the scenario.

Job. The job that contains the scenario.

Job Version. The version of the job used to analyze the scenario.

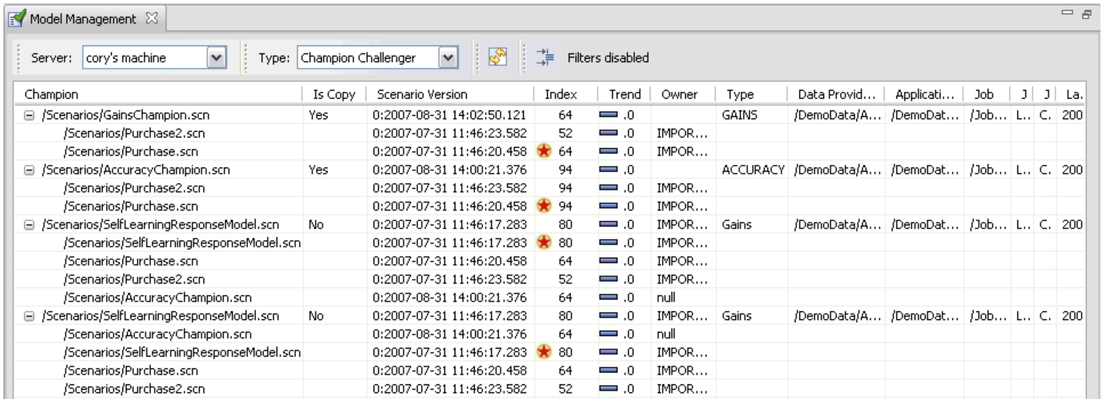
Last successful run. The most recent date and time at which the job was run successfully. All of the model evaluation views only display information for the most recent job that was run successfully. A full job history, including any previously run jobs or failed jobs is available in the job history view. For more information, see the topic Job History View on p. 263.

Champion Challenger View

The Champion Challenger view describes scenarios that were compared to one another within a Champion Challenger job step. The scenario that was deemed to be the most effective scenario is designated as the champion. The champion scenario appears as the uppermost row in the Champion Challenger view. To see all of the scenarios analyzed in the Champion Challenger job step, simply expand the tree.

It is important to note that one of the scenarios in the list was designated the champion scenario. The champion scenario is then renamed in accordance with the parameters specified on the Champion tab of the champion-challenger job step. Thus, the renamed scenario appears in the uppermost row.

Figure 12-11
Champion-challenger view



Model Management

Server: cory's machine   Type: Champion Challenger   Filters disabled

Champion	Is Copy	Scenario Version	Index	Trend	Owner	Type	Data Provid…	Applicati…	Job	J	J	La.
⊟ /Scenarios/GainsChampion.scn	Yes	0:2007-08-31 14:02:50.121	64	▬ .0		GAINS	/DemoData/A…	/DemoDat…	/Job…	L..	C.	200
/Scenarios/Purchase2.scn		0:2007-07-31 11:46:23.582	52	▬ .0	IMPOR…							
/Scenarios/Purchase.scn		0:2007-07-31 11:46:20.458	★ 64	▬ .0	IMPOR…							
⊟ /Scenarios/AccuracyChampion.scn	Yes	0:2007-08-31 14:00:21.376	94	▬ .0		ACCURACY	/DemoData/A…	/DemoDat…	/Job…	L..	C.	200
/Scenarios/Purchase2.scn		0:2007-07-31 11:46:23.582	94	▬ .0	IMPOR…							
/Scenarios/Purchase.scn		0:2007-07-31 11:46:20.458	★ 94	▬ .0	IMPOR…							
⊟ /Scenarios/SelfLearningResponseModel.scn	No	0:2007-07-31 11:46:17.283	80	▬ .0	IMPOR…	Gains	/DemoData/A…	/DemoDat…	/Job…	L..	C.	200
/Scenarios/SelfLearningResponseModel.scn		0:2007-07-31 11:46:17.283	★ 80	▬ .0	IMPOR…							
/Scenarios/Purchase.scn		0:2007-07-31 11:46:20.458	64	▬ .0	IMPOR…							
/Scenarios/Purchase2.scn		0:2007-07-31 11:46:23.582	52	▬ .0	IMPOR…							
/Scenarios/AccuracyChampion.scn		0:2007-08-31 14:00:21.376	64	▬ .0	null							
⊟ /Scenarios/SelfLearningResponseModel.scn	No	0:2007-07-31 11:46:17.283	80	▬ .0	IMPOR…	Gains	/DemoData/A…	/DemoDat…	/Job…	L..	C.	200
/Scenarios/AccuracyChampion.scn		0:2007-08-31 14:00:21.376	64	▬ .0	null							
/Scenarios/SelfLearningResponseModel.scn		0:2007-07-31 11:46:17.283	★ 80	▬ .0	IMPOR…							
/Scenarios/Purchase.scn		0:2007-07-31 11:46:20.458	64	▬ .0	IMPOR…							
/Scenarios/Purchase2.scn		0:2007-07-31 11:46:23.582	52	▬ .0	IMPOR…							

The Champion Challenger table contains the following information.

Champion. Champion-challenger scenarios are organized around the champion. Thus, the first line introducing a set of champion-challenger scenarios contains scenario name and path information of the champion scenario. The scenarios underneath the champion scenarios are challenger scenarios.

Is copy. This column indicates whether or not the champion scenario listed in the first row of the current set of champion-challenger scenarios is a copy of the winning scenario. (The winning scenario is designated by a star in the Index column.) If *Yes* appears in this field, then a copy of the champion scenario is displayed in the first row. By default, Deployment Manager makes a copy of the winning champion scenario. This setting can be disabled when creating or editing a champion-challenger job step. If the option to copy the champion scenario was deselected, then the winning champion scenario appears in the first row under the scenario name.

Scenario Version. The version of the scenario used to generate the results.

Index. A percentage value. If a star appears next in the Index column, this scenario was the winning scenario in the champion-challenger analysis.

Trend. The percentage change in the model. An arrow indicates whether the trend is up or down. If the trend is 0.00, a horizontal bar appears, which represents no change. The first time the scenario is run, the trend field is blank.

Author. The author of the scenario.

Type. The type of analysis. Examples include *gains* and *accuracy*.

Data provider definition. The data provider definition (DPD) associated with the scenario.

Application view. The application view associated with the scenario.

Job. The job that contains the scenario.

Job Version. The version of the job used to analyze the scenario.

Job step. The type of job step that contains the scenario.

Last successful run. The most recent date and time at which the job was run successfully. All of the model evaluation views only display information for the most recent job that was run successfully. A full job history, including any previously run jobs or failed jobs is available in the job history view. For more information, see the topic Job History View on p. 263.

Predictors View

Both champion challenger and model evaluation scenarios use predictors to generate scenario results. A predictor is a variable that is used as an input for a scenario. Typically, a scenario contains multiple predictors. Predictors are evaluated and then ranked based on their relevance to the results.

To view the Predictors table:

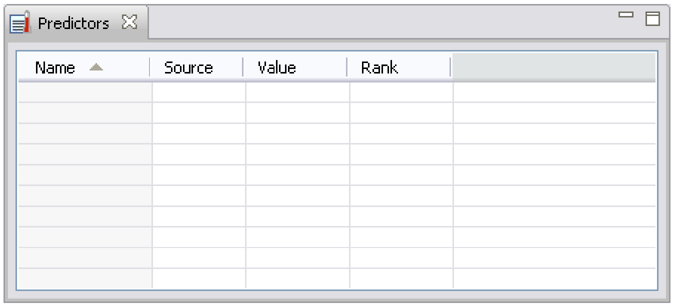
1. In the Content Explorer, select an object.

2. From the View menu select:

   Show View
     Predictors

   The Predictors table appears.

Figure 12-12
Predictor effectiveness view



Name	Source	Value	Rank

The predictor effectiveness table contains the following information. Some parameters apply at the predictor level. Other columns in the table apply to the entire scenario.

Predictor-level Information

Name. The predictor name. Predictors are organized by scenario. Thus, the first line introducing a set of predictors contains scenario name and path information.

Source. The data source used to obtain predictors—for example, *Enterprise View*.

Value. The value of the predictor.

Rank. The rank of the predictor. Predictors are listed in order of descending rank. The rank of a predictor corresponds to its importance within the model. For example, if the a predictor for household income is listed first on the list, then for this scenario, the level of household income was most highly correlated with a positive response.

Scenario-level Information

Scenario Version. The version of the scenario used to generate the results.

Author. The author of the scenario.

Data provider definition. The data provider definition (DPD) associated with the scenario.

Application view. The application view associated with the scenario.

Job. The job that contains the scenario.

Application view label. The label associated with the application view used by the scenario.

Last successful run. The most recent date and time at which the job was run successfully. All of the model evaluation views only display information for the most recent job that was run successfully. A full job history, including any previously run jobs or failed jobs is available in the job history view. For more information, see the topic Job History View on p. 263.

Note: The Predictors view presents information about scenarios stored in the repository. To store scenarios in the repository, the PASW Collaboration and Deployment Services environment must include PASW Modeler adapters. For information on installing the adapters, see the PASW Modeler documentation.

Filters

Any of the available views can be filtered to reduce the number of results that appear in the view table. Although some filtering options are common to all views, specific filtering parameters may vary by the view type. Multiple filters can be used simultaneously. For example, the job filter is often used in conjunction with other filters.

By default, filtering is disabled. Once filtering is enabled, the ability to filter the job history persists as long as the same Deployment Manager is used. In addition, filter settings persist across server connections. For example, if filtering is enabled and the server selection is changed from server A to server B, the filter settings established for server A persist for server B.

Filters Common to All Status Views

The following filters are common to all views:

Job. The jobs that appear in the view table can be limited to the following:

- **Selected job in job editor.** Filters the table, so that only information for the currently selected job appears.
- **User-defined job.** Used to search for a job schedule by name. Selecting Browse allows a search of the entire content repository.

Version Label. Narrows the list to objects containing the specified version label. Typically, this option is used in conjunction with the job filter. For example, if the version label *LATEST* were chosen without a corresponding job filter, every job in repository would appear in the job history because the current version of any job is implicitly labeled *LATEST*. For the model management filters, the results can be further narrowed refined by job version label or scenario version label.

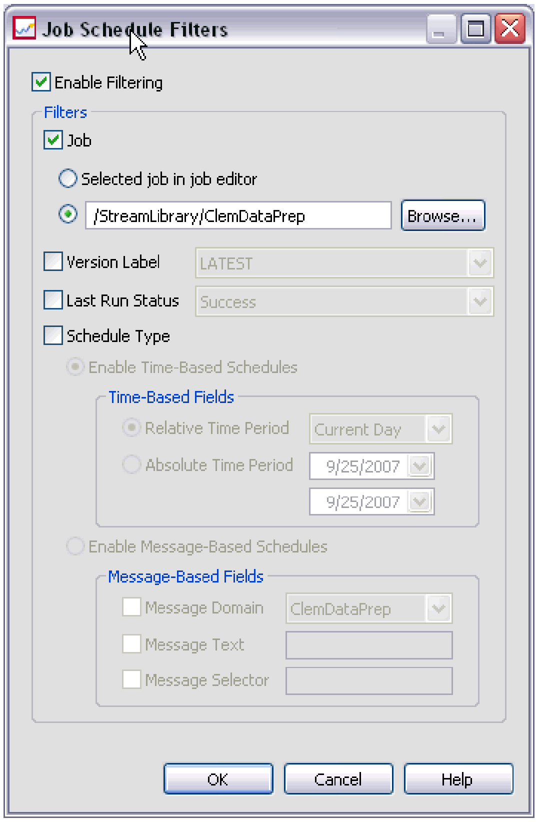
Job Schedule Filters

To access job schedule filters:

▶ In the job schedule view, click the Filters button.

The Job Schedule Filters dialog box opens.

Figure 12-13
Filtering the job schedule



Job Schedule Filters

☑ Enable Filtering

Filters

☑ Job

○ Selected job in job editor

● /StreamLibrary/ClemDataPrep

□ Version Label: LATEST

□ Last Run Status: Success

□ Schedule Type

● Enable Time-Based Schedules

Time-Based Fields

● Relative Time Period: Current Day

○ Absolute Time Period: 9/25/2007 – 9/25/2007

○ Enable Message-Based Schedules

Message-Based Fields

□ Message Domain: ClemDataPrep

□ Message Text

□ Message Selector

Enable Filtering. If filtering has not been previously enabled, select the Enable Filtering check box.

The job schedule can be filtered based on any of the following criteria:

Job. The jobs that appear in the job schedule can be limited to the following:

- **Selected job in job editor.** Filters the job schedule table, so that only the history for the currently selected job appears.
- **User-defined job.** Used to search for a job schedule by name. Selecting Browse allows a search of the entire content repository.

Version Label. Narrows the list to jobs containing the specified version label. Typically, this option is used in conjunction with the job filter. For example, if the version label *LATEST* were chosen without a corresponding job filter, every job in the repository would appear in the job schedule because the current version of any job is implicitly labeled *LATEST*.

Last Run Status. Narrows the list to jobs containing the selected status. The following status options can be specified:

- Success
- Failure
- Canceled

Schedule type Enables filtering by schedule type (message-based or time-based).

- Time-Based Fields. Selecting the option allows you to specify one the following criteria for filtering time-based schedules:

Field	Description
Relative Time Period	A finite time period relative to the current date. The following options are available: Current day Current week Current month Current quarter Current year Most recent 30 daysMost recent 3 months Most recent 6 monthsMost recent 9 months
Absolute Time Period	A specific date-based range of time. A valid range must be provided—for example, the end date must be equal to or greater than the start date.

- Message-Based Fields. Selecting the option allows you to specify the following criteria for filtering message-based schedules:

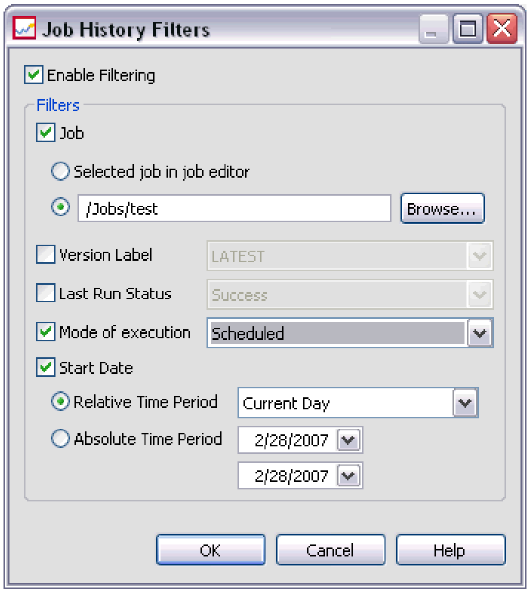
Field	Description
Message Domain	Filters schedules based on the JMS topic of the corresponding messages.
Message Text	Filters schedules based on the text of the corresponding JMS text messages.
Message Selector	Filters schedules based on the contents of the message headers, for example NewsType='Sports' or NewsType='Business'

Job History Filters

To access job history filters:

▶ In the job history view, click the Filters button. The Job History Filters dialog appears.

Figure 12-14
Filtering the job history



Job History Filters

☑ Enable Filtering

Filters

☑ Job

○ Selected job in job editor

● /Jobs/test Browse...

☐ Version Label LATEST

☐ Last Run Status Success

☑ Mode of execution Scheduled

☑ Start Date

● Relative Time Period Current Day

○ Absolute Time Period 2/28/2007 2/28/2007

Enable filtering. If filtering has not been previously enabled, select the Enable Filtering check box. The job history can be filtered based on any of the following criteria:

Job. The jobs that appear in the job history can be limited to the following:

- **The selected job in job editor.** Filters the job history table, so that only the history for the currently selected job appears.
- **A user-defined job.** Used to search for a job history by name. Selecting Browse allows a search of the entire content repository.

Version Label. Narrows the list to jobs containing the specified version label. Typically, this option is used in conjunction with the job filter. For example, if the version label *LATEST* were chosen without a corresponding job filter, every job in repository would appear in the job history because the current version of any job is implicitly labeled *LATEST*.

Last Run Status. Narrows the list to jobs containing the selected status. The following status options can be specified:

- Success
- Failure
- Canceled

Mode of execution. Narrows the list by mode of execution. Valid values include:

- Scheduled—Jobs that were previously scheduled and ran automatically at the specified date and time.
- Manual—Jobs that were executed manually using Run Job Now.
- Submitted—Jobs that were submitted for execution by an external source.

Start Date. Narrows the job history list based on the date the job was started. A relative or absolute time period can be specified.

- **Relative time period.** A finite time period relative to the current date. The following options are available:
- Current day
- Current week
- Current month
- Current quarter
- Current year
- Most recent 30 days
- Most recent 3 months
- Most recent 6 months
- Most recent 9 months
- **Absolute time period.** A specific date-based range of time. A valid range must be provided—for example, the end date must be equal or greater than the start date.

Model Management Filters

The results that appear in the model management view can be filtered based upon the following criteria.

Trend. The trends that appear in model management views can be limited to the following:

- **Direction.** The directional value of the trend—either up or down.
- **Values.** The range of the trend increase (or decrease). A minimum and maximum percentage must be specified.

Index. A percentage value. A minimum and maximum percentage must be specified.

Evaluation type. Valid values include *Gains*, *Accuracy*, and *Accreditation*.

Data provider definition. The data provider definition associated with the scenario.

Application view. The application view associated with the scenario.

Scenario Execution Date. A range of execution dates for scenarios. Specify the start and end dates for the range.

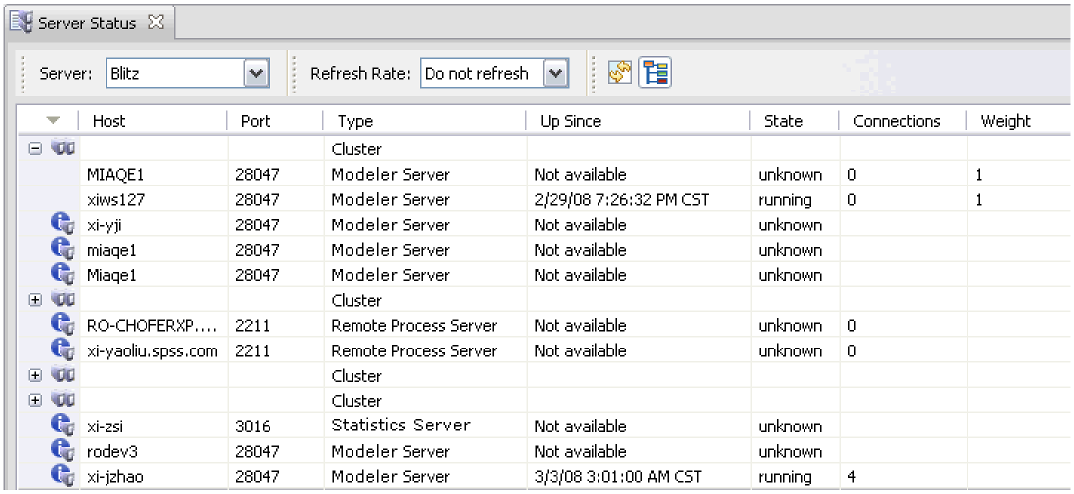
Server Status View

The status of servers and server clusters is available in the Server Status view. This view provides metadata about servers and server clusters for which you have permissions. Information cannot be modified in the server status table. Any modifications to server or server cluster configurations must be done in the *Resource Definitions* folder. For more information, see the topic Resource Definitions in Chapter 6 on p. 82.

To view the status for a server:

▶ From the **Server** drop-down list, select a server. A list of servers appears.

Figure 12-15
Server status view



Server: Blitz   Refresh Rate: Do not refresh

	Host	Port	Type	Up Since	State	Connections	Weight
⊟			Cluster				
	MIAQE1	28047	Modeler Server	Not available	unknown	0	1
	xiws127	28047	Modeler Server	2/29/08 7:26:32 PM CST	running	0	1
	xi-yji	28047	Modeler Server	Not available	unknown		
	miaqe1	28047	Modeler Server	Not available	unknown		
	Miaqe1	28047	Modeler Server	Not available	unknown		
⊞			Cluster				
	RO-CHOFERXP....	2211	Remote Process Server	Not available	unknown	0	
	xi-yaoliu.spss.com	2211	Remote Process Server	Not available	unknown	0	
⊞			Cluster				
⊞			Cluster				
	xi-zsi	3016	Statistics Server	Not available	unknown		
	rodev3	28047	Modeler Server	Not available	unknown		
	xi-jzhao	28047	Modeler Server	3/3/08 3:01:00 AM CST	running	4	

The server status table contains the following information:

Host. The host name for the server.

Port. The port number for the server.

Type. The type of server; examples include *PASW Statistics server*, *PASW Modeler server*, *Remote process server*, and *Cluster*.

Up Since. The last date and time at which the server was last started.

State. The status of the server—for example, *running*.

Connections. The number of connections into the server.

Weight. The weight of the server within a server cluster.

Viewing Servers by Cluster

By default, the server status table contains a list of servers. To view servers organized by cluster, click the show clusters icon.

Figure 12-16

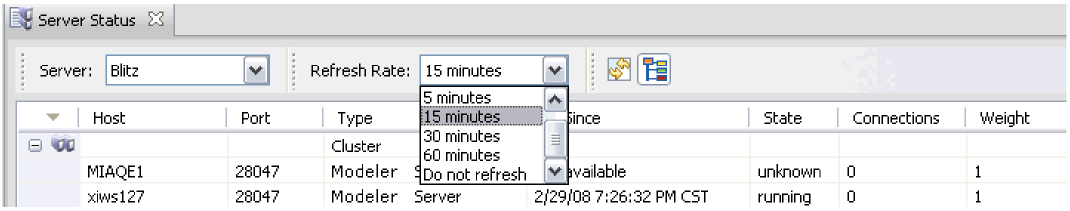
For server clusters, the host cell is blank. To view the servers within a cluster, click the + sign to expand the list. The host name for each server within the cluster appears in the table.

Setting the Refresh Rate

By default, the server status view is not automatically refreshed. To modify the refresh rate:

▶ From the Refresh Rate drop-down list, select the refresh rate increment—for example, *15 minutes*.

Figure 12-17
Server status refresh rate



Server: Blitz

Refresh Rate: 15 minutes

	Host	Port	Type		ince	State	Connections	Weight
⊟			Cluster					
	MIAQE1	28047	Modeler	S	vailable	unknown	0	1
	xiws127	28047	Modeler	Server	2/29/08 7:26:32 PM CST	running	0	1

Chapter

13

Notifications and Subscriptions

PASW Collaboration and Deployment Services provides the mechanisms of **notifications** and **subscriptions** for keeping the users informed about changes to repository objects and job processing results. Notifications and subscriptions generate e-mail messages or JMS (Java Messaging Service) messages when corresponding events occur. Notifications can be defined for multiple recipients while subscriptions to repository objects are defined by the system users only for themselves.

Notifications and subscriptions are retained when repository content is exported and imported.

Notifications

Notifications can be defined for changes to the repository content and processing events. File notifications are triggered by the creation of new file and job versions. Folder notifications are triggered by folder content changes, such as addition of files, creation of new file and job versions, and creation of subfolders. For more information, see the topic Content Notifications on p. 280. Notifications can also be defined for job processing events such as job and job step success and failure. See Job Success and Failure Notifications on p. 278, Job Step Notifications on p. 279 and Notifications Based On Model Evaluation Return Codes on p. 281 for more information. A user must be assigned the corresponding action through roles to define notifications.

Already existing notifications can be modified or removed for a single object or for multiple repository objects.

Job Success and Failure Notifications

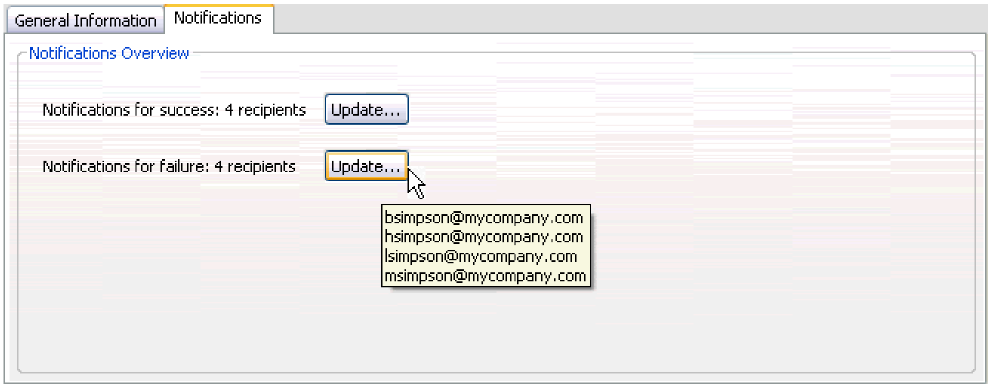
repository provides the capability to send e-mail notification messages for a job success or failure.

To display job notifications:

1. Open the job.

2. Click the Notifications tab. The Notifications for Success and Notifications for Failure fields display the number of recipients defined for each type of notification.

Figure 13-1
Job Notifications tab



Notifications for success: 4 recipients

Notifications for failure: 4 recipients

bsimpson@mycompany.com
hsimpson@mycompany.com
lsimpson@mycompany.com
msimpson@mycompany.com

To display the list of notification recipients, place the mouse pointer over the Update button.

To specify or update job notifications:

1. On the Notifications tab, click Update next to the notification type. The Notifications dialog box appears.

2. Specify the notification sender and recipients, and customize the message as necessary. For more information, see the topicNotification Settings on p. 281.

3. Save the job.

   Note: Notifications can also be defined for jobs as content repository objects. Such notifications will be triggered by the creation of a new version of a job. For more information, see the topic Content Notifications on p. 280. For example, defining job success or failure notifications for a job or a job step creates a new version of the job and subsequently triggers notification for the change.

Job Step Notifications

repository provides the capability to set up notifications for the success and failure of individual job steps, as well as jobs. A special case of job step notifications are notifications of success and failure of individual iterations; such notifications can be defined only for iterative job steps.

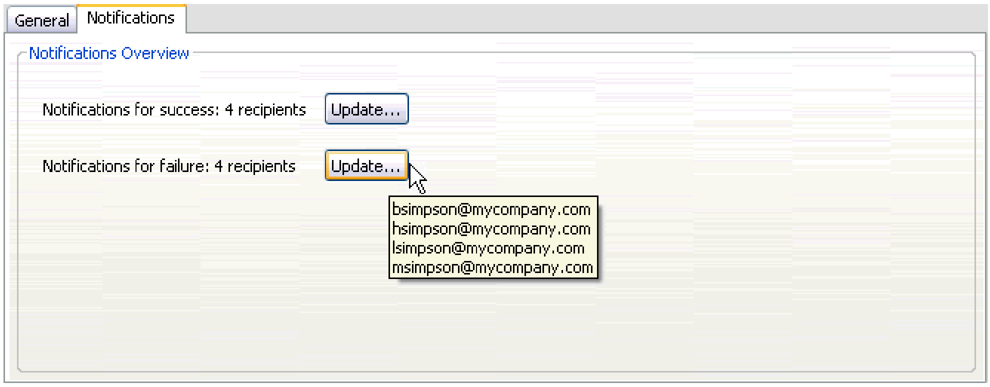
To display job step notifications:

1. Open the job.

2. Select the job step.

3. Click the Notifications tab. The Notifications for Success and Notifications for Failure fields display the number of recipients defined for each type of notification.

Figure 13-2
Job step Notifications tab for Windows command job step



Notifications for success: 4 recipients

Notifications for failure: 4 recipients

bsimpson@mycompany.com
hsimpson@mycompany.com
lsimpson@mycompany.com
msimpson@mycompany.com

To display the list of notification recipients, place the mouse pointer over the Update button.

To specify or update job step notifications:

1. On the Notifications tab, click Update next to the notification type. The Notifications dialog box appears.

2. Specify the notification sender and recipients, and customize the message as necessary. For more information, see the topicNotification Settings on p. 281.

3. Save the job.

Content Notifications

Content notifications keep users informed about changes to repository objects, such as the creation of new versions of files and jobs, and folder structures, such as the creation of new subfolders. Content notifications can be defined for a single object or multiple objects selected simultaneously.

To define or modify file or job notifications:

1. Right-click the file or job name and choose Notifications. To select multiple objects, hold down Shift or Ctrl key. The Notifications dialog box appears.

2. In the Notifications dialog box, specify the sender and recipients, customize the message as necessary, and, for folder notifications, indicate folder options. For more information, see the topicNotification Settings on p. 281.

To define or modify folder notifications:

1. Right-click the folder name and choose Notifications, then click Folder Events to define notifications for folder structure events or Folder Content Events to define notifications for changes to the objects in the folder.

2. In the Notifications dialog box, specify the sender, recipients, customize the message as necessary and, for folder notifications, indicate folder options. For more information, see the topicNotification Settings on p. 281.

To remove content notifications:

1. Right-click the file, job, or folder name and choose Notifications. The Notifications dialog box appears. Content notifications can also be removed for multiple objects selected simultaneously. To select multiple object, hold down Shift or Ctrl key.

2. In the Notifications dialog box, click Remove Notification. For more information, see the topicNotification Settings on p. 281.

   Notes

- Notifications option is not available if objects of different kind are selected simultaneously (for example, files and folders, folders and jobs).

- PASW Collaboration and Deployment Services jobs cannot be opened outside Deployment Manager. If content notifications are defined for jobs, it is strongly recommended to modify the notification settings to remove the hyperlink to the job because it cannot be opened in a Web browser. For more information, see the topic Customizing the Notification Message on p. 288.

Notifications Based On Model Evaluation Return Codes

Notifications can be set up to monitor specific model evaluation outcomes. The following are examples of when notifications might be based on model evaluation return codes:

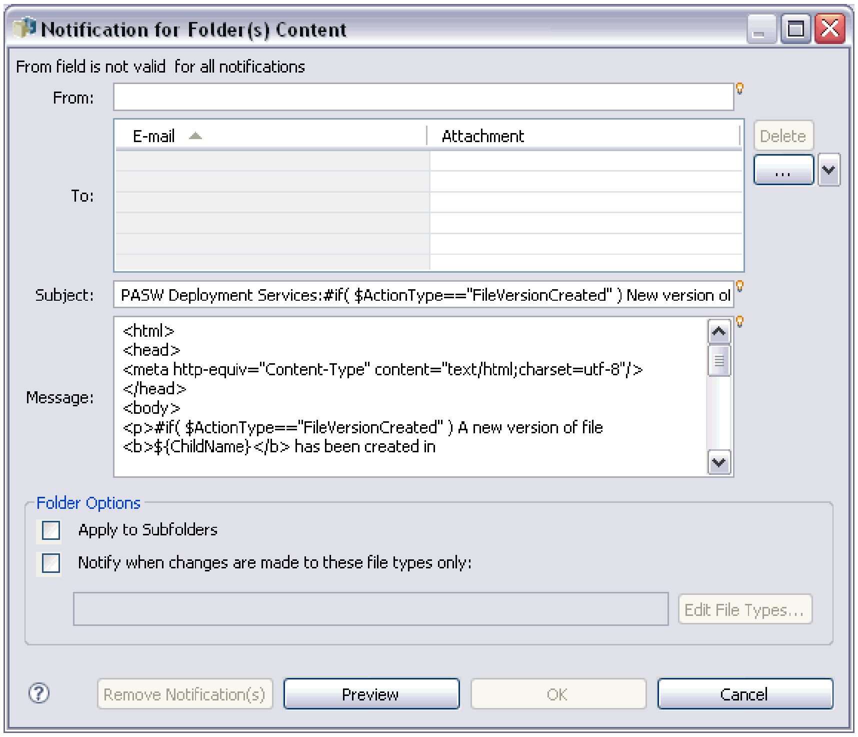
Notifications can be set up based on a specific outcome of the model evaluation. For example, if the result returned by a model evaluation analysis is less than 0.85, notifications could be sent to a predefined list of recipients.

Notifications can be set up based on the traffic light level of a model evaluation job. For example, if the result is red, a notification could be sent. In addition, multiple notification rules could be set up with different e-mail recipients for each traffic light level.

Notification Settings

Notification settings, including the notification sender, recipients, message customizations, and folder options, are specified in the Notifications dialog box.

Figure 13-3
Notifications dialog box



From field is not valid for all notifications

From:

To:

Subject: PASW Deployment Services:#if($ActionType=="FileVersionCreated") New version of

Message:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
</head>
<body>
<p>#if( $ActionType=="FileVersionCreated" ) A new version of file
<b>${ChildName}</b> has been created in
```

□ Apply to Subfolders

□ Notify when changes are made to these file types only:

From. The address of the sender notification message. The field is prepopulated with the default e-mail address in repository configuration options.

To. The list of notification recipients. The addresses can be entered, specified as an iterative variable for iterative job step notifications, specified as a dynamic list based on a query, or selected from a directory listing of a supported e-mail application such as Microsoft Outlook or Lotus Notes. To edit an address, click the ellipsis button next to it in the list. The list also specifies whether the modified object is to be attached to the message for each individual recipient. For more information, see the topic Notification Message Attachments on p. 287. Recipients can also be specified as variables. To remove a recipient, click Delete.

Subject. The subject of the notification message. By default, the field is populated by the default template associated with the notification event. Modify the message subject, if necessary. For more information, see the topic Customizing the Notification Message on p. 288.

Message. The body text of the notification message. The field is populated by the default template associated with the notification event. Modify the message body text, if necessary. For more information, see the topic Customizing the Notification Message on p. 288.

Remove Notification. Click to remove notification. When multiple objects are selected, clicking this button will remove all notifications for the objects.

Apply to subfolders. For folder notifications, applies the subscription to subfolders.

Notify when changes are made to these file types only. For folder notifications, limits notification to selected file types. Click the ellipsis button to open the File Types dialog box. For more information, see the topic Notification File Types on p. 289.

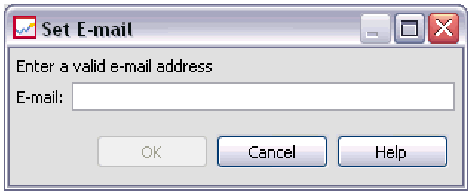
Preview. Click to preview the notification message. For more information, see the topic Notification Message Preview on p. 288.

Note: When notifications are modified for multiple objects, the list of recipients also displays a number next to the e-mail address indicating how many of the selected objects have the recipient in their notification settings. For multiple objects with different notification settings, the fields with varying values are blank and marked with alert signs. When notifications are modified for multiple folders with varying folder option settings, the options are marked with green squares.

Entering Notification Recipient E-mail Addresses

To enter a recipient e-mail address, on the Notification screen, click the ellipsis button next to the To field or click the down arrow next and then choose E-mail Subscriber. The Set E-mail dialog box appears.

Figure 13-4
Set E-mail dialog box



E-mail Address. The e-mail address of the notification recipient. The address string must contain a @ character followed by a period.

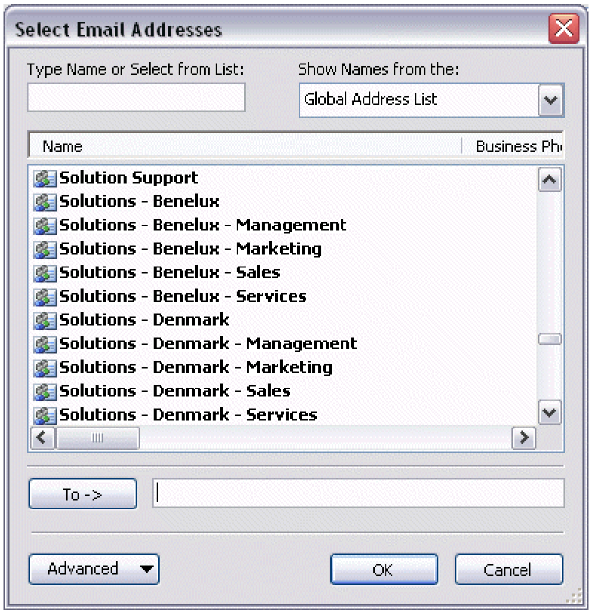
1. Enter the e-mail address. Only one address can be entered at a time.

2. Click OK. The address will appear in the To list of the Notifications dialog box.

Selecting Notification Recipients from Microsoft Outlook

To select a notification address from Microsoft Outlook directory, in the Notifications dialog box, click the down arrow next to the To field and then choose Outlook Subscriber. The Select E-mail Addresses dialog box appears.

Figure 13-5
Select E-mail Addresses dialog box



Select Email Addresses

Type Name or Select from List:

Show Names from the: Global Address List

Name	Business Ph
Solution Support	
Solutions - Benelux	
Solutions - Benelux - Management	
Solutions - Benelux - Marketing	
Solutions - Benelux - Sales	
Solutions - Benelux - Services	
Solutions - Denmark	
Solutions - Denmark - Management	
Solutions - Denmark - Marketing	
Solutions - Denmark - Sales	
Solutions - Denmark - Services	

Type Name or Select from List. Filters the list of available users according to the string entered. Use this field to refine the list of available users by user name.

Show Names from the. The list of organizational units defined in the directory. Use this field to refine the list of available users by organizational unit.

To—>. Displays selected users.

Advanced. Advanced options include Find and Properties. Click Find for an Outlook advanced directory search. Click Properties to display detailed information for the selected Outlook entry.

1. To select a recipient, double-click a directory entry in the list. The address appears in the To—> field. To remove a selected recipient, select the entry in the To—> field and press Delete key.

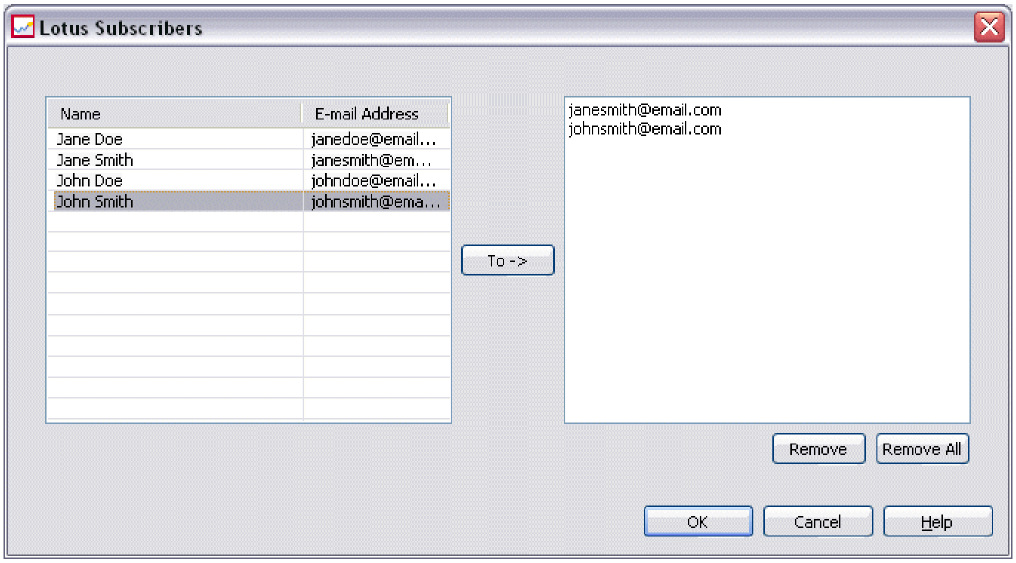
2. After all recipients have been selected, click OK. The e-mail addresses of selected recipients appear in the To list of the Notification for Job Success dialog box.

Note: If the Outlook Subscriber option is not available in the drop-down, verify that *CDO.DLL* is installed on your system (the default directory is *C:\Program Files\Common Files\System\MSMAPI\1033*). If the file is not present, the Collaboration Data Objects component of Microsoft Outlook must be installed from the Microsoft Office installation CD. The component can also be obtained from the Microsoft download site (*http://www.microsoft.com/downloads*).

Selecting Notification Recipients from Lotus Notes

To add a notification e-mail address from Lotus, in the Notifications dialog box, click the down arrow next to the To field and then choose Lotus Subscriber. You will be prompted for the Lotus Domino Server credentials, and then the Lotus Subscribers dialog box appears.

Figure 13-6
Lotus Subscribers dialog box



Name	E-mail Address
Jane Doe	janedoe@email...
Jane Smith	janesmith@em...
John Doe	johndoe@email...
John Smith	johnsmith@ema...

janesmith@email.com
johnsmith@email.com

1. To select a recipient, highlight a directory entry in the list and click To ->. The address appears in the To-> field. To remove a recipient, select the entry in the To-> field and click Remove. Click Remove All to clear all recipients.

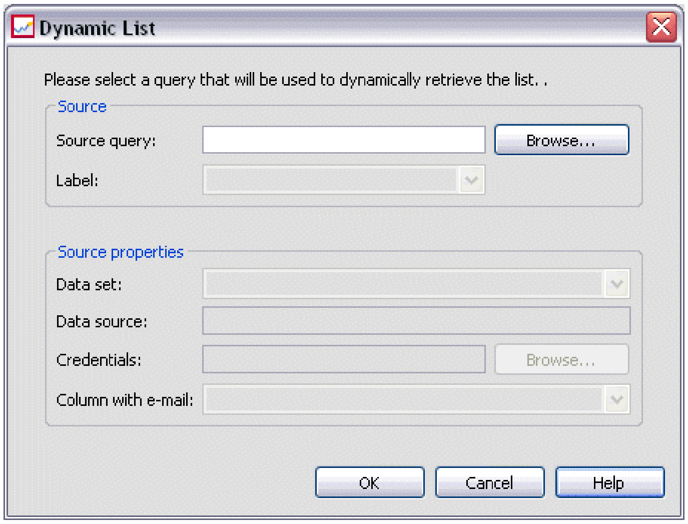
2. After all recipients have been selected, click OK. The e-mail addresses of the selected recipients appear in the To list of the Notification for Job Success dialog box.

Dynamic Notification Lists

A list of notification recipients can also be generated dynamically using a database query. The input for the notification list can be the results of a Business Intelligence Reporting Tools (BIRT) report (**.rptdesign*) or a ShowCase Query file (**.dbq*).

To generate a list of notification e-mail addresses dynamically, in the Notification for Job Success dialog box, click the down arrow next to the To field and then choose Dynamic List. The Dynamic List dialog box appears.

Figure 13-7
Dynamic List dialog box



1. To select the file that will be used to dynamically retrieve the list of recipients, click Browse next to the Source query field and select the file from the Content Repository. Then select the labeled version to use.

2. Select the data set, data source, credentials, and the column containing the list of e-mail addresses to retrieve from the source query:

   - **Data set** is the name of the database.
   - **Data source** is the URL of the server.
   - **Credentials** come from PASW Collaboration and Deployment Services resource definitions and must provide access to the specified data set and data source.
   - **Column with e-mail** is the database column containing the address value retrieved by the query.

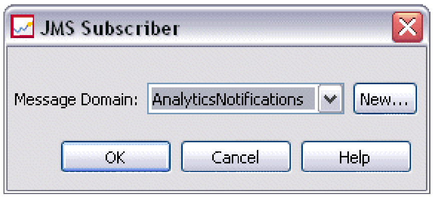
3. After all recipients have been selected, click OK. The source file name, version, data set, credentials and column appear in the To list of the Notification for Job Success dialog box.

JMS Subscriber

Notification events can generate JMS (Java Messaging Service) messages that trigger internal PASW Collaboration and Deployment Services processing as well as external applications. For example, a PASW Collaboration and Deployment Services user can set up JMS notification for a PASW Modeler stream to rerun a job which includes the stream when a new version of the stream is created. For that, a JMS message domain must first be created. Then a message-based schedule must be set up for the job specifying the resource ID of the stream in the message selector. For more information, see the topic Message-Based Schedule Settings in Chapter 11 on p. 254. And finally, a notification must be set up for the PASW Modeler stream based on the JMS subscriber to be triggered when a new version of the stream is created.

To specify a JMS subscriber, in the Notifications dialog box click the down arrow next to the To field and then choose JMS Subscriber. The JMS Subscriber dialog box appears.

Figure 13-8
JMS Subscriber dialog box



Message Domain. The JMS message domain for the subscriber. For more information, see the topic Message Domains in Chapter 6 on p. 94.

1. Select the message domain. A message domain can only be selected once for a given notification. To define a new message domain, click New ....

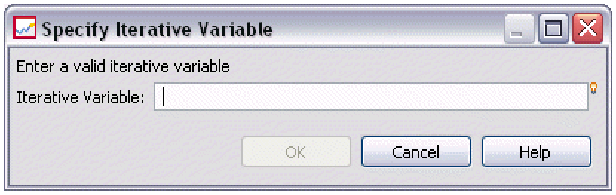
2. Click OK. The JMS subscriber will appear in the To list of the Notifications dialog box.

Iterative Address Variables

Recipient addresses can also be specified as variables for iterative reporting job step success and failure notifications. In such cases, the variables defined within the step are used to retrieve the recipient address value every time the step is executed. For more information, see the topic Type for Reporting Steps in Chapter 14 on p. 296.

To add a notification e-mail as an iterative variable, in the Notifications dialog box, click the down arrow next to the To field and then choose Iterative Subscriber. The Specify Iterative Variable dialog box appears.

Figure 13-9
Specify Iterative Variable dialog box



1. Enter the variable in the Iterative Variable field. Use field content assist if necessary. For more information, see the topicEntry Field Content Assist in Chapter 3 on p. 12.

2. After all recipients have been selected, click OK. The specified variables appear in the To list of the Notifications dialog box.

Notification Message Attachments

Content notification messages can include file attachments. If an attachment is specified for a file or a job notification, the new version of the file or the job is included in the message. If an attachment is specified for a folder notification, the object that triggers folder notification (new file or job, or the new version of an existing file or job) is included in the message. Attachments are specified for individual notification recipients separately.

To include a file attachment in a notification message:

▶ In the To list, click in the Attachment column next to the notification recipient and choose Yes or No from the drop-down.

Customizing the Notification Message

Notification message can be customized for individual notifications. The subject line and body text of the notification e-mail message are defined by the default message template assigned to the corresponding event type. The Subject and Message fields of the Notification screen contain the default text, variable property values, and for HTML templates, formatting markup. If multiple objects are selected with varying sender, subject, and body text field values, the fields are displayed as blank. Modifying the values of these fields will apply the changes to all selected objects.

▶ To change the subject or message body text or formatting, type the changes in the corresponding fields.

If you make changes to the HTML formatting, you must make sure that the changes are valid. Invalid template may cause notification to fail.

▶ You can use content assist to insert system property variable values into the From field, message subject, or body. For more information, see the topic Entry Field Content Assist in Chapter 3 on p. 12.

▶ Click Preview to display the message. If the template cannot be parsed, an error message is displayed. For more information, see the topic Notification Message Preview on p. 288.

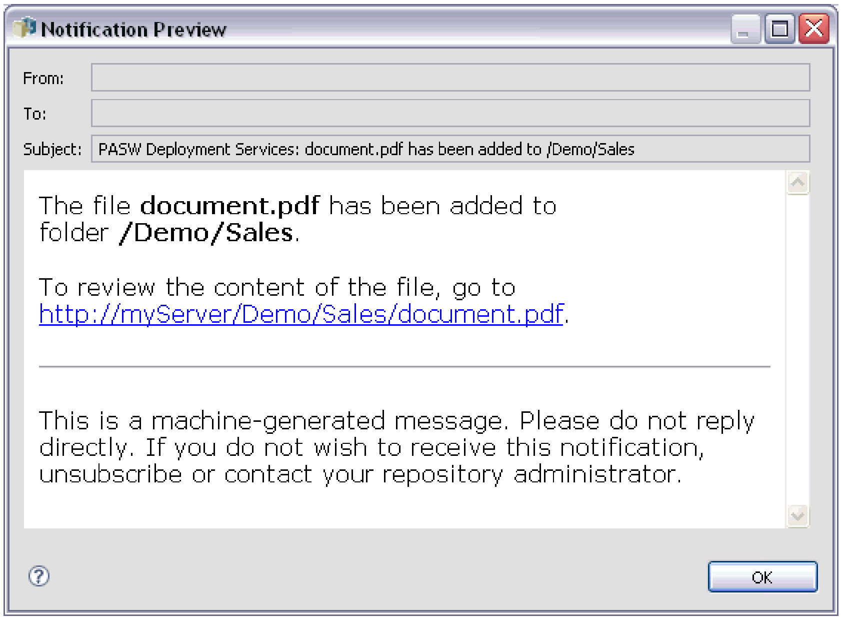
The customized message is saved for a specific notification. Customizing messages for individual notifications does not alter the default templates.

Note: The appearance and content of the notification message can also be customized by modifying the default template associated the notification event. Contact your PASW Collaboration and Deployment Services administrator to request global template changes.

Notification Message Preview

To preview notification on the notification screen, click Preview. The Notification Preview dialog box appears.

Figure 13-10
Notification Preview dialog box



From:

To:

Subject: PASW Deployment Services: document.pdf has been added to /Demo/Sales

The file **document.pdf** has been added to folder **/Demo/Sales**.

To review the content of the file, go to http://myServer/Demo/Sales/document.pdf.

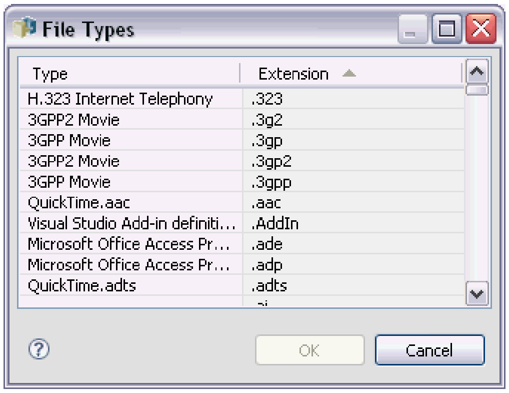
This is a machine-generated message. Please do not reply directly. If you do not wish to receive this notification, unsubscribe or contact your repository administrator.

The preview displays the e-mail message resulting from merging the template with sample values. The preview includes the specified sender and recipient addresses and customizations made to the message subject, text, and formatting. For more information, see the topic Customizing the Notification Message on p. 288. If the customized message cannot be previewed due to invalid formatting, an error message appears.

Notification File Types

To limit notifications to certain file types within the folder, in the Folder Options editor of the Notifications dialog box, click the ellipsis button. The File Types dialog box appears.

Figure 13-11
File Types dialog box



Type	Extension
H.323 Internet Telephony	.323
3GPP2 Movie	.3g2
3GPP Movie	.3gp
3GPP2 Movie	.3gp2
3GPP Movie	.3gpp
QuickTime.aac	.aac
Visual Studio Add-in definiti...	.AddIn
Microsoft Office Access Pr...	.ade
Microsoft Office Access Pr...	.adp
QuickTime.adts	.adts

Select the file types for the notification.

1. Click the file type entries to select or deselect. Use the Ctrl or Shift keys to select multiple entries.

2. After all file types have been selected, click OK.

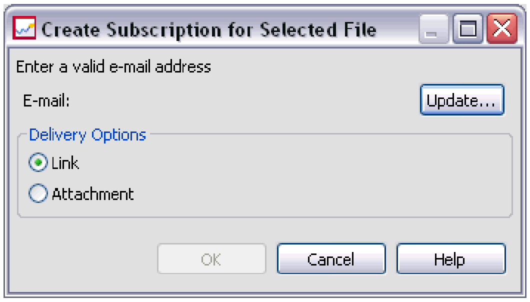
Subscriptions

repository enables the user to subscribe to PASW Collaboration and Deployment Services files and jobs. Unlike notifications, subscriptions are defined by individual users for themselves. A subscriber receives e-mail messages when changes occur to the contents of the object and a new version is created. The e-mail message contains a link to the URL of the changed repository object or a file attachment. The Content Explorer also provides the ability to manage subscriptions to individual objects. This may be necessary, for example, when an administrator must remove a subscription for a user who is no longer employed with the company. A user must be assigned corresponding actions through roles to subscribe to content and manage subscriptions.

Subscribing to Files

To create a subscription to a file, right-click the file name and choose Subscriptions. The Create Subscription for Selected File dialog box appears.

Figure 13-12
Create Subscription for Selected File dialog box



Create Subscription for Selected File

Enter a valid e-mail address

E-mail:

Delivery Options

● Link

○ Attachment

E-mail. The e-mail address of the user. By default, the e-mail address specified in server preferences is used. For more information, see the topic Subscriptions Recipient in Chapter 5 on p. 59.

Link. Provide a link to the URL of the file to which the recipient is subscribed.

Attachment. Attach the file to which the recipient is subscribed to the message.

1. If the default e-mail address has not been set for the user, click Update. The Set User Preferences dialog box appears. Update the address. For more information, see the topicSubscription E-mail Address on p. 291.

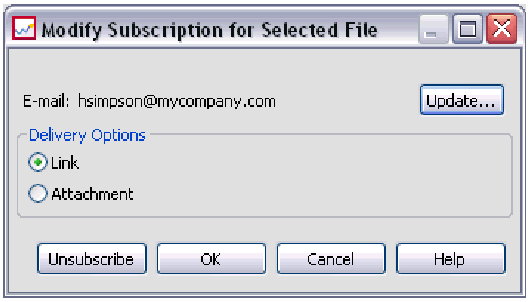
2. Specify whether a file link or attachment is included in the e-mail message.

3. Click OK.

Modifying File Subscriptions and Unsubscribing from Files

To modify a subscription for a file or a repository job, or to unsubscribe, right-click the file name and choose Subscriptions. The Modify Subscription for Selected File dialog box appears.

Figure 13-13
Modify Subscription for Selected File dialog box



E-mail: hsimpson@mycompany.com

Delivery Options
● Link
○ Attachment

E-mail. The e-mail address of the user.

Link. Provide a link to the URL of the file to which the recipient is subscribed.

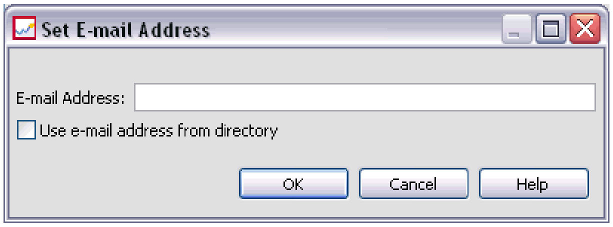
Attachment. Attach the file to which the recipient is subscribed to the message.

Modify subscription settings as necessary and click OK, or click Unsubscribe to remove the subscription.

Subscription E-mail Address

To change the default e-mail address for subscriptions, in the Create/Modify Subscriptions for Selected File dialog box. click Update. The Set E-mail Address dialog box appears.

Figure 13-14
Set E-mail Address dialog box



E-mail Address:

□ Use e-mail address from directory

E-mail Address. The e-mail address of the user. The address string must contain a @ character and a period. The default address preference for the user will be modified and all existing subscriptions will use the new address. For more information, see the topic Subscriptions Recipient in Chapter 5 on p. 59.

Note: The Use e-mail address from directory option is available only for users authenticated by Active Directory and Active Directory with Local Override. The e-mail address from the directory is used by default.

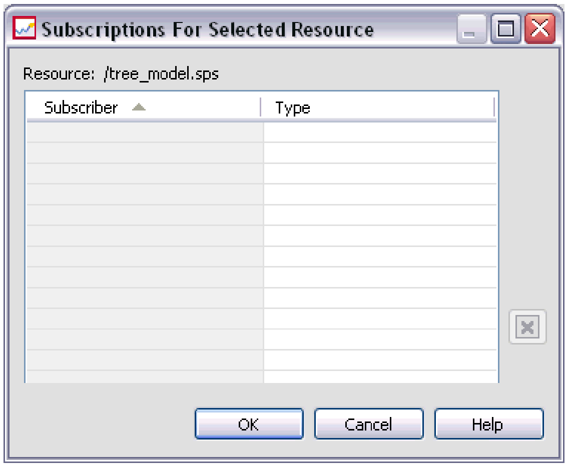
1. Enter the e-mail address or select the Use e-mail address from directory option.

2. Click OK. The address will appear in the To list of the Notifications dialog box.

Managing Subscriptions

To manage subscriptions to a repository object, right-click the object and choose Manage Subscriptions. The Subscriptions for Selected Resource dialog box appears.

Figure 13-15
Subscriptions for Selected Resource dialog box



Resource: /tree_model.sps

The list of subscriptions displays the name of the user and the subscription delivery type (link or attachment). Existing subscriptions can be removed.

1. Select the subscription. Use the Ctrl and Shift keys to select multiple entries.
2. Click the X button.
3. Click OK.

Delivery Failures

Notification Message Delivery Failures

If a notifications e-mail messages cannot be delivered to a specified recipient, a warning message similar to the following will be generated.

Your message did not reach some or all of the intended recipients.

Subject: PASW Deployment Services: Job ChurnAnalysis failed
Sent: 4/5/2007 2:21 PM

The following recipient(s) could not be reached:

Jsmiht@mycompany.com on 4/5/2007 2:21 PM

The e-mail account does not exist at the organization this message was sent to.

```
Check the e-mail address, or contact the recipient directly to find out the correct address.
<smtp.mycompany.com #5.1.1>
```

The message is sent to the address indicated in the From field of the Notifications dialog box. If the address is invalid, the message is sent to the default PASW Collaboration and Deployment Services administrator's address. To correct the problem with the notifications delivery, verify that notifications recipients are specified correctly. Notification delivery problems can also occur due to e-mail server setup, network configuration, and so on. If notifications delivery problems persist, contact your system administrator.

Subscription Message Delivery Failures

If you do not receive messages from subscriptions that have been defined, verify that your default subscriptions address is specified correctly. For more information, see the topic Subscriptions Recipient in Chapter 5 on p. 59. If subscription message delivery problems persist, contact your PASW Collaboration and Deployment Services administrator. The administrator is also notified of any delivery failures by system-generated messages.

Chapter
14
Reporting Job Steps

A reporting step submits report syntax to an execution server. For example, a reporting step using an **.rptdesign* report submits **Business Intelligence and Reporting Tools (BIRT)** report syntax to the BIRT execution engine. The PASW BIRT Report Designer version shipped with PASW Collaboration and Deployment Services is supported, and can be installed from the PASW Collaboration and Deployment Services media or from the repository. For details about ShowCase Query/Report Writer (separately installed SPSS Inc. applications), see your ShowCase Suite documentation.

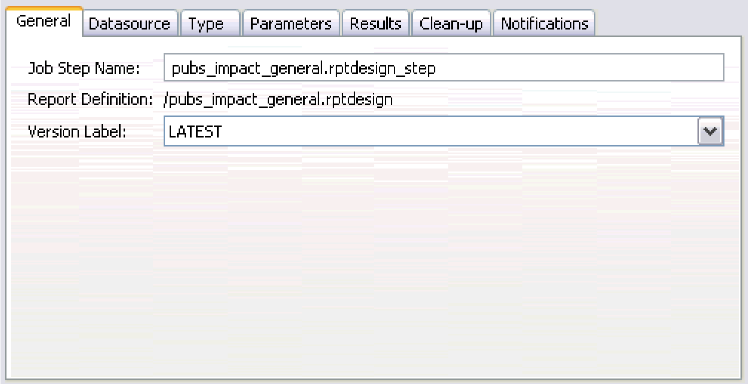
To add a reporting step to a job, drag a report from the Content Explorer to the job canvas. However, before the step can successfully run, several properties of the reporting step must be defined. These properties can be classified into several categories:

- General properties, such as the job step name and version of the report.
- Data source properties, including the data source used by the report and its credentials.
- Type properties, such as whether the report type is a single run or an iterative run.
- Parameter properties, such as prompt variable values.
- Result properties, such as output file format and location.
- Clean-up properties, for automatically moving, deleting, or expiring existing output when a job runs.
- Notification properties, for setting up e-mail notifications.

General Properties for Reporting Steps

The general properties for a reporting step specify what runs and which server processes the step when the job runs. To define general properties for a reporting step, click the step in an open job. Click the General tab to view or edit the step properties.

Figure 14-1
General reporting step properties



Job Step Name: pubs_impact_general.rptdesign_step
Report Definition: /pubs_impact_general.rptdesign
Version Label: LATEST

Job Step Name. Type a name for the step. The default name appends a *_step* suffix to the filename.

Report Definition. Displays the location of the report used for the step.

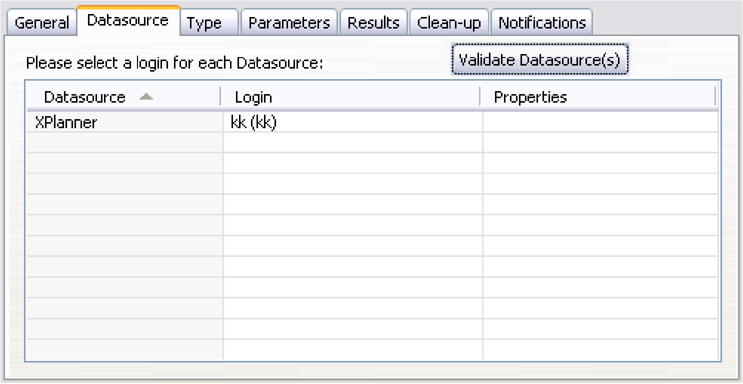
Version Label. Select the version of the report submitted to the execution server using the drop-down list.

Use internal Report Services or **Use external Enterprise Server.** If working with a ShowCase query or report, you can also specify whether to process the reporting step using the default Report Services or a configured Enterprise Server PC. To select an Enterprise Server, use the drop-down list and choose <auto-select> or a select a specific Enterprise Server PC. The list contains all servers currently configured. For details about ShowCase queries and reports, and Enterprise Servers, see your ShowCase Suite documentation.

Data Source for Reporting Steps

To schedule and run reporting job steps, you need to specify the data source login credentials for the report used in the reporting step. To define data source login credentials, click the step in an open job. Click the Datasource tab to view or edit the data source credentials.

Figure 14-2
Datasource for reporting steps



Please select a login for each Datasource:

Datasource	Login	Properties
XPlanner	kk (kk)	

Click the Login cell for a data source to select the credentials from those defined in the system.

If the data source for the report uses single sign-on credentials for accessing the data, the report cannot be included in a scheduled job. The credentials for the current user are not stored with the job, and when the job would execute, valid credentials would be needed. The report must specify credentials as a credential defined in the repository or as a username/password pair.

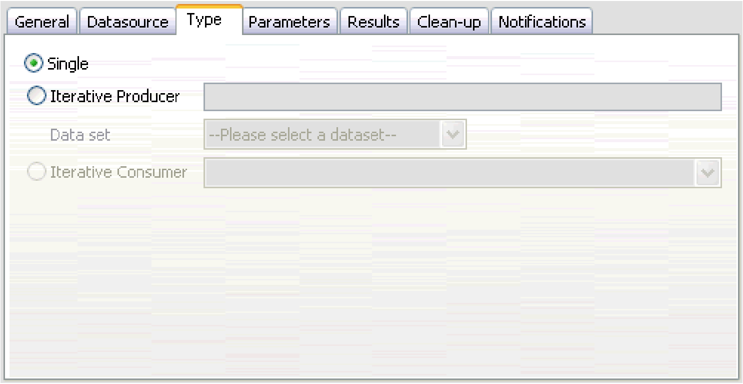
If a data source for the report is based on the Enterprise View, the data provider definition must be specified. Click on the Properties cell for the data source to select the data provider definition from those defined in the system. For more information, see the topic Data Provider Definition in Chapter 8 on p. 155.

To verify that the specified login information can successfully connect to the data source, click the Validate Datasource(s) button. The system reports any problematic login specifications that must be corrected for the job step to run successfully.

Type for Reporting Steps

To define report type properties, click a reporting step in an open job. Click the Type tab to view or edit the step properties.

Figure 14-3
Type for reporting steps



● Single
○ Iterative Producer
Data set: --Please select a dataset--
○ Iterative Consumer

On the Type tab, select the report type for the reporting step. The types available depend on which reporting applications you have installed and which file type you are working with.

Single

A *single* report runs once and returns a single output file. If the report contains variables, you can specify values for the prompt variables on the Parameters tab (see Parameters for Reporting Steps on p. 298 for more information). At run time, the report generates output based on the specified variable values.

Iterative

An *iterative* job runs once but runs in multiple iterations using multiple input parameter mappings. An Iterative Producer report file produces the set of input parameters used for each report iteration. Each iteration produces unique output. For example, to schedule a report to run for each of five departments in your company, you do not need to schedule it five separate times. Instead, you schedule it once and specify variables that run the report multiple times—once for each variable value. The report is scheduled once, but when it runs at the scheduled time, it generates multiple output files—one for each of your five departments.

An iterative job contains an Iterative Producer step and an Iterative Consumer step:

1. Add a new reporting step to the job. The report should contain the list of values to iterate on (for example, a list of departments in your company).
2. On the Type tab, select Iterative Producer. Type a name for the iterative producer (to be used by the iterative consumer) and select the data set from the report. Verify the report runs successfully.
3. Add a second reporting step to the job.
4. On the Type tab, select Iterative Consumer. Select the iterative producer name created in step 2. When the job runs, the iterative consumer job step pulls values from the iterative producer job step to produce the report output iterations.

Notes and Requirements:

- Iterative Producer job steps cannot contain prompts.
- You must assign Parameters for Iterative Consumer job steps. See Parameters for Reporting Steps on p. 298.
- Iterative Producer and Iterative Consumer job steps must use the same data source.
- For BIRT reports, only JDBC data sources are supported for the Iterative Producer—not other BIRT data source types (such as XML).
- BIRT reports can generate iteration source data, or be used to render a report, but not both.

Burst

A *burst* report is scheduled once, runs once, yet generates multiple output files by using multiple input parameters taken from *break groups* in the report. Break groups are rows of returned data grouped according to a common column value. For example, in a column of states, the rows of data for each state are grouped together. The break groups produce the set of input parameters used for each report output. For example, to schedule a report for each of five departments in your company, you do not need to schedule it five separate times. Instead, you can schedule it once and specify a break group column that will produce multiple output files—one for each break group value. The report will be scheduled once, but when it runs at the scheduled time it will generate multiple output files—one for each of your five departments.

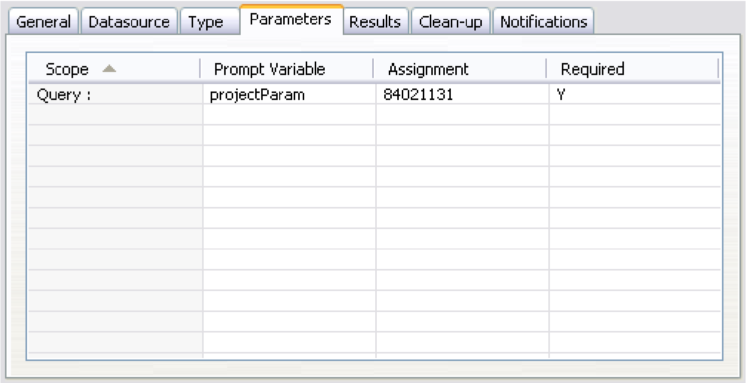
This option is only available for ShowCase queries and reports. For details about ShowCase queries and reports, see your ShowCase Suite documentation. To use the Burst option, your report must contain break groups. For a complete list of restrictions for the Burst report type, see the *ShowCase Suite Installation Guide*.

Parameters for Reporting Steps

If the report contains variables (also known as prompts), you can specify values for the variables on the Parameters tab. At run time, the report generates output based on the specified prompt variable values. The number of prompt variables available for selection is dependent on the number of prompts in the source file and the options used to define those prompts when the file was created. You can only assign prompt variables if your source file contains prompts.

To define parameters, click a reporting step in an open job. Click the Parameters tab to view or edit the step properties.

Figure 14-4
Parameters for reporting steps



Scope	Prompt Variable	Assignment	Required
Query :	projectParam	84021131	Y

With your cursor in the Assignment column, click the button on the right edge of the column. The Prompt Variable Values dialog box will be displayed. Select the variable value to be used for the report. The Required column indicates whether or not the report requires a value for the parameter.

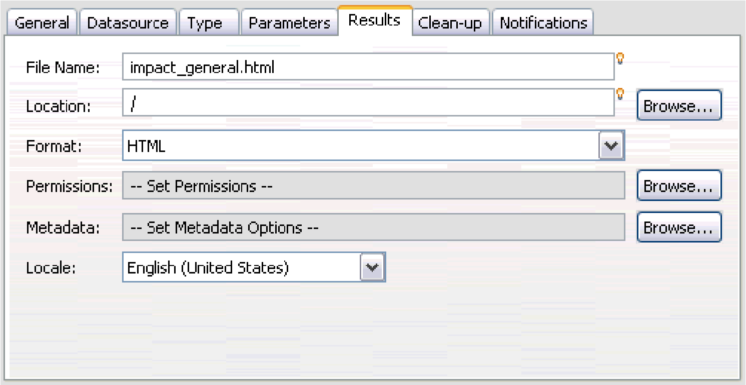
Notes:

- Parameters must be defined for Iterative Consumer job steps. See Type for Reporting Steps on p. 296.
- BIRT reports with cascading prompt variables can run in Deployment Portal but are not supported in Deployment Manager.
- For BIRT reports, the prompt variable sources must match the data source of the report.

Results for Reporting Steps

Execution of a reporting step yields report output. To define properties for this output, click a reporting step in an open job. Click the Results tab to view or edit the step properties.

Figure 14-5
Results for reporting steps



File Name: impact_general.html

Location: /

Format: HTML

Permissions: -- Set Permissions --

Metadata: -- Set Metadata Options --

Locale: English (United States)

The Format drop-down identifies the different result types for reporting steps. Properties for each result type include:

File Name. Define the filename for the results. You can insert variables into the file output name, and various other fields. To insert variables, position your cursor in the desired location in the File Name field, type $, and select one or more values from the drop-down. These variables can help users distinguish between result output. You can insert variables for information such as the date or time. When the report runs as scheduled, the variable information is inserted into the output file name(s). For example, if you have a scheduled report running daily, a user may have trouble determining the information they are viewing. Adding a date variable will insert the current date into the file name at run time, indicating the date the report ran. For more information, see Entry Field Content Assist on p. 12.

Location. Define the location in which to save the results by typing a path or clicking Browse. The dialog box will appear. If the specified folder does not exist, it will be created automatically at run time. To insert variables to append the date and/or time to the folder name at run time, see Entry Field Content Assist on p. 12.

Format. Select the desired file format for reporting step output. For example, select PDF to produce Portable Document Format (*.*pdf*) output. For reports containing images, use the *HTML Archive* format to generate a single archive file containing an HTML report and its ancillary files. This archive can be viewed in a variety of PASW Collaboration and Deployment Services clients, including Deployment Portal and Deployment Manager.

If the output format is specified as HTML for a report with images, an *.mht* file is generated. This allows all the images and HTML to be combined into a single file. Internet Explorer is able to open and render *.mht* files automatically.

Permissions. Define access permissions for the results by clicking Browse. The Output Permissions dialog box will appear. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.

Metadata. Define output properties for the results by clicking Browse. The Output Properties dialog box will appear. See Output File Metadata on p. 247.

Locale. Define the locale to use for the results by selecting from the list.

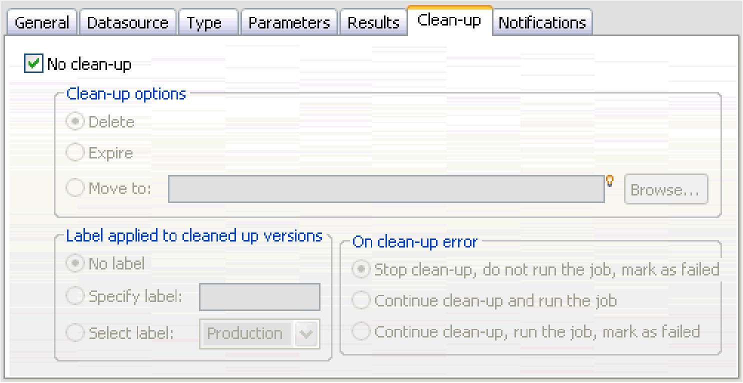
Rendering Options. Select from any of the following rendering options.

- Select Save column headings with data to display the source column headings with your output.
- Select Apply edit masks to data to apply edit masks to your output data. An edit mask is a predefined format users can apply to data in a viewer.
- Select Output report details only to display only default data in your output. This option suppresses break group and summary information such as subtotals, grand totals, and counts that are part of your source report.
- Select Include summary text with data to display summary text such as subtotals, grand totals, and counts in your output data. Summary text is the descriptive text that displays to the left of the summary line data within a break group. This option is disabled if the Output report details only option is selected.

Clean-up for Reporting Steps

To define options for cleaning up existing output when a job runs, select a job step in an open job. Select the Clean-up tab to view or edit the step's clean-up options. We recommend specifying clean-up options every time you create a new job step, and then disabling or enabling clean-up later as necessary.

Figure 14-6
Clean-up for reporting steps



☑ No clean-up

Clean-up options

● Delete
○ Expire
○ Move to:

Label applied to cleaned up versions

● No label
○ Specify label:
○ Select label: Production

On clean-up error

● Stop clean-up, do not run the job, mark as failed
○ Continue clean-up and run the job
○ Continue clean-up, run the job, mark as failed

Important: Clean-up options are only applied if saving to the repository.

Choose from the following options.

- **No clean-up**. Output files from the previous run of the current job are left intact.

Clean-up Options

- **Delete**. Output files from the previous run of the current job will be deleted (even if they have since been moved to a different folder in the repository).
- **Expire**. Output files from the previous run of the current job will be expired. For more information, see the topic Working with Expiration Dates and Expired Files in Chapter 5 on p. 50.
- **Move to**. Output files from the previous run of the current job will be moved to a different folder. Click Browse or type a target folder for old output files. If a file with the same name already exists in the Move to folder, it will be overwritten. If the specified folder does not exist,

it will be created automatically during the clean-up process. To insert variables to append the date and/or time to the file name at run time, see Entry Field Content Assist on p. 12.

Label applied to cleaned-up versions

- **No label**. Select this option to specify no label is applied to the moved output.
- **Specify label**. Select this option to specify a new label to apply to the moved output.
- **Select label**. Select this option to select an existing label to apply to the moved output.

On Clean-up Error

- **Fail immediately**. Select this option if you want the job to fail immediately if it encounters errors while attempting clean-up.
- **Continue and mark as success**. Select this option if you want the job to keep running if it encounters errors while attempting clean-up. If the job runs successfully, the clean-up error will not trigger a failed job history message.
- **Continue and mark as failed**. Select this option if you want the job to keep running if it encounters errors while attempting clean-up. If the job runs successfully, the clean-up error will trigger a failed job history message.

Notifications for Reporting Steps

To define e-mail notification properties, click a reporting step in an open job. Click the Notifications tab to view or edit the step properties. For details about notifications, see Job Step Notifications on p. 279.

Chapter

15

Visualization Report Job Steps

A visualization report generates a visual representation of the information contained in a data source. The report is typically created using the PASW Viz Designer, which provides an advanced visualization environment enabling users to create graphs ranging from basic business charts to rich, interactive representations using data from a variety of sources. Within the PASW Viz Designer, users can access and explore data and define both the structure and the style of graphs. The application provides a range of deployment options for the visualization specifications and resulting graphs; for example, they can be stored in repository. Once in the repository, graph specifications can be associated with data as needed, rendered on demand, rendered on a recurring basis, and served to applications and web pages as required.

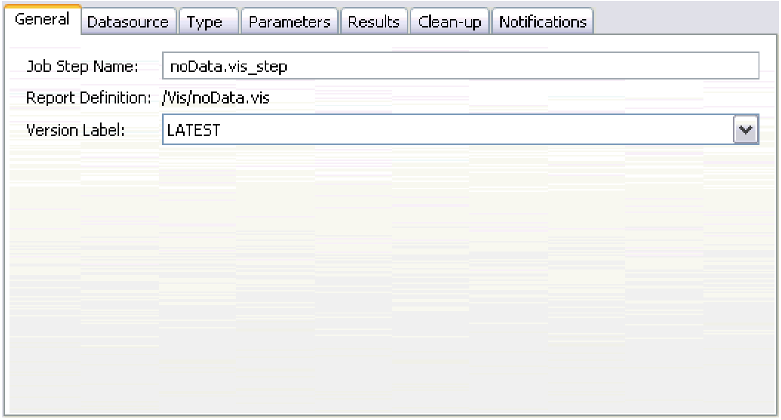
To add a visualization report step to a job, drag a *.viz* file from the Content Explorer to the job canvas. However, before the step can successfully run, several properties of the visualization reporting step must be defined. These properties can be classified into several categories:

- General properties, such as the job step name and version of the report.
- Data source properties, including the data source used by the report and its credentials.
- Type properties, such as whether the report type is a single run or an iterative run.
- Parameter properties, such as prompt variable values.
- Result properties, such as output file format and location.
- Clean-up properties, for automatically moving, deleting, or expiring existing output when a job runs.
- Notification properties, for setting up e-mail notifications.

General Properties for Visualization Report Steps

The general properties for a visualization report step identify what runs when the job executes. To define general properties for the step, click the step in an open job. Click the General tab to view or edit the step properties.

Figure 15-1
General tab for visualization report steps



Job Step Name: noData.vis_step
Report Definition: /Vis/noData.vis
Version Label: LATEST

Job Step Name. Type a name for the step. The default name appends a *_step* suffix to the filename.

Report Definition. Displays the location of the report used for the step.

Version Label. Select the version of the report submitted for processing using the drop-down list.

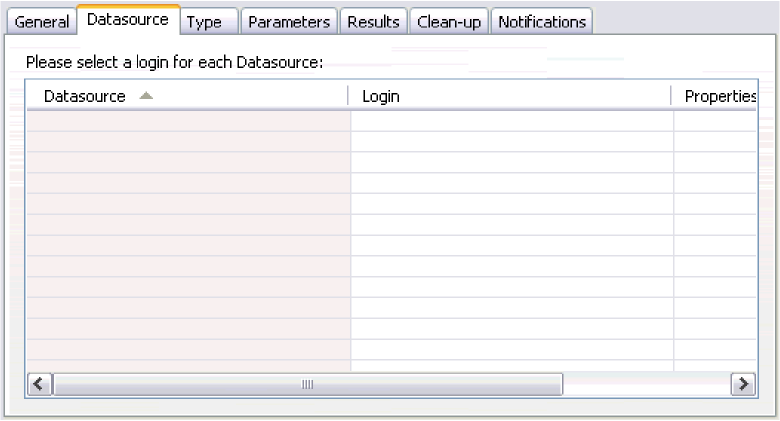
Data Source for Visualization Report Steps

Visualization reports support a variety of input data sources, including:

- Delimited text files
- PASW Statisticsdata files (*.sav*)
- Dimensions Data Models
- PASW Collaboration and Deployment Services reports
- JDBC sources

Data sources based on files can access files stored on file systems or in the repository. For data sources based on reports or JDBC, you need to specify the data source login credentials to access the data. To define data source login credentials, click the step in an open job. Click the Datasource tab to view or edit the data source credentials.

Figure 15-2
Datasource tab for visualization report steps



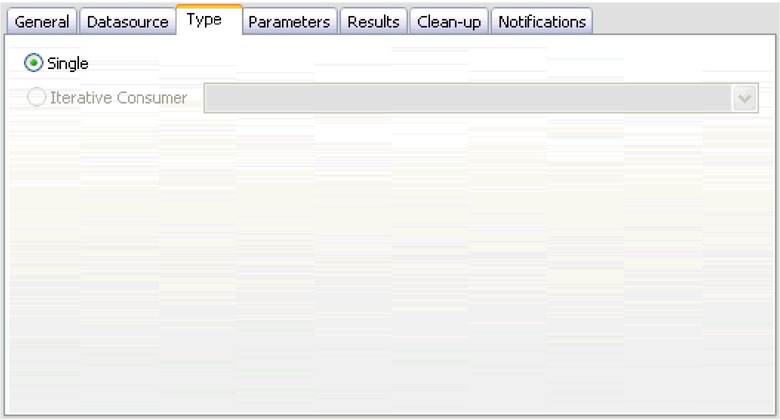
Please select a login for each Datasource:

Datasource	Login	Properties

Type for Visualization Report Steps

To define report type properties, click a visualization report step in an open job. Click the Type tab to view or edit the step properties.

Figure 15-3
Type tab for visualization report steps



● Single

○ Iterative Consumer

On the Type tab, select the report type for the reporting step.

Single. Runs once and returns a single output file. If the report contains variables, you can specify values for the prompt variables on the Parameters tab (see Parameters for Visualization Report Steps on p. 305 for more information). At run time, the report generates output based on the specified variable values.

Iterative Consumer. Runs once but runs in multiple iterations using multiple input parameter mappings. An iterative producer report file produces the set of input parameters used by an iterative consumer report iteration. Each iteration produces unique output. For example, to schedule a report to run for each of five departments in your company, you do not need to schedule it five separate times. Instead, you schedule it once and specify variables that run the report multiple times—once for each variable value. The report is scheduled once, but when it runs at the scheduled time, it generates multiple output files—one for each of your five departments.

An iterative job contains an Iterative Producer step and an Iterative Consumer step:

1. Add a new reporting step to the job. The report should contain the list of values to iterate on (for example, a list of departments in your company).

2. On the Type tab, select Iterative Producer. Type a name for the iterative producer (to be used by the iterative consumer). Verify the report runs successfully.

3. Add a visualization reporting step to the job, connecting it as a consequent step of the iterative producer using a relationship connector.

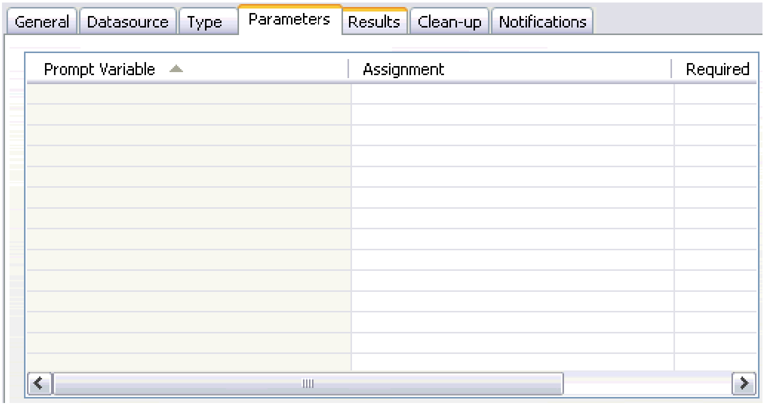
4. On the Type tab for the visualization report step, select Iterative Consumer. Select the iterative producer name created in step 2. When the job runs, the iterative consumer job step pulls values from the iterative producer job step to produce the report output iterations.

Parameters for Visualization Report Steps

Report data sources may contain variables, or **prompts**, that need an assigned value for proper processing. At run time, the report generates output based on the specified prompt variable values. The number of prompt variables available for selection is dependent on the number of prompts in the source file and the options used to define those prompts when the file was created. You can only assign prompt variables if your source file contains prompts.

To define prompt variables, click a reporting step in an open job. Click the Parameters tab to view or edit the step properties.

Figure 15-4
Parameters tab for visualization report steps



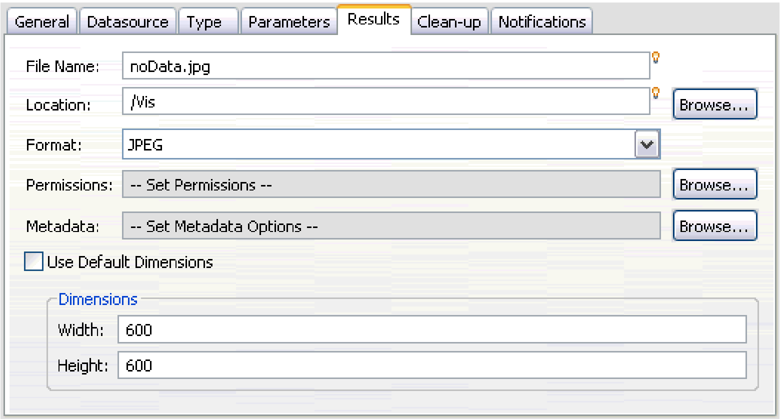
Prompt Variable	Assignment	Required

With your cursor in the Assignment column, click the button on the right edge of the column. The Prompt Variable Values dialog box will be displayed. Select the variable value to be used for the report. Select N if no value is required for this prompt variable at run time, and you don’t want to restrict the report results based on this prompt value (often referred to as a “null” value).

Results for Visualization Report Steps

Execution of a reporting step yields report output. To define properties for this output, click a reporting step in an open job. Click the Results tab to view or edit the step properties.

Figure 15-5
Results tab for visualization report steps



File Name: noData.jpg

Location: /Vis

Format: JPEG

Permissions: -- Set Permissions --

Metadata: -- Set Metadata Options --

□ Use Default Dimensions

Dimensions

Width: 600

Height: 600

File Name. Define the filename for the results. You can insert variables into the file output name, and various other fields. To insert variables, position your cursor in the desired location in the File Name field, type $, and select one or more values from the drop-down. These variables can help users distinguish between result output. You can insert variables for information such as the date or time. When the report runs as scheduled, the variable information is inserted into the output file name(s). For example, if you have a scheduled report running daily, a user may have trouble determining the information they are viewing. Adding a date variable will insert the current date into the file name at run time, indicating the date the report ran. For more information, see Entry Field Content Assist on p. 12.

Location. Define the location in which to save the results by typing a path or clicking Browse. The Results Location dialog box will appear. If the specified folder does not exist, it will be created automatically at run time. To insert variables to append the date and/or time to the folder name at run time, see Entry Field Content Assist on p. 12.

Format. Select the desired file format for reporting step output. For example, select PDF to produce Portable Document Format (**.pdf*) output.

Permissions. Define access permissions for the results by clicking Browse. The Output Permissions dialog box will appear. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.

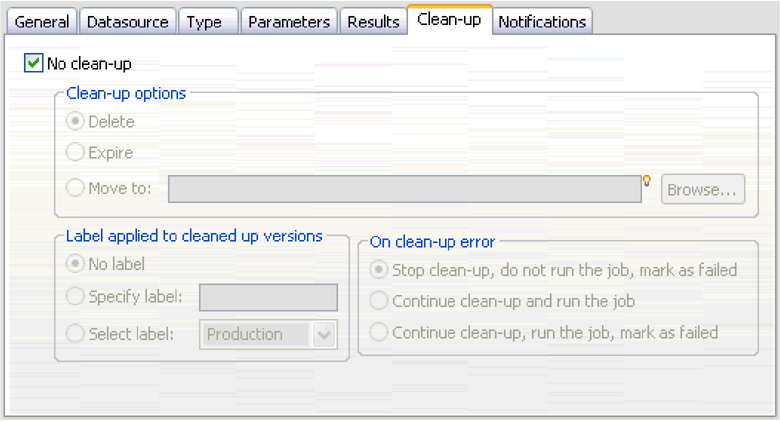
Metadata. Define output properties for the results by clicking Browse. The Output Properties dialog box will appear. See Output File Metadata on p. 247.

Dimensions. Define the size of the resulting visualization output by specifying values for the output height and width. Alternatively, select Use Default Dimensions to accept the default output size.

Clean-up for Visualization Report Steps

To define options for cleaning up existing output when a job runs, select a job step in an open job. Select the Clean-up tab to view or edit the step's clean-up options. We recommend specifying clean-up options every time you create a new job step, and then disabling or enabling clean-up later as necessary.

Figure 15-6
Clean-up tab for visualization report steps



☑ No clean-up

Clean-up options

● Delete

○ Expire

○ Move to:

Label applied to cleaned up versions

● No label

○ Specify label:

○ Select label: Production

On clean-up error

● Stop clean-up, do not run the job, mark as failed

○ Continue clean-up and run the job

○ Continue clean-up, run the job, mark as failed

Important: Clean-up options are only applied if saving to the repository.

Choose from the following options.

- **No clean-up**. Output files from the previous run of the current job are left intact.

Clean-up Options

- **Delete**. Output files from the previous run of the current job will be deleted (even if they have since been moved to a different folder in the repository).
- **Expire**. Output files from the previous run of the current job will be expired. For more information, see the topic Working with Expiration Dates and Expired Files in Chapter 5 on p. 50.
- **Move to**. Output files from the previous run of the current job will be moved to a different folder. Click Browse or type a target folder for old output files. If a file with the same name already exists in the Move to folder, it will be overwritten. If the specified folder does not exist, it will be created automatically during the clean-up process. To insert variables to append the date and/or time to the file name at run time, see Entry Field Content Assist on p. 12.

Label applied to cleaned-up versions

- **No label**. Select this option to specify no label is applied to the moved output.
- **Specify label**. Select this option to specify a new label to apply to the moved output.
- **Select label**. Select this option to select an existing label to apply to the moved output.

On Clean-up Error

- **Fail immediately**. Select this option if you want the job to fail immediately if it encounters errors while attempting clean-up.
- **Continue and mark as success**. Select this option if you want the job to keep running if it encounters errors while attempting clean-up. If the job runs successfully, the clean-up error will not trigger a failed job history message.
- **Continue and mark as failed**. Select this option if you want the job to keep running if it encounters errors while attempting clean-up. If the job runs successfully, the clean-up error will trigger a failed job history message.

Notifications for Visualization Report Steps

To define e-mail notification properties, click a reporting step in an open job. Click the Notifications tab to view or edit the step properties. For details about notifications, see Job Step Notifications on p. 279.

Chapter

16

SAS® Job Steps

A SAS step submits SAS syntax to a SAS execution server. This server corresponds to running SAS from the command line, using invocation options to control processing. Step output consists of a log file detailing the processing history and a text listing file containing the procedure results. If desired, the Output Delivery System (ODS) can be used within the syntax to generate HTML results.

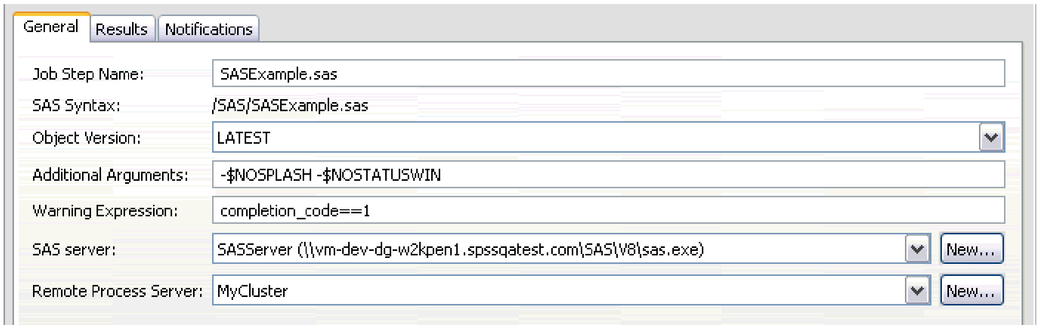
To add a SAS step to a job, drag a SAS syntax file from the Content Explorer to the job canvas. However, before the step can be successfully run, several properties of the SAS step must be defined. These properties can be classified into two categories:

- General properties, such as the execution server to process the step
- Result properties, such as output format and location

General Properties for SAS Steps

The general properties for a SAS step specify what gets run and which server processes the step when the job is run. To define general properties for a SAS step, click the step in an open job. Click the General tab to view or edit the step properties.

Figure 16-1
SAS general properties



Job Step Name: SASExample.sas

SAS Syntax: /SAS/SASExample.sas

Object Version: LATEST

Additional Arguments: -$NOSPLASH -$NOSTATUSWIN

Warning Expression: completion_code==1

SAS server: SASServer (\\vm-dev-dg-w2kpen1.spssqatest.com\SAS\V8\sas.exe)

Remote Process Server: MyCluster

Job step name. Type a name for the step. The default name appends a *_step* suffix to the file name.

Object version. Select the version of the file submitted to the execution server using the drop-down list.

Additional arguments. Define optional system options that get passed to the SAS execution server when the job is run.

Warning expression. Define warnings for job steps connected by a Conditional connector. The warning expression (e.g., completion_code, warning, or success) must be lowercase.

To use warning expressions:

1. Connect two job steps with a Conditional connector. In the Expression field for the conditional connector, type warning==true.

2. Navigate to the General tab of the parent job step.

3. In the Warning expression field, specify a warning code—for example, completion_code==18. This expression overrides the default warning code, if any.

   When the job is run, the system will execute the parent job step. Then the system will evaluate the condition for warning==true. If true, the system will look at the warning expression specified and determine if the condition was met. If the condition specified in the warning expression was met, the system proceeds to the next job step.
   For SAS job steps, the default completion code is completion_code==1.

Note: If a SAS job step fails because SAS execution server is not defined correctly and the default warning code of 1 is used, the job step status will still be displayed as *Success* in Job History table. This is due to the conflict between Windows warning code 1 (an executable path cannot be found) and SAS default warning code value.

SAS server. Use the drop-down list to select a SAS server to process the step. The list contains all servers currently configured to execute SAS steps. To add a new server to the list, click New. For more information, see the topic Creating New Content Server Connections in Chapter 4 on p. 16.

Remote Process Server. If SAS is not installed on the machine hosting repository, use the drop-down list to select a remote process server to process the step. The list contains all servers currently configured as remote process servers. The machine hosting SAS needs to be configured as a remote process server and defined as such in PASW Collaboration and Deployment Services. To add a new server to the list, click New. For more information, see the topic Creating New Content Server Connections in Chapter 4 on p. 16.

Additional Arguments for SAS Steps

The additional arguments list for SAS job steps corresponds to the system options that can be supplied to the executable file when invoking SAS. By default, the additional arguments list includes the two options shown in the following table.

Table 16-1
Default arguments

Option	Description
-NOSPLASH	Suppresses the splash screen when invoking the SAS execution server.
-$NOSTATUSWIN	Suppresses the status window when invoking the SAS execution server.

Additional options can be added to this list for any SAS job step. Each option must be preceded by a hyphen. See the SAS options documentation at *http://v8doc.sas.com/sashtml/lrcon/z0928597.htm* for a comprehensive list of available arguments.

Controlling Graph Output

SAS steps that create graphs without the use of **ODS** to save them with HTML output require the specification of a device driver.

Table 16-2
DEVICE argument

Option	**Description**
-DEVICE *devicename*	Defines *devicename* as the SAS/GRAPH® output device driver.

For example,

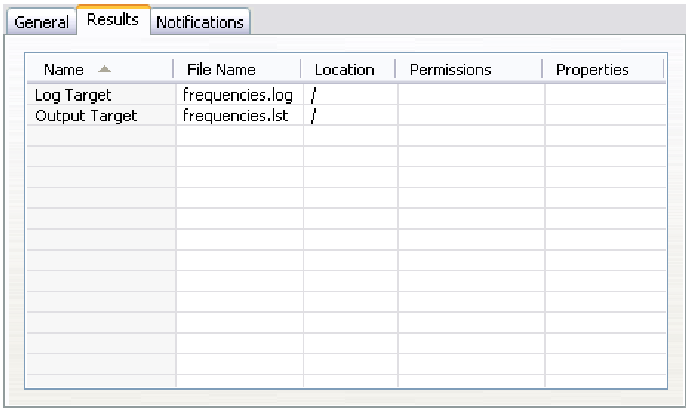
```
-DEVICE Jpeg
```

defines the JPEG driver for creating graphs. In addition to *.jpeg*, devices commonly used for exporting graphs include *.bmp* (bitmaps), *.gif* (gif files), and *.wmf* (Windows metafiles). To see a list of devices available for your system, consult the *Devices* catalog for the SAS execution server associated with the step.

Results for SAS Steps

Execution of a SAS step yields two general types of results: a processing log and an output listing file. To define properties for these results, click a SAS step in an open job. Click the Results tab to view or edit the step properties.

Figure 16-2
SAS results



Name	File Name	Location	Permissions	Properties
Log Target	frequencies.log	/		
Output Target	frequencies.lst	/		

The Name column identifies the different result types for SAS steps. Properties for each result type include:

File Name. Specify a name for the results file by clicking the cell for the desired target and typing the new name.

Location. Define the location in which to save the results by clicking the Location cell for the desired target. Click the resulting ellipsis button to specify the location. Defining result locations for SAS results is identical to doing so for PASW Statistics results.

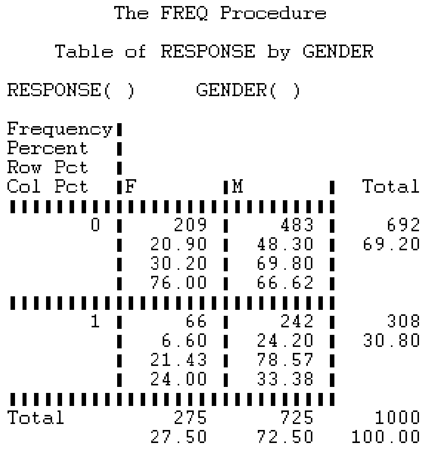
Permissions. Define access permissions for the results file by clicking the Permissions cell for the output target. Click the resulting ellipsis button to specify the permissions. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.

The contents of the output listing file correspond to the output that would be produced in the Output window by executing the step syntax directly within the SAS environment. Although the syntax may produce additional files, such as exported data files or PMML model files, you cannot define properties for those files using a SAS step. If control over the files is needed, use a General job step to execute the SAS syntax.

SAS Result Formats

Both the log and listing output are saved as text files. The listing can be saved as HTML by including **ODS** statements within the SAS program file.

Figure 16-3
Text listing output



```
          The FREQ Procedure

     Table of RESPONSE by GENDER

 RESPONSE( )     GENDER( )

 Frequency▌
 Percent  ▌
 Row Pct  ▌
 Col Pct  ▌F       ▌M       ▌  Total
 ▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌
        0 ▌    209 ▌    483 ▌    692
          ▌  20.90 ▌  48.30 ▌  69.20
          ▌  30.20 ▌  69.80 ▌
          ▌  76.00 ▌  66.62 ▌
 ▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌
        1 ▌     66 ▌    242 ▌    308
          ▌   6.60 ▌  24.20 ▌  30.80
          ▌  21.43 ▌  78.57 ▌
          ▌  24.00 ▌  33.38 ▌
 ▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌
 Total         275      725     1000
             27.50    72.50   100.00
```

When using ODS to control output format, if the results are saved to the repository, all output files generated by the step are saved to a single zip file. The name of the file corresponds to the name specified for the Output Target.

SAS/Graph® Results

If the SAS command file associated with a step does not use **ODS** for saving graphs with HTML output, the file must include the proper syntax for exporting the graph output:

1. Use FILENAME to create a *fileref* corresponding to the output directory.
2. Assign the *fileref* to the graphics stream using the GSFNAME graphic option.

   For example, the syntax

```
FILENAME grout 'c:\graph_output';
GOPTIONS GSFNAME=grout;
```

uses the *c:\graph_output* directory for the graphics stream. To use the current working directory, specify:

```
FILENAME grout '.';
GOPTIONS GSFNAME=grout;
```

The format of the graph files depends on the -DEVICE option specified as an additional argument for the step. For more information, see the topic Controlling Graph Output on p. 311.

To save the step results to the repository, use the current working directory as the *fileref* for the graphics stream. Deployment Manager stores the graph output within a single zip file in the repository. To save the graph files individually, use a General job step to process the SAS syntax.

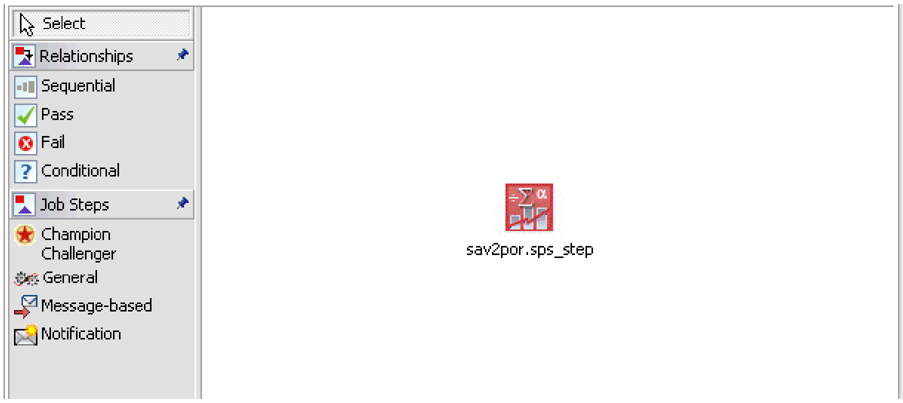
Example SAS Step

To illustrate the use of a SAS step in a job, we will use SAS to compute descriptive output for the data introduced in the PASW Statistics example. See for information about the data. To enable the SAS step to use the data, we transform it to an PASW Statistics portable file using the following PASW Statistics syntax:

```
GET FILE='C:\Program Files\data\Tutorial\sample_files\customers_model.sav'.
EXPORT OUTFILE='C:data\customers_model.por'.
```

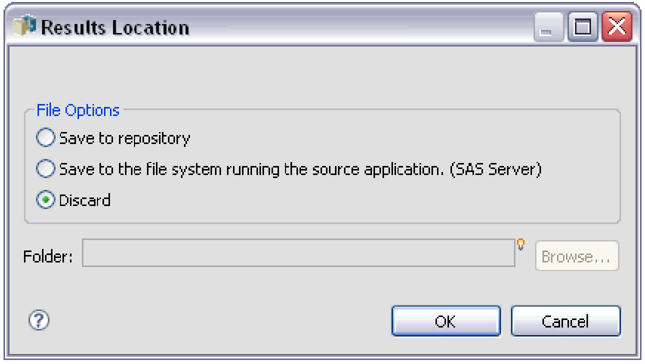
Create an PASW Statistics command file (*sav2por.sps*) containing this syntax and save it to the Content Repository. Create a new job and drag the file to the job canvas.

Figure 16-4
Job canvas with PASW Statistics step



Click the step to set its properties. On the General tab, specify the PASW Statistics execution server to process the step. On the Results tab, click the Location cell for the Output Target, click the resulting ellipsis button, and specify a location of *Discard*.

Figure 16-5



Results Location

File Options

○ Save to repository

○ Save to the file system running the source application. (SAS Server)

● Discard

Folder:

This step is designed to create a new file; the PASW Statistics output is not of interest.

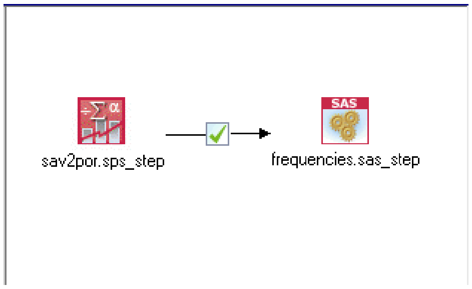
If the PASW Statistics step runs successfully, a file named *customers_model.por* will be created in the *C:\data* directory on the server. We will use the file as input to a SAS step. Consider the following SAS syntax:

```
FILENAME CUST DISK 'C:\data\customers_model.por';
LIBNAME CUST SPSS;
PROC FREQ DATA=CUST._FIRST_;
TABLES response*gender;
RUN;
```

These commands read the PASW Statistics portable file into a SAS data view and create a crosstabulation of *response* and *gender*. Create a SAS command file containing this syntax, save it under the name *frequencies.sas*, and save it to the Content Repository. Drag the file to the job canvas containing the PASW Statistics step.

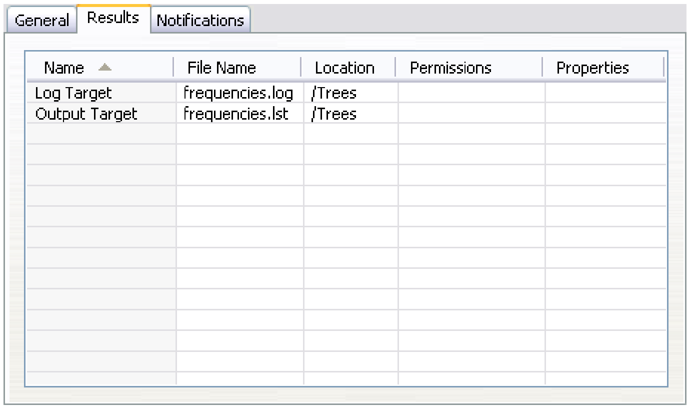
Execution of the SAS step depends on the successful completion of the PASW Statistics step. To enforce this dependency, we create a *Pass* relationship from the PASW Statistics step to the SAS step. Select the Pass tool from the job palette, click the PASW Statistics step, and click the SAS step. With this relationship in place, the *sav2por.sps* step must succeed for the *frequencies.sas* step to be invoked.

Figure 16-6
Job canvas with Pass relationship



Define properties for the SAS step by clicking it. On the General tab, select the SAS server for processing the step. On the Results tab, click the Location cell for the Output Target to define the Location. Click the ellipsis button in the cell to access the Location dialog box. For this example, we save the results to the *Trees* folder in the repository. Click OK to return to the Results tab.

Figure 16-7
Results properties for a SAS step



Name	File Name	Location	Permissions	Properties
Log Target	frequencies.log	/Trees		
Output Target	frequencies.lst	/Trees		

Save the job and click the Run Job Now button to execute it. When it completes, press the F5 key to refresh the Content Explorer. The output and log files appear in the *Trees* folder.

Suppose we wanted to include graphs in the output. Consider the modified syntax below:

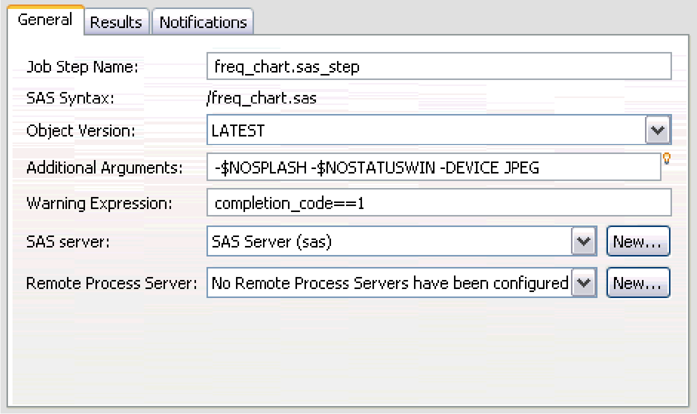
```
FILENAME CUST DISK 'C:\data\customers_model.por';
LIBNAME CUST SPSS;
PROC FREQ DATA=CUST._FIRST_;
TABLES response*gender;
RUN;

FILENAME grout '.';
GOPTIONS GSFNAME=grout;

AXIS1
   label=('Frequency')
   MINOR=NONE
   ;
AXIS2
   label=('Response')
   MINOR=NONE
   ;
TITLE;
PROC gchart DATA=CUST._FIRST_;
   vbar3d response /
   GROUP=GENDER
   DISCRETE
   RAXIS=AXIS1
   MAXIS=AXIS2
   AUTOREF
   SHAPE=Block
   PATTERNID=GROUP
   COUTLINE=BLACK
   FRAME
   ;
RUN;QUIT;
```

```
PROC gchart DATA=CUST._FIRST_;
   pie3d response /
   GROUP=GENDER
   NOLEGEND
   SLICE=OUTSIDE
   PERCENT=ARROW
   VALUE=NONE
   ACROSS=2
   COUTLINE=BLACK
   ;
RUN;QUIT;
```

These commands add bar and pie charts to the output, exporting them to the working folder as defined by the fileref *grout*. Create a new file named *freq_chart.sas* containing this new syntax and add it to the Content Repository. Delete the *frequencies.sas* file from the job canvas and add *freq_chart.sas*. Add a Pass relationship from the PASW Statistics step to the new SAS step.

The inclusion of graphs requires the addition of a system option to define the format. Click the SAS step to modify the properties and click the General tab.

Figure 16-8
General tab with DEVICE option



Job Step Name: freq_chart.sas_step
SAS Syntax: /freq_chart.sas
Object Version: LATEST
Additional Arguments: -$NOSPLASH -$NOSTATUSWIN -DEVICE JPEG
Warning Expression: completion_code==1
SAS server: SAS Server (sas)
Remote Process Server: No Remote Process Servers have been configured

Add the -DEVICE JPEG option to save the graphs in the JPG format. Save and run the job. When the job completes, press the F5 key to refresh the Content Explorer.

The *Trees* folder contains the results of the run in the file *freq_chart.zip*. The listing file contains the results of the FREQ procedure. The JPG files *gchart.jpg* and *gchart1.jpg* contain the two graphs. To save the three files individually, use a General job step to process the SAS syntax.

Chapter **17**

General Job Steps

General job steps provide a method for scheduling generic executable processes using Deployment Manager. These processes may involve executables or batch files and are often controlled using command line options or switches.

In contrast to other types of job steps, General job steps can access files stored within the repository. As a result, stored data or model files can serve as input to processes. In addition, General job steps provide control over individual output files. For example, individual graphs can be stored in the repository by these steps.

To add a General job step to a job, select the General job tool from the job palette and click on the job canvas. However, before the job can run successfully, several properties of the step must be defined. These properties can be classified into three categories:

- General properties, such as the command to run
- Input file properties, such as the location of files used as input to the command
- Output file properties, such as the location of files generated by the command

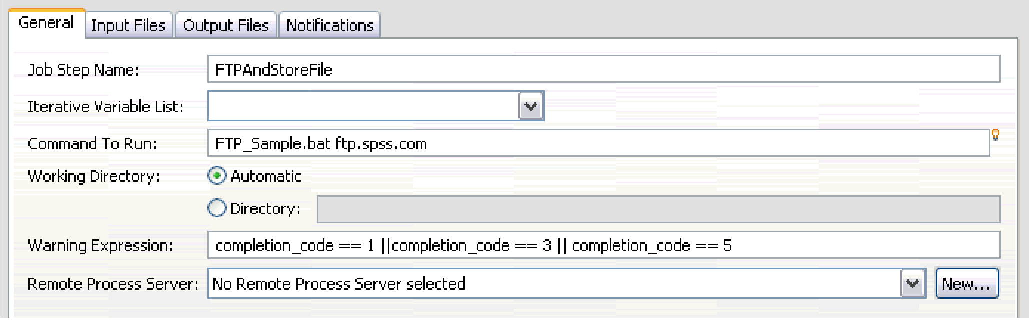
General Properties for General Job Steps

The general properties for a General job step specify the job to run and the directory in which to place intermediate files.

To define General properties for a General job step:

1. Click a General job step in an open job.

Figure 17-1
General properties for a General job step



Job Step Name: FTPAndStoreFile

Iterative Variable List:

Command To Run: FTP_Sample.bat ftp.spss.com

Working Directory: ● Automatic ○ Directory:

Warning Expression: completion_code == 1 ||completion_code == 3 || completion_code == 5

Remote Process Server: No Remote Process Server selected

2. Click the General tab to view or edit the step properties.

- **Job step name.** Type a name for the step.

- **Iterative Variable List.** If the job step is connected to an iterative producer job step, select the name of the iterative producer from the drop-down list. For more information, see the topic Type for Reporting Steps in Chapter 14 on p. 296.
- **Command to run.** Type the command to be executed by the step, including the path to the file. Paths containing spaces must be enclosed in quotes or specified using short names, such as *PROGRA~1* for *Program Files*.
- **Working Directory.** Specify the working directory the step should use when dealing with files. Select Automatic to use a custom internal directory, or select Directory to specify a particular directory.
- **Warning expression.** The warning code expression for the job step. The warning expression (e.g., completion_code, warning, or success) must be lowercase. For example, specifying the warning expression completion_code==26 instructs the system to treat the return code 26 as a success of the job step execution, although it may be different from the default success code.

Remote Process Server. If the executable providing the functionality invoked by the job step is not installed on the machine hosting repository, use the drop-down list to select a remote process server to process the step. The list contains all servers currently configured as remote process servers. The machine hosting the executable needs to be configured as a remote process server and defined as such in PASW Collaboration and Deployment Services. To add a new server to the list, click New. For more information, see the topic Creating New Content Server Connections in Chapter 4 on p. 16.

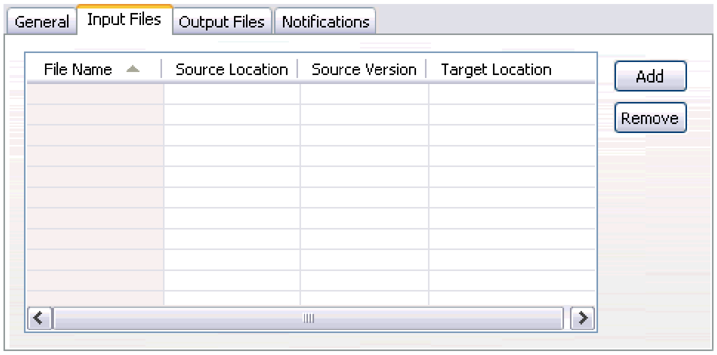
Input Files for General Job Steps

The Input Files tab identifies files used as input to the General job step. The step copies an input file from its source location to a specified target location for subsequent processing when running the command.

To define Input File properties:

1. Click a General job step in an open job.
2. Click the Input Files tab to view or edit the properties for input files.

Figure 17-2
Input Files properties for a General job step



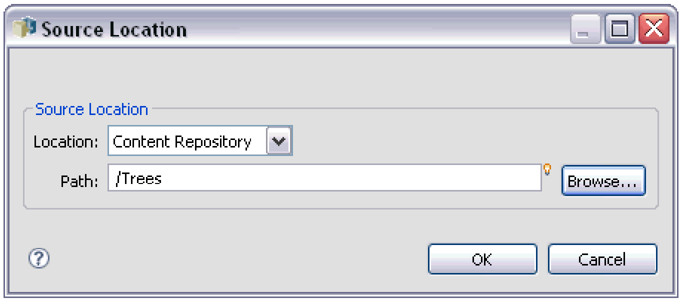
3. Click the Add button to add an input file to the step.

- **File Name.** Type the name for the input file. This name should correspond to the name of the file used when the command is run. By default, a new input file is named *inputFile*.
- **Source Location.** Specify the location of the file. Define this location by clicking the cell and clicking the resulting ellipsis button. For more information, see the topic Input File Source Location on p. 319.
- **Source Version.** For input files contained within the repository, specify the version of the file to be copied to the target location.
- **Target Location.** Specify the location to which the step should copy the file. This value should correspond to the location of the file that is accessed when the command is run. Define this location by clicking the cell and clicking the resulting ellipsis button. For more information, see the topic Input File Target Location on p. 319.

Input File Source Location

▶ Click the ellipsis button in a selected Source Location cell to define the location of an input file.

Figure 17-3
Source Location dialog box



Source Location

Source Location

Location: Content Repository

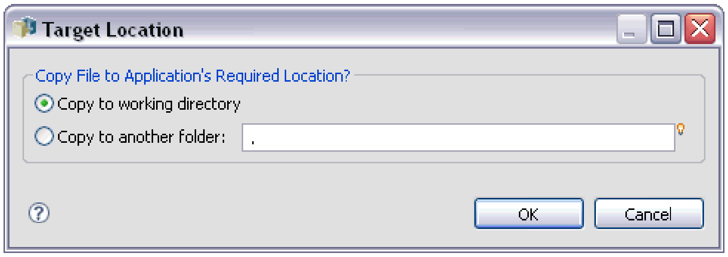
Path: /Trees

- **Location.** Define the location of the input file as either Content Repository or File System.
- **Path.** Specify the path to the input file. For input files in the Content Repository, click the Browse button to navigate to the desired path.

Input File Target Location

▶ Click the ellipsis button in a selected Target Location cell to define the location to which the input file should be copied.

Figure 17-4
Target Location dialog box



Target Location

Copy File to Application's Required Location?

● Copy to working directory

○ Copy to another folder: .

- **Copy to working directory.** Copies the input file from the source location to the working directory defined for the step. For more information, see the topic General Properties for General Job Steps on p. 317.
- **Copy to another folder.** Copies the input file to a specified folder.

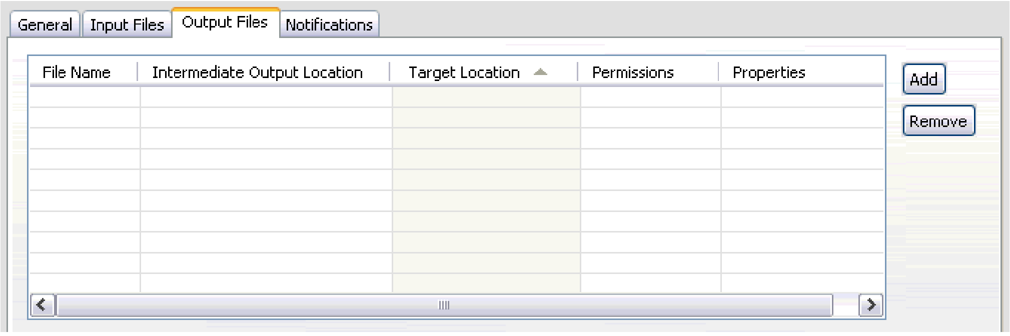
Output Files for General Job Steps

The Output Files tab identifies files generated by the General job step. When the command run completes successfully, the step copies output files from their intermediate locations to specified target locations.

To define Output Files properties:

1. Click a General job step in an open job.
2. Click the Output Files tab to view or edit the properties for output files.

Figure 17-5
Output File properties for a General job step



File Name	Intermediate Output Location	Target Location	Permissions	Properties

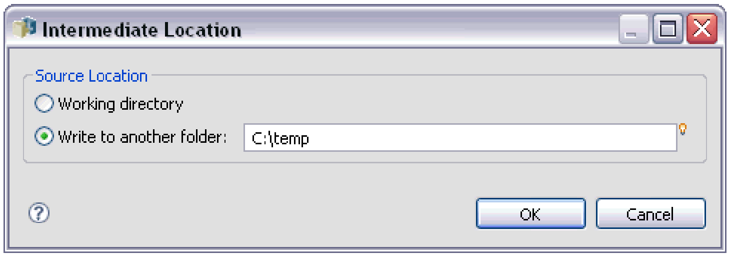
3. Click the Add button to add an output file to the step.

- **File Name.** The name of the output file. This name should correspond to the name of the file generated by running the command. By default, a newly added output file is named *outputFile*. To change the name, click in the File Name column and type the new name.
- **Intermediate Output Location.** Specify the location of the file generated by running the command. Define this location by clicking the cell and clicking the resulting ellipsis button. For more information, see the topic Input File Source Location on p. 319.
- **Target Location.** The location of the file. To define location, open the Results Location dialog box by clicking in the column and then clicking the resulting ellipsis button. For more information, see the topic Input File Target Location on p. 319.
- **Permissions.** Access permissions for the file if saved to the repository. To define permissions, open the Output Permissions dialog box by clicking in the Permissions column and then clicking the resulting ellipsis button. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.
- **Properties.** The properties (metadata) of the file. To define properties, open the Output Properties dialog box by clicking in the Properties column and then clicking the resulting ellipsis button. For more information, see the topic Output File Metadata in Chapter 10 on p. 247.

Output File Intermediate Location

▶ Click the ellipsis button in a selected Intermediate Output Location cell to define the location of the output file created by running the command.

Figure 17-6
Intermediate Location dialog box



Intermediate Location

Source Location

○ Working directory

● Write to another folder: C:\temp

The command can create an intermediate file in one of two locations:

- **Working directory.** Identifies the working directory defined for the step as the location for the intermediate file. For more information, see the topic General Properties for General Job Steps on p. 317.
- **Write to another folder.** Identifies a specific folder as the location for the intermediate file.

General Job Step Examples

Applications of the General job step are numerous and varied. The following examples illustrate some of the possible uses.

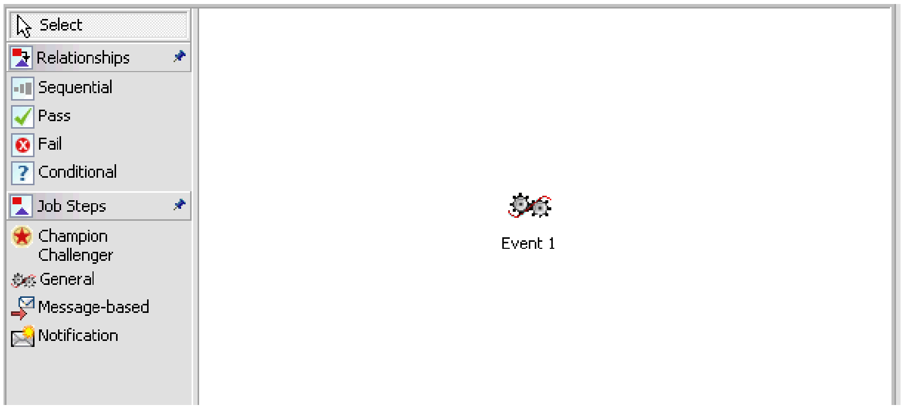
Storing SPSS Inc. PMML Files

Recall that the PASW Statistics syntax introduced in created a PMML file for the generated tree model, but that file could not be stored in the repository by the PASW Statistics step. A General job step can execute the PASW Statistics syntax and save all results to the repository.

To create a General job step for storing:

1. Open a new job.
2. Select the General job tool from the job palette.
3. Click in the job canvas.

Figure 17-7
Job canvas with General job step

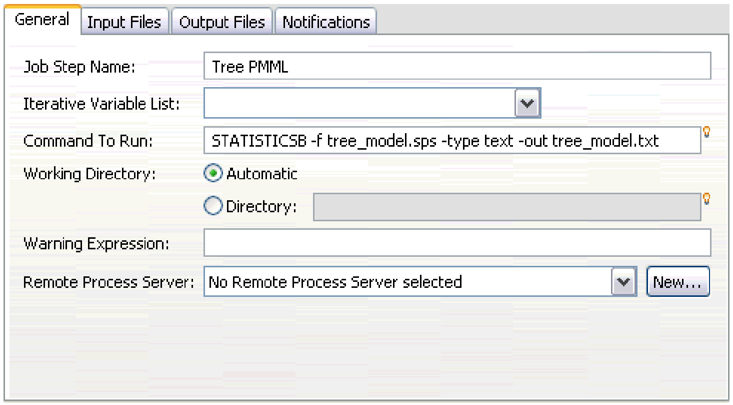


Defining General Properties

To define General properties for a General job step:

1. Click the step to open the properties dialog box.
2. On the General tab, type a name of *Tree PMML* for the job step.
3. For Command to run, type STATISTICSB -f tree_model.sps -type text -out tree_model.txt. The switches instruct STATISTICSB to process the syntax file *tree_model.sps*, saving the results as text in the file *tree_model.txt*.
4. Select Automatic for the Working Directory.

Figure 17-8
General properties for a General job step



Job Step Name: Tree PMML
Iterative Variable List:
Command To Run: STATISTICSB -f tree_model.sps -type text -out tree_model.txt
Working Directory: ● Automatic ○ Directory:
Warning Expression:
Remote Process Server: No Remote Process Server selected

Defining Input File Properties

To define Input File properties for a General job step:

1. Click the Input Files tab to define the input for the step. Running the command uses a single input, the syntax file.

2. Add an input file for the step by clicking Add. Change the name of this input file to match the name of the syntax file stored within the repository, *tree_model.sps*.

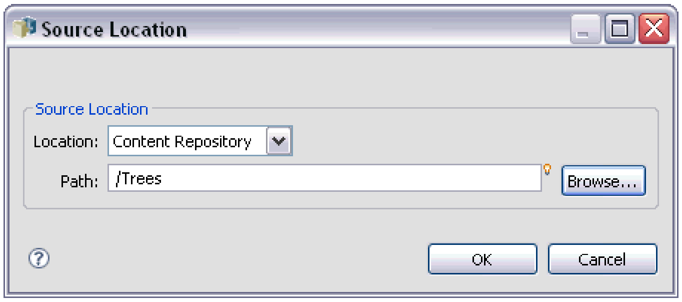
3. Click the Source Location cell and click the resulting ellipsis button to define the location of the file. The syntax file is stored within the repository, so specify Content Repository as the Location.

4. Click Browse to navigate to the *Trees* folder to define the path to the file.

5. Click OK to return to the Input Files tab.

   Note: The syntax file itself references a data file, *customers_model.sav*, stored within the file system on the server. The step does not need to relocate the file so it does not need to be listed as an input. If the data file were stored in the repository or needed to be moved to another system folder for the syntax to access it, it would be listed as an input file for the step.

Figure 17-9
Source Location dialog box

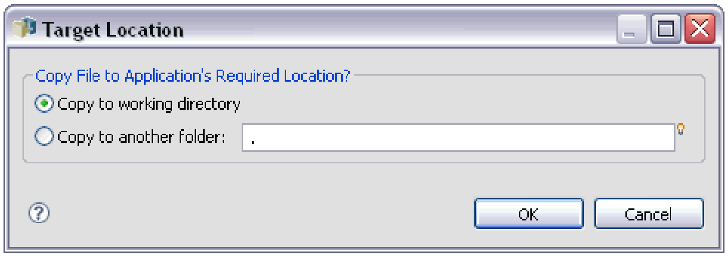


Location: Content Repository

Path: /Trees

6. Click the Target Location cell for the input file and click the resulting ellipsis button to define a location to which the step should copy the file. The switch passed to the executable does not include a path to the syntax file, so select Copy to working directory in the Target Location dialog box.

Figure 17-10
Target Location dialog box



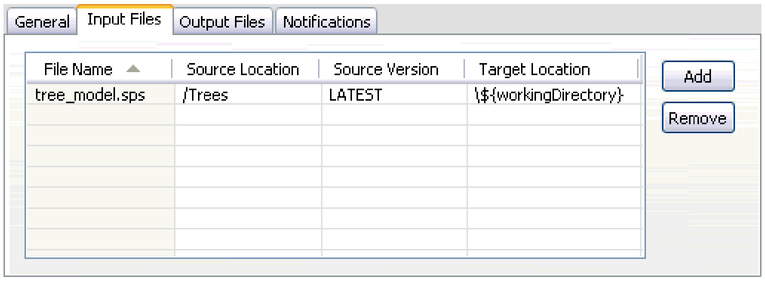
Copy File to Application's Required Location?

● Copy to working directory

○ Copy to another folder: .

7. Click OK to return to the Input Files tab. The resulting Input File properties appear below.

Figure 17-11
Input Files tab



File Name	Source Location	Source Version	Target Location
tree_model.sps	/Trees	LATEST	\${workingDirectory}

Defining Output File Properties

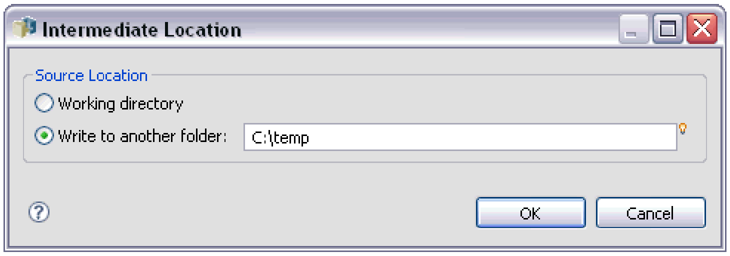
To define Output File properties for a General job step:

1. Click the Output File tab to define the output for the step. Running the command creates a text output file, but we are currently not interested in that file. Our focus is on the PMML file generated by the syntax.

2. Add an output file for the step by clicking Add. Change the name of this output file to match the name of the file created by the syntax, *tree_model.xml*.

3. Click the Intermediate Output Location cell and click the resulting ellipsis button to define the location of the file. The syntax file creates the PMML file in the *C:\Program Files\SPSS* folder, so select Write to another folder in the Intermediate Location dialog box and type the path *C:\temp*.

Figure 17-12
Intermediate Location dialog box



Source Location

○ Working directory

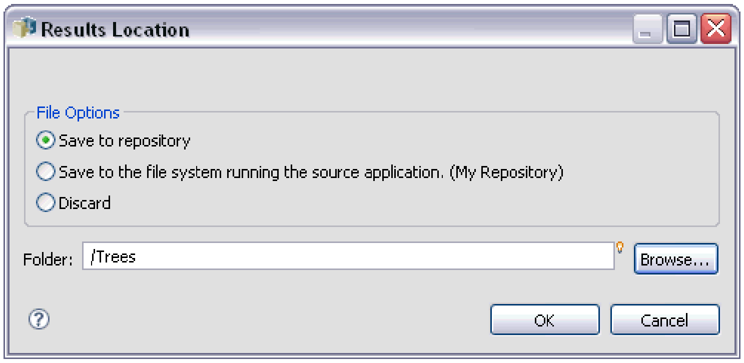
● Write to another folder: C:\temp

4. Click OK to return to the Output File tab.

5. Click the Target Location cell and click the resulting ellipsis button to define the desired location for the output file.

6. Select Save to repository in the Results Location dialog box and click Browse to specify the *Trees* folder.

Figure 17-13
Results Location dialog box



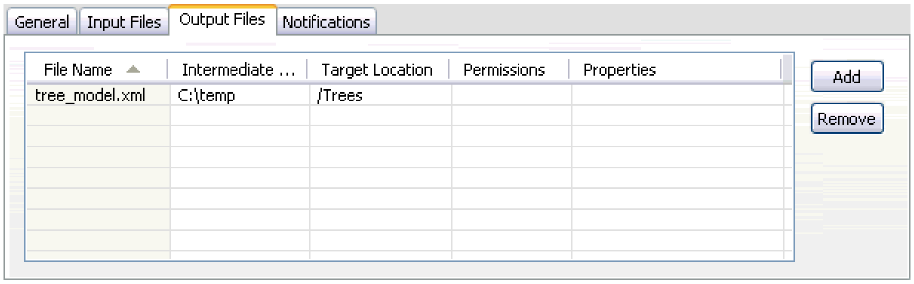
File Options

● Save to repository
○ Save to the file system running the source application. (My Repository)
○ Discard

Folder: /Trees

7. Click OK to return to the Output File tab. The resulting Output File properties appear below.

Figure 17-14
Output File tab



File Name	Intermediate ...	Target Location	Permissions	Properties
tree_model.xml	C:\temp	/Trees		

8. Save the job and click the Run Job Now button to execute the job. Deployment Manager runs the STATISTICSB command and stores the PMML file in the repository. Press the F5 key to refresh the Content Explorer and view the XML file containing the model.

Batch Scoring using PASW Statistics PMML Files

Once a predictive model has been built and the model specifications have been saved as an XML file, the model can be used to score data. Consider the following syntax:

```
GET FILE='C:\data\customers_new.sav'.
RECODE Age
   ( MISSING = COPY )
   ( LO THRU 37 =1 )
   ( LO THRU 43 =2 )
   ( LO THRU 49 =3 )
   ( LO THRU HI = 4 ) INTO Age_Group.

IF MISSING(Age) Age_Group = -9.
COMPUTE Log_Amount = ln(Amount).

MODEL HANDLE NAME=tree FILE='C:\models\tree_model.xml'
   /MAP VARIABLES=Income_Grp MODELVARIABLES=Income_Group.
```

```
COMPUTE PredCatTree = ApplyModel(tree,'predict').

SAVE OUTFILE='C:\scores\scoring_results.sav'.
```

This syntax uses the PMML file *tree_model.xml* to generate scores for the data in *customers_new.sav*. The results are saved to *scoring_results.sav*. Save the syntax into a file named *scoring.sps* and add it to a folder named *Trees* in the Content Repository.

To create a General job step for scoring:

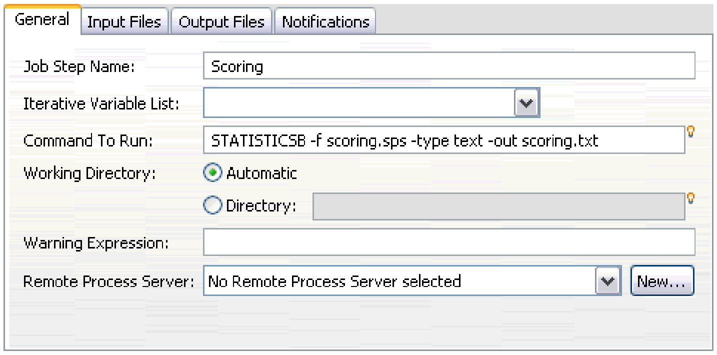
1. Open a new job.
2. Select the General job tool from the job palette
3. Click in the job canvas.

Defining General Properties

To define General properties for a General job step:

1. Click the step to open the properties dialog box.
2. On the General tab, type a name of *Scoring* for the job step.
3. For Command to run, type STATISTICSB -f scoring.sps -type text -out scoring.txt. The switches instruct STATISTICSB to process the syntax file *scoring.sps*, saving the results as text in the file *scoring.txt*.
4. Select Automatic for the Working Directory.

Figure 17-15
General properties for a General job step



Job Step Name: Scoring

Iterative Variable List:

Command To Run: STATISTICSB -f scoring.sps -type text -out scoring.txt

Working Directory: ● Automatic

○ Directory:

Warning Expression:

Remote Process Server: No Remote Process Server selected

Defining Input File Properties

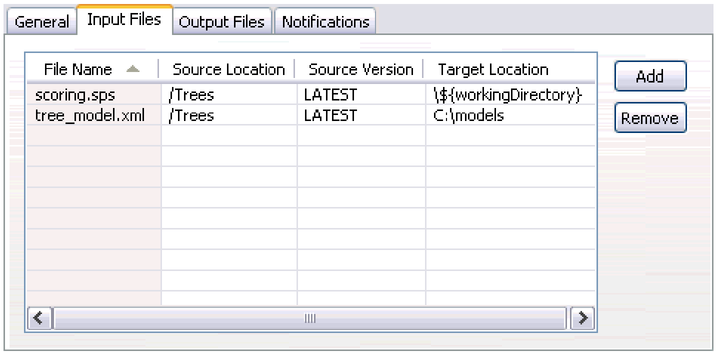
To define Input File properties for a General job step:

1. Click the Input Files tab to define the input for the step. This step requires two input files:
   - Syntax file
   - PMML file

Both files are stored in the *Trees* folder of the Content Repository. To add the files, click Add to add an input file for the syntax. Change the name of the input file to match the name of the syntax file stored within the repository, *scoring.sps*.

2. Click the Source Location cell and click the resulting ellipsis button to define the location of the file. The syntax file is stored within the repository, so specify Content Repository as the Location.
3. Click Browse to navigate to the *Trees* folder to define the path to the file. Click OK to return to the Input Files tab.
4. Click the Target Location cell and click the resulting ellipsis button to define a location to which the step should copy the file. The switch passed to the executable does not include a path to the syntax file, so select Copy to working directory. Click OK to return to the Input Files tab.
5. Click Add to add an input file for the PMML file. Change the name of the input file to match the name of the file stored within the repository, *tree_model.xml*.
6. Click the Source Location cell and click the resulting ellipsis button to define the location of the file. The syntax file is stored within the repository, so specify Content Repository as the Location.
7. Click Browse to navigate to the *Trees* folder to define the path to the file. Click OK to return to the Input Files tab
8. Click the Target Location cell and click the resulting ellipsis button to define a location to which the step should copy the file. The syntax requires the model file to be in the *C:\models* folder, so select Copy to another folder. Type the path *C:\models* and click OK.
9. Click OK to return to the Input Files tab. The resulting Input File properties appear below.

Figure 17-16
Input Files tab



File Name	Source Location	Source Version	Target Location
scoring.sps	/Trees	LATEST	\${workingDirectory}
tree_model.xml	/Trees	LATEST	C:\models

Defining Output File Properties

To define Output File properties for a General job step:

1. Click the Output File tab to define the output for the step. Running the command creates a text output file and the syntax creates a data file containing the scores.
2. Add an output file for the step by clicking Add. Change the name of this output file to match the name of the output file, *scores.txt*.

3. Click the Intermediate Output Location cell and click the resulting ellipsis button to define the location of the file. The command creates the file in the working directory, so select Working directory and click OK to return to the Output File tab.

4. Click the Target Location cell and click the resulting ellipsis button to define the desired location for the output file.

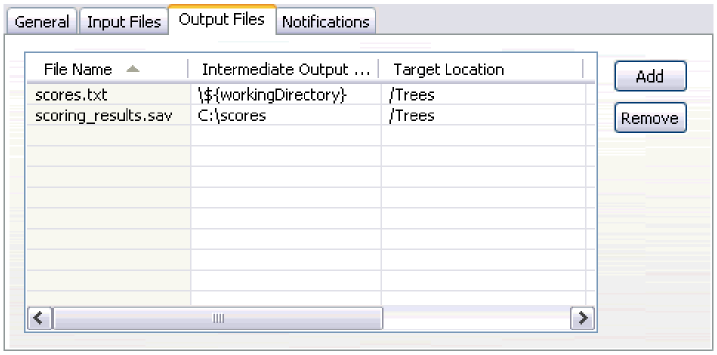
5. Select Save to repository and click Browse to specify the *Trees* folder. Click OK to return to the Output File tab.

6. Running the command creates a text output file and the syntax creates a data file containing the scores. To add the files to the Content Repository, add another output file for the step by clicking Add. Change the name of this output file to match the name of the data file containing the scores, *scoring_results.sav*.

7. Click the Intermediate Output Location cell and click the resulting ellipsis button to define the location of the file. The command creates the file in the *C:\scores* folder, so select Write to another folder and type the path *C:\scores*. Click OK to return to the Output File tab.

8. Click the Target Location cell and click the resulting ellipsis button to define the desired location for the output file. Select Save to repository and click Browse to specify the *Trees* folder. Click OK to return to the Output File tab. The resulting Output File properties appear below.

Figure 17-17
Output File tab



File Name	Intermediate Output ...	Target Location
scores.txt	\${workingDirectory}	/Trees
scoring_results.sav	C:\scores	/Trees

9. Save the job and click the Run Job Now button to execute the job. Deployment Manager runs the STATISTICSB command and scores the new data, storing the results in the repository. Press the F5 key to refresh the Content Explorer and view the file containing the scores.

Event-based Scheduling

Typically, jobs are scheduled for execution based on a particular time-based recurrence pattern—for example, weekly or monthly. In addition, you can also create event-based (or triggered) steps within a job. In this case, your job would contain an event that must be completed successfully before the next step begins. Essentially, an event-based step is a special type of conditional step.

To use an event-based trigger, you create a job step that monitors the existence or the contents of a file. If the file exists or its contents indicate that a new external event has occurred, the job step executes successfully. The completed job step then passes control to the next step in the job. For example, the first step in your job could be a shell script or a batch file that checks to see if a particular file has been updated. If the file was updated, the step completes successfully. The system then proceeds to the next step in the job.

If you use a file in your event-based step, you may need to know the name of the file and specify the file name as a parameter in the script. For example, you may want to use the event name as the file name.

If the event-based step involves a time component, such as a polling interval, it is important to ensure that the polling interval does not conflict with the scheduling recurrence pattern. For example, if your polling interval is one minute and you have scheduled the job to run every hour, then you need to ensure that the monitoring step is complete before the job is scheduled to run—that is, the event-based step has to either succeed or fail before the next job is executed. In this instance, it would be very useful to set up a time-out value in the monitoring script that is less than the scheduling recurrence pattern—for example, a time-out value of 50 minutes.

Event-based Scheduling Process Overview

Although the individual parameters may vary, the process for implementing an event-based step consists of the following tasks:

1. Create a script that monitors the contents of a file. For more information, see the topicSample Event-based Scheduling Script on p. 329.

2. Create a new job. For more information, see the topicCreating New Jobs in Chapter 10 on p. 230.

3. Add a General job step to the job that executes the script. You may also need to specify parameters for the script—for example, file name and polling interval. This is the antecedent step. For more information, see the topicGeneral Job Steps on p. 317.

4. Add another step to the job that depends on the event-based step. This is the consequent step. For more information, see the topicAdding Steps to a Job in Chapter 10 on p. 236.

5. Connect the antecedent step to the consequent step with a pass connector. For more information, see the topicPass Connector in Chapter 10 on p. 238.

6. Schedule the job for execution.

Sample Event-based Scheduling Script

The following is a sample shell script that monitors the contents of a predefined file. This script:

- Assumes that a separate process writes a timestamp to the event file when the given event is completed
- Uses this timestamp to determine whether or not the event has been processed
- Returns an exit code of 0 by default, if the script completes successfully

Script Arguments

The script takes the following command line arguments, which you can specify in the Deployment Manager when you create the General job step:

- **CONTROL_FILE**. The script monitors this file for changes. An external process writes events to this file. Periodically, the script checks the file for new events and processes them when found. After a new event has been processed, the script exits successfully. If the script exits successfully, the General job job proceeds to the next step in the job.
- **SLEEP_TIME**. This argument refers to the amount of time, in seconds, that the script sleeps between checks of the *CONTROL_FILE*.
- **TIMEOUT_MINS**. This argument controls the amount of time for which the script should run. If an event has not been processed after the number of minutes specified, script execution stops, and the script returns an exit code of 1. In this example, an exit code of 1 causes the job to fail because the event-based step was not successful. As a result, the job cannot proceed with the remaining steps.

Script Execution

The script executes the following tasks:

1. The script checks the last row of the *CONTROL_FILE* file for an event.
2. If the *RESULT_FILE* exists, the last event has already been processed. The script waits for the designated SLEEP_TIME and then returns to the *CONTROL_FILE* and repeats the process.
3. If the *RESULT_FILE* does not exist, then the script creates a new *RESULT_FILE* for the last event. The script considers the creation of the new *RESULT_FILE* a success and returns an exit code of 0. The General job job then proceeds to the next step in the job.

```
#!/bin/bash
#——————————————
#
# Copyright (c) 2005 SPSS, Inc.
# All Rights Reserved
#
# This software is the confidential and proprietary information of
# SPSS, Inc. ("Confidential Information"). You shall not disclose
# such Confidential Information and shall use it only in accordance
# with the terms of the license agreement you entered into with SPSS.



#
#
#——————————————


# Control file is written to by an external process.
# This is the file we are monitoring.
CONTROL_FILE=$1
```

```
# The duration in seconds we should sleep between
# checks on the control file
SLEEP_TIME=$2

# The timeout (in minutes)
TIMEOUT_MINS=$3

#————————————————
# usage: Print usage information then exit
#————————————————
function usage
{
  echo "usage watch.sh <control file> <sleep duration> <timeout in minutes>"
  exit -1
}

#————————————————
# check_args: Validates the command line arguments
#————————————————
function check_args
{
  if [ -z $CONTROL_FILE ]; then
    usage;
  elif [ -z $SLEEP_TIME ] ; then
    usage;
  elif [ -z $TIMEOUT_MINS ] ; then
    usage
  fi
}


#————————————————
# check_file: Watches $CONTROL_FILE for new events.  If a new event occurs,
# touch an event file.  If not, sleep for $SLEEP_TIME seconds and try again.
#————————————————
function check_file
{
    if [ -f $CONTROL_FILE ] ; then
      # Get the last event from the file
      LAST_EVENT=`tail -1 $CONTROL_FILE`;
      echo "Last event: $LAST_EVENT"
    else
      "$CONTROL_FILE does not exist.  Exiting."
      exit 1
    fi

    # Check to see if we've processed this event
    RESULT_FILE=${0}.$LAST_EVENT;
    echo "Checking for event $LAST_EVENT with file $RESULT_FILE";

    if [ -e $RESULT_FILE ] ; then
```

```
        # We've  processed it, sleep and
        # check again later for a new event
        echo "Sleeping for $SLEEP_TIME seconds"
        sleep $SLEEP_TIME;
    else
        # We haven't processed the latest event
        echo "A new event $LAST_EVENT has been fired.  Processing.";
        touch $RESULT_FILE;
        FINISHED=finished;
    fi
}

check_args;


# How long we should run before failing
DURATION=$[$TIMEOUT_MINS*60]

# Get the current time and add our duration to it
NOW=`date +%s`
END_TIME=$[$DURATION + $NOW]

# Finished flag - when set to "finished" we exit
FINISHED=not

echo "Running for a maximum of $DURATION seconds"

while [ $FINISHED != "finished" ] ; do
  check_file;
  NOW=`date +%s`
  if [ $NOW -gt $END_TIME ] ; then
    "Timed out after $DURATION seconds"
    exit 1
  fi
done

# If we've got this far we were successful
# so just return a success code
exit 0
```

Script Failure

Common reasons for a script failure are:

- A new *RESULT_FILE* is not created before the script times out.
- The *CONTROL_FILE* cannot be found.
- Improper specification of command line arguments.

If the script fails, it returns an exit code of 1. In this example, this failure will cause the job to fail because the next step in the process cannot be executed.

Chapter
18

Message-Based Job Steps

Message-Based job steps provide a method for triggering processing within an PASW Collaboration and Deployment Services job with external events signalled by JMS (Java Messaging Service) messages. Message-based job steps allow to specify message text and message selector as well as the timeout (a time to wait for the message). After the specified message is received, the system proceeds to the next step in the job. If the timeout is exceeded, the job is marked as failed.

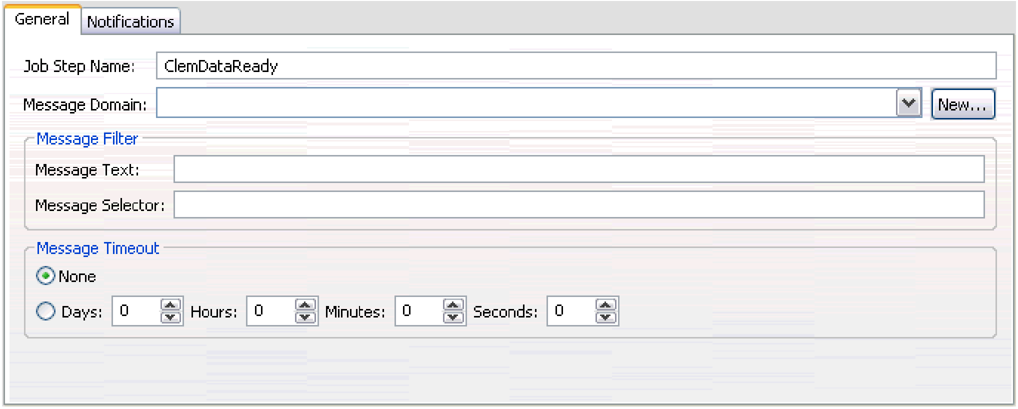
To add a message-based step to a job, select the Message-driven Work tool from the job palette and click on the job canvas. Before the job step can run successfully, its properties must be defined.

General Properties of Message-Based Job Steps

The general properties for a message-based job step specify the message domain, message text, message selector, and the step timeout.

To define the properties for a message-based job step, click a message-driven step in an open job. Click the General tab to view or edit the step properties.

Figure 18-1
General tab properties for a message-driven job step



Job Step Name: ClemDataReady

Message Domain:

Message Filter

Message Text:

Message Selector:

Message Timeout

● None

○ Days: 0 Hours: 0 Minutes: 0 Seconds: 0

1. The properties include:

- **Job step name.** Type a name for the step.
- **Message Domain.** The message domain identifies the JMS topic to listen on. For more information, see the topic Message Domains in Chapter 6 on p. 94.
- **Message Text.** For text message, the text of the expected message of to trigger the job step execution.

- **Message Selector** The selector text to filter messages by contents of the message header.
- **Message Timeout.** The time (in days, hours, minutes, and seconds) to wait for the message triggering the job step. The timeout is effective from the start of the job step after predecessor steps have been completed.

Chapter

19

Notification Job Steps

E-mail notifications can be set up in Deployment Manager using either of the following methods:

- Notifications tab. For more information, see the topic Job Success and Failure Notifications in Chapter 13 on p. 278.
- Notification job step.

A notification job step is a specialized method for sending notifications. Typically, the notification job step is used in conjunction with model evaluation.

Notifications can be set up to monitor specific model evaluation outcomes. The following are examples of when notifications might be based on model evaluation return codes:

Notifications can be set up based on a specific outcome of the model evaluation. For example, if the result returned by a model evaluation analysis is less than 0.85, notifications could be sent to a predefined list of recipients.

Notifications can be set up based on the traffic light level of a model evaluation job. For example, if the result is red, a notification could be sent. In addition, multiple notification rules could be set up with different e-mail recipients for each traffic light level.

Using a notification job step is similar to specifying notifications as part of another job step. However, the following key differences apply:

- Using notification job steps, different notification templates can be selected for different outcomes. In addition, notification rules can be established for multiple sets of notification recipients.
- In notification job steps, more specific criteria can be established for sending notifications, using conditional connectors. Notifications sent via the Notifications tab apply only for success or failure.
- With notification job steps, all e-mail messages are sent as links. The ability to send an e-mail message as an attachment is not available.

Adding a Notification Job Step to a Job

Typically, a notification job step is used in conjunction with another job step. To add a notification job step to a job:

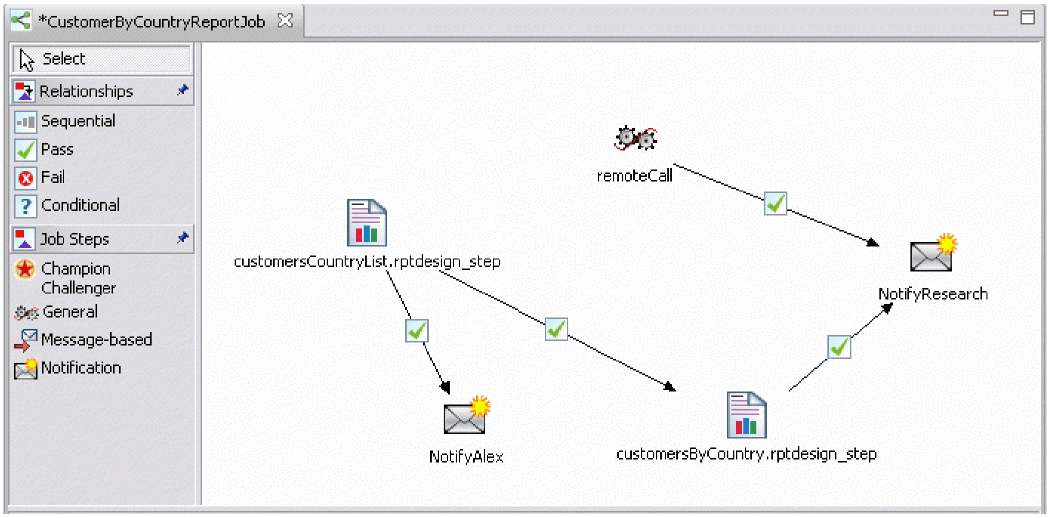
1. Create a new job or open an existing job.

2. From the job palette, click Notification.

3. Click anywhere in the job canvas. The notification job step is added to the job.

4. Because a notification job step is typically connected to another job step, at this stage relationships would be established between the notification job step and other steps within the job. For more information, see the topicSpecifying Relationships in a Job in Chapter 10 on p. 237.

Figure 19-1
notification job step general information

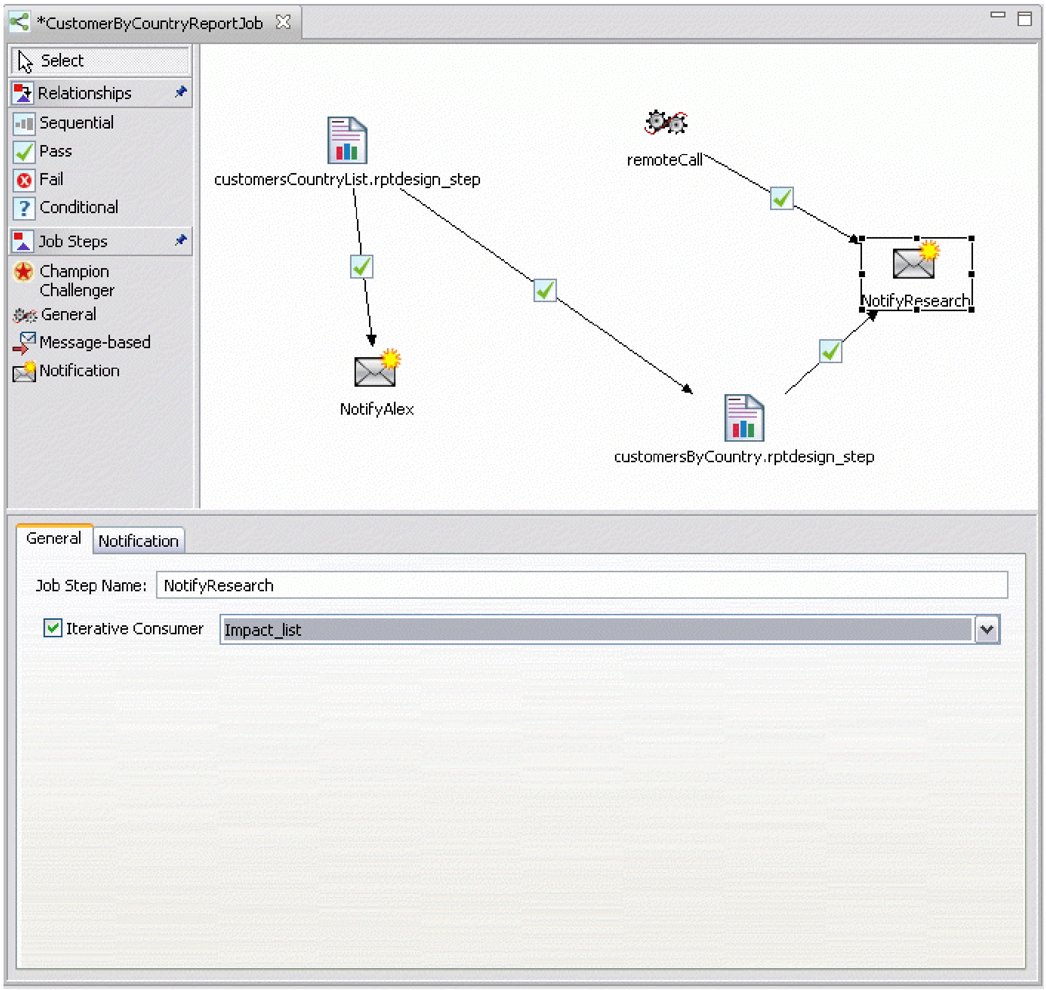


After the notification job step has been added to the job, its general information and notification recipients must be specified on the corresponding tabs.

General Information

The General tab contains information that pertains to the overall notification job step.

Figure 19-2
General tab of a notification job step



Job Step Name: NotifyResearch

☑ Iterative Consumer Impact_list

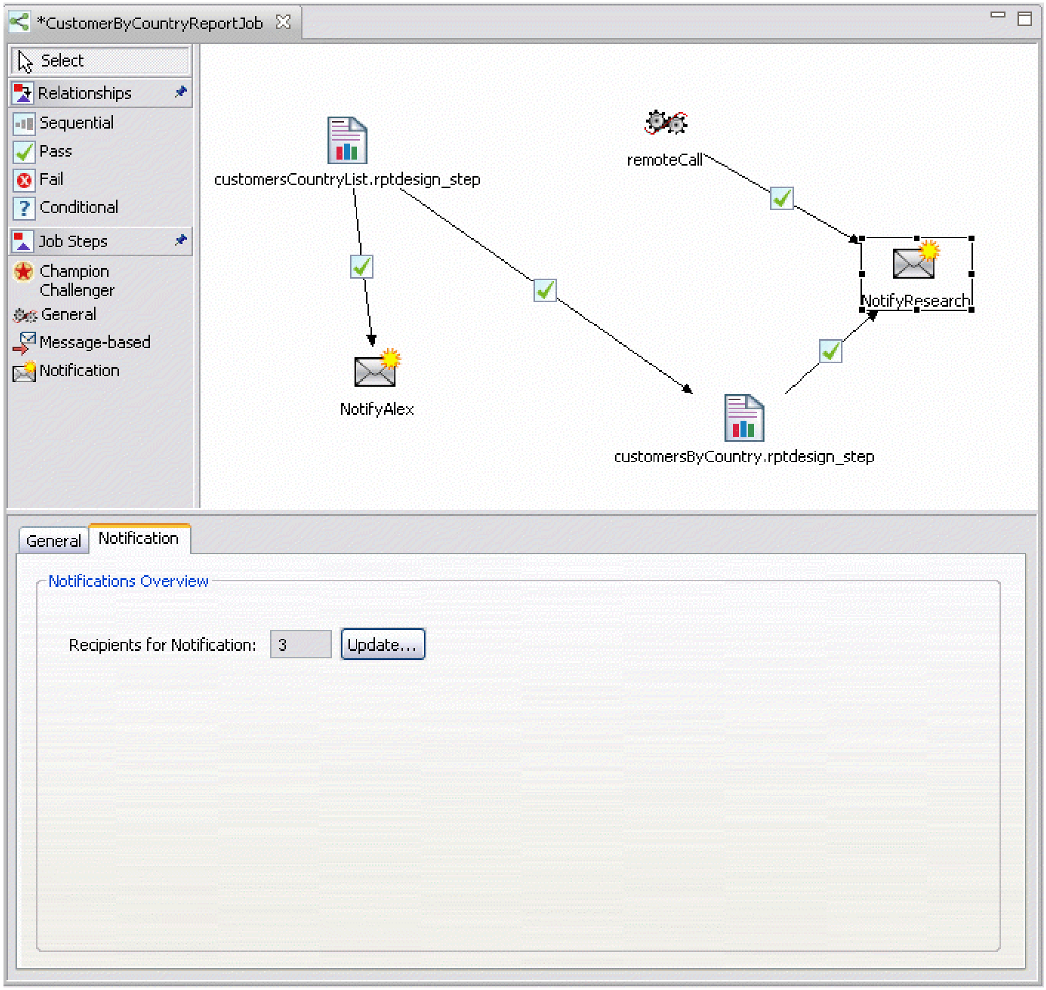
The General tab contains the following information:

- **Job step name.** Type a name for the step.
- **Iterative consumer.** Specifies whether or not the notification job step is an iterative consumer of another job step. Select the Iterative Consumer check box if this job step is an iterative consumer of another job step. If this check box is selected, one of the available iterative variables must be selected from the drop-down list.

Notification

The Notification tab is used to configure the recipients of notifications for the job step.

Figure 19-3
Notification tab of a notification job step



Recipients for Notification: 3

The tab contains the following information:

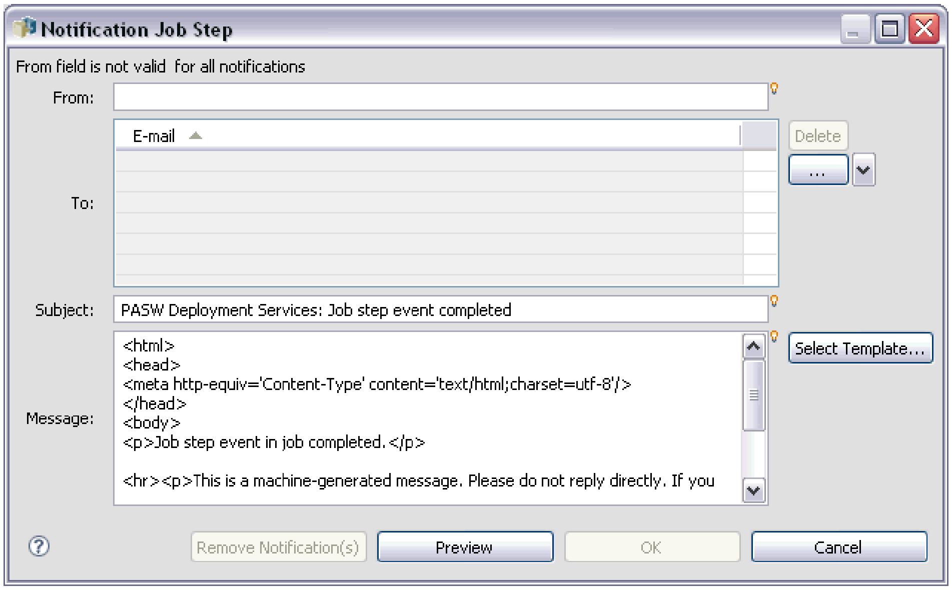
- **Recipients for notifications.** The number of recipients that receive notifications. This number is generated by the system based on the number of recipients specified in the Notifications dialog.

Updating Notifications

To update notification parameters:

1. In the General tab, click Update. The Notifications dialog appears.

Figure 19-4
Notification Job Step dialog



From field is not valid for all notifications

Subject: PASW Deployment Services: Job step event completed

Message:

```
<html>
<head>
<meta http-equiv='Content-Type' content='text/html;charset=utf-8'/>
</head>
<body>
<p>Job step event in job completed.</p>

<hr><p>This is a machine-generated message. Please do not reply directly. If you
```

2. In this dialog, the following parameters can be modified:

   - **From.** The address of the sender notification message. The field is prepopulated with the default e-mail address in repository configuration options.
   - **To.** The list of notification recipients. The addresses can be entered, specified as an iterative variable for iterative job step notifications, specified as a dynamic list based on a query, or selected from a directory listing of a supported e-mail application, such as Microsoft Outlook or Lotus Notes. To edit an address, click the ellipsis button next to it in the list. Recipients can also be specified as variables. To remove a recipient, highlight the recipient's e-mail address and click Delete.
   - **Subject.** The subject of the notification message. By default, the field is populated by the default template associated with the notification event. Modify the message subject, if necessary. For more information, see the topic Customizing the Notification Message in Chapter 13 on p. 288.
   - **Message.** The body text of the notification message. The field is populated by the default template associated with the notification event. Modify the message body text, if necessary. For more information, see the topic Customizing the Notification Message in Chapter 13 on p. 288.

3. Preview the notification message, if desired. For more information, see the topicNotification Message Preview in Chapter 13 on p. 288.

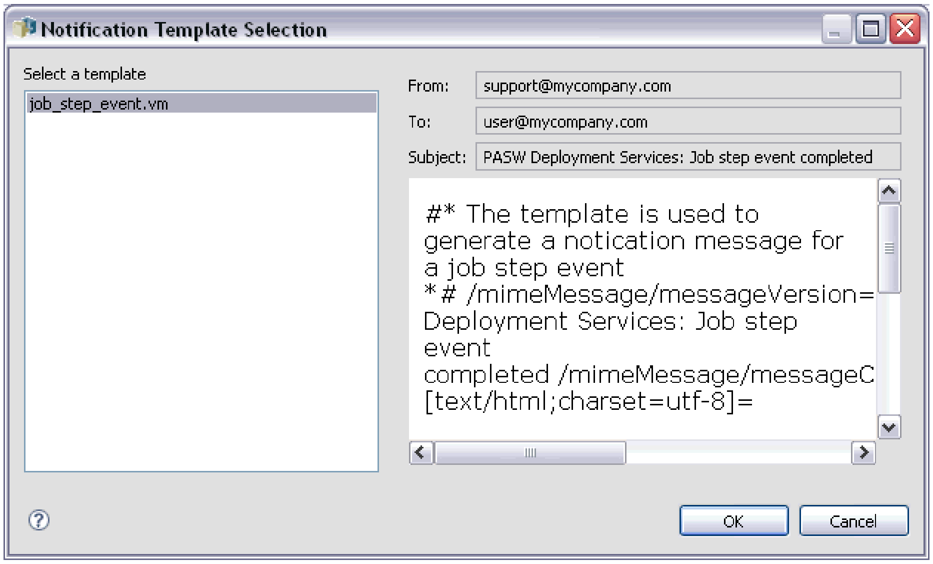
4. Click OK.

Selecting a New Template

When a notification job step is initially created, the template fields are blank, and a template needs to be selected from the list of available templates. The template selected applies to the current notification job step only and persists until a new template is selected. In this dialog, templates can only be modified. New templates cannot be created. To select a new default template:

1. In the Notifications dialog, click Select Template. The Notification Template Selection dialog appears.

Figure 19-5
notification job step template HTML



job_step_event.vm

From: support@mycompany.com

To: user@mycompany.com

Subject: PASW Deployment Services: Job step event completed

```
#* The template is used to
generate a notication message for
a job step event
*# /mimeMessage/messageVersion=
Deployment Services: Job step
event
completed /mimeMessage/messageC
[text/html;charset=utf-8]=
```

2. Select a template from the list. A preview of the template appears in the Notification Template Selection dialog. The template can only be viewed in this dialog. Modifications can only be made in the main Notifications dialog.

3. Click OK. The Notifications dialog reappears.

4. Make any modifications necessary to the notification message. For more information, see the topicUpdating Notifications on p. 338.

5. Click OK.

Chapter

20

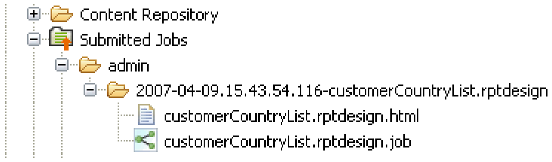
Submitted Jobs

The *Submitted Jobs* folder is a staging area in repository that displays the results of running reports using Deployment Portal. From the reports, jobs and additional output are generated and appear in the *Submitted Jobs* folder. For example, *Business Intelligence and Reporting Tools (BIRT)* reports can be submitted using Deployment Portal, and their associated jobs and output appear in *Submitted Jobs*. For details about BIRT, see *http://www.eclipse.org/birt*.

The *Submitted Jobs* folder is at the same level as the *Content Repository* in the Content Explorer. The *Submitted Jobs* folder appears in the Content Explorer, whether or not content is available within it.

When a user submits a report to PASW Collaboration and Deployment Services, a folder is created for the user within the *Submitted Jobs* folder. The folder name corresponds to the user's login ID. Subsequently, output is placed into the folder. Each report is contained in a separate folder, whose name contains a timestamp. Each report folder contains a corresponding job and any additional artifacts related to the report.

Figure 20-1
Submitted jobs folder example


Content Repository
Submitted Jobs
admin
2007-04-09.15.43.54.116-customerCountryList.rptdesign
customerCountryList.rptdesign.html
customerCountryList.rptdesign.job

The following guidelines apply to objects in the *Submitted Jobs* folder:

- The objects within the *Submitted Jobs* folder are saved to the repository.
- The contents of the *Submitted Jobs* folder are searchable. For more information, see the topic Searching in Chapter 4 on p. 22.
- Objects, such as files and jobs, can be opened but only in read-only mode.

For more information about reporting, see Reporting Job Steps on p. 294.

Restrictions within the Submitted Jobs Folder

Objects in the *Submitted Jobs* folder cannot be modified—for example, names cannot be changed, jobs cannot be scheduled, and versioning is disabled. To modify an object, it must be moved from the *Submitted Jobs* folder to the *Content Repository*. Once an object is moved into the *Content Repository*, full functionality is restored.

Although objects can be moved from the *Submitted Jobs* folder to the *Content Repository*, the reverse is not possible. Objects from the *Content Repository* cannot be moved into the *Submitted Jobs* folder.

The only exception to these restrictions is permissions. Permissions can be changed for objects within the *Submitted Jobs* folder. All of the standard permissions are available; however, only the owner of an object can modify its permissions. For example, the owner can grant permissions to modify an object. By default the user who created the report is the owner. For more information, see the topic Modifying Permissions in Chapter 5 on p. 41.

Submitted Jobs and Expiration Dates

By default, objects within the *Submitted Jobs* folder automatically expire after five days. To prevent an object from expiring, it needs to be moved out of the *Submitted Jobs* folder and into the *Content Repository*. Once the object is moved into the *Content Repository*, the five-day limit is removed, and no expiration date is attached to the object. After the object has been moved into the *Content Repository*, an expiration date must be explicitly applied, if desired.

It is important to note that the five-day time period in the *Submitted Jobs* folder is different from setting a specific expiration date in the *Content Repository*. Unlike expired objects in the *Content Repository*, expired objects in the *Submitted Jobs* folder are automatically deleted from the repository. In addition, the five-day default in the *Submitted Jobs* folder is a configuration setting, which can be modified by an administrator. Expiration dates in the *Content Repository* are user-defined and modifiable. For more information, see the topic Working with Expiration Dates and Expired Files in Chapter 5 on p. 50.

Chapter

21

Accessibility

Deployment Manager offers accessibility for all users, as well as specific support for users with visual and other functional impairments. This section describes the features and methods of working in Deployment Manager using accessibility enhancements, such as screen readers and keyboard shortcuts.

Types of Accessibility Support

Whether you have a visual impairment or are dependent upon the keyboard for manipulation, there is a wide variety of alternative methods for using this product. Available keyboard shortcuts are listed in the topics that follow. Additionally, Deployment Manager provides support for screen readers, such as JAWS for Windows. You can also optimize the color scheme used in Deployment Manager to provide additional contrast.

JAWS and Java

Different versions of JAWS provide varying levels of support for Java-based software applications. Although Deployment Manager may work with prior or later versions of JAWS, version 10.0.1154 is recommended. Visit the JAWS for Windows Web site at *http://www.FreedomScientific.com*.

Keyboard Accessibility

Deployment Manager offers keyboard shortcuts to make the product's functionality more accessible. At the most basic level, you can press the Alt key plus the appropriate key to activate window menus (for example, Alt + F to access the File menu) or press the Tab key to scroll through dialog box controls.

General Shortcuts

The following table describes general shortcuts that may be used throughout Deployment Manager.

Table 21-1
General keyboard shortcuts

Shortcut	Action
Shift + F10	Brings up a context menu, similar to a right-click.
Ctrl + F7	Brings up a list of active views and allows you to select one.
Tab	Iterates through the different controls on the view (tables, toolbars, etc.)

Shortcut	Action
Ctrl + M	■ Maximizes the active view ■ Returns maximized view to original size
Alt	Highlights mnemonics—the menu titles in the menu bar have their mnemonic letters underlined.
Alt + mnemonic	Selects the control (menu item, button, check box, text field, etc.) that is mapped to that mnemonic. Examples include: ■ Alt + F opens the *File* menu. ■ Alt + N in the Roles editor brings up the Create New Role dialog box.

Content Explorer Shortcuts

The following table describes general shortcuts that may be used within the Content Explorer.

Table 21-2
Content Explorer shortcuts

Shortcut	Action
Up/down Arrows	Highlights items in the tree.
Enter	Opens items.

Table Shortcuts

The following table describes shortcuts that may be used within tables appearing in views and dialog boxes. Use the Tab key to select the table for subsequent navigation.

Table 21-3
Table shortcuts

Shortcut	Action
Up/down Arrows	Highlights items in the tables.
Right arrow	Expands a selected item, if the item can be expanded.
Left arrow	Collapses an expanded item.
Ctrl+<column number>	For a highlighted row, moves focus to the cell in the specified column if the cell value can be modified. Both keys must be released simultaneously. If the value cannot be modified, the focus is unchanged.

Job History and Job Schedule Shortcuts

The following table describes general shortcuts that may be used within the Job History and the Job Schedule.

Table 21-4
Job History and Job Schedule shortcuts

Shortcut	Action
Up/down Arrows	Highlights items in the tables.

Shortcut	Action
Right arrow	■ Highlights items in tables. ■ Expands a selected item, if the item can be expanded.
Left arrow	Collapses an expanded item.
Enter	Opens a selected item in the editor. (*Note*: For jobs, use Shift + F10 to bring up a context menu on a selected Job and select Open In Job Editor.)

Job Editor Shortcuts

The following table describes general shortcuts that may be used within the Job Editor.

Table 21-5
Job Editor shortcuts

Shortcut	Action
Up/down Arrows	■ Highlights buttons in the job palette. ■ Moves the cursor around in the graphical editor.
Left/right Arrows	■ Cycles the selection through the different job steps and the event flow items as well. ■ Moves a job step around the editor if the Move anchor was selected. ■ Moves the cursor around in the graphical editor.
Period (.)	Cycles through the move; resizes anchors on the currently selected job step.
Plus sign (+)	Adds a new step of the type that is selected in the palette to the job.

Help System Shortcuts

The following table describes general shortcuts that may be used within the help system.

Table 21-6
Help system shortcuts

Shortcut	Action
Up/down Arrows	On the Contents tab, moves up and down through the topics.
Right Arrow	On the Contents tab, expands any subtopics of the current topic.
Left Arrow	On the Contents tab, moves to the parent of the current topic. If the current topic has expanded subtopics, the subtopics are closed.
Shift+F10	On the Contents tab, displays the context menu, from which the selected topic can be opened.

Chapter

22

Support Information

Support Resource Overview

The following additional resources are available:

- Documentation resources
- Technical support

Documentation Resources

The following documentation is available for the system:

Online help. The online help system is accessible from the system's user interface. For more information, see the topic Accessing Help in Chapter 3 on p. 12.

PDF documentation. The following additional documentation is available in portable document format (PDF).

- *Deployment Manager 4.1 User's Guide*
- *PASW Collaboration and Deployment Services 4.1 Administrator's Guide* (*administrator_guide.pdf*)
- *PASW Collaboration and Deployment Services 4.1 Installation and Configuration Guide* for each supported platform

Technical Support

You can contact a Technical Support representative at:

SPSS Inc. Technical Support
SPSS Inc.
233 S. Wacker Drive
11th Floor
Sears Tower
Chicago, Illinois 60606

Phone: 800.443.8802
Web site: http://support.spss.com

Index

508 compliance, 343

abs function, 186
accessibility, 343
  product features, 343
accessing
  expired files, 53
add data set
  Application Server Data Source, 170
  connection settings, 169
  Data Service data source, 171
  JDBC data source, 169
  select tables to add, 170
adddays function, 190
adding
  job variables, 235
  principals for label, 57
adding columns
  connection settings, 128
  select columns, 129
adding steps, 236
adding tables
  connection settings, 128
  select columns, 129
addition function, 184
addmonths function, 191
addyears function, 191
advanced search, 22–23, 25–30
aggregate functions, 175, 178
  handling nil values, 176
and function, 199
anomaly detection models
  scoring, 213
any data type, 175
any:1 data type, 175
application
  exiting, 14
  navigating, 10
  starting, 10
Application Server Data Source
  add data set, 170
application server data sources, 86, 91
Application View, 146
  changing, 158
  column attributes, 148
  column count, 148
  columns, 147
  create, 147
  direction, 151
  merging versions, 153–154
  removing columns, 148
  removing tables, 148
  saving, 151
  tables, 147
Application View Editor, 147
applying
  locks, 31–32
arithmetic functions, 184
attachment, 281, 287
auditing
  for scoring, 220
avg function, 179

basic functions, 175
  handling nil values, 176
batch scoring, 223
Bayesian networks
  scoring, 213
between function, 200
BIRT steps, 294
  adding to jobs, 294
  burst, 296
  clean-up, 300
  data source, 295
  defining general properties, 294
  iterative, 296
  locale, 299
  metadata, 299
  naming, 295
  notifications, 301
  output file format, 299
  output file location, 299
  output file name, 299
  parameters, 298
  prompts, 298
  rendering options, 299
  result permissions, 299
  results, 298
  single, 296
  type, 296
  variables, 298
  versions, 295
bit function, 187
blank cells
  in job history, 264
bulk property updates, 77
  general, 78
  permission, 81
  version, 80

cache hits
  for scoring configurations, 221

cache misses
for scoring configurations, 222
cache sizes
for scoring, 223
canceling
jobs, 264
categories
column attributes, 135
exporting, 138
removing, 139
cells
blank, 264
reordering, 262
champion challenger, 265, 267
changing
Application View, 158
Enterprise View, 152
passwords, 19
chrAt function, 203
clean up, 300, 307
client
starting, 10
cluster models
scoring, 213
clusters, 103
collaboration, 1
column
select columns, 129
column attributes
Application View, 148
categories, 135
column name, 133, 149
description, 140, 151
direction, 151
Enterprise View, 133
environment, 150
min/max, 139
type, 134, 149
column count, 148
column name
column attributes, 133, 149
columns
Application View, 147
Enterprise View, 131
index, 133, 149
reordering, 262
sorting, 132, 148
concatenate function, 202
conditional connector, 238
confidences
in scoring, 212
configuration objects, 158
configurations
for scoring models, 214
conflict resolution, 114–115
duplicates, 117, 119–120
individual, 117, 119–120
invalid versions, 117, 119–120
conflicts
duplicates, 117, 119–120
import, 114–115, 117, 119–120
invalid versions, 117, 119–120
connection
existing, 17
new, 16
server, 16–19
terminating, 18
connection settings
add data set, 169
adding columns, 128
adding tables, 128
select data set to add, 171
connections
server, 54–55, 60
connectors
conditional, 238
deleting, 242
fail, 238
pass, 238
relationship, 237–238, 240, 242
sequential, 238
content assist, 12–13
content explorer
deleting files, 21
files, 20–21
jobs, 230, 232
organization, 15
overview, 15
permissions, 20
properties, 34
search, 22–23, 25–30
searching, 29
server connections, 16–19
versions, 46, 48–49, 61
content object
properties, 34
content objects
definition, 15
permissions, 20
storage, 15, 20–21
versions, 46, 48–49
content repository, 15, 20–21, 341–342
deleting files, 21
Content Repository
execution server, 99
server definition, 99
conventions
naming, 14
conversion functions, 187
copying
topics, 75
count function, 179
Cox regression models
scoring, 213

create
  Application View, 147
  Data Provider Definition, 155
  Data Provider Definition - Real Time, 163
creating
  custom properties, 64–66
  schedules, 252
  server clusters, 103
  topics, 73–74
credential definitions, 82–84, 124
credential destination, 83
credentials, 95–96
  for schedules, 252
  job, 229
cumulative hazard values
  in scoring, 212
custom properties
  accessing, 65
  creating, 37, 64–66, 69–70
  defining, 37
  deleting, 71
  editing, 37, 69–70
  existing, 37, 69–70
  label, 66
  modifying, 37, 69–70
  new, 65
  overview, 64
  property type, 66
  property types, 37
  searching, 71
  selection values, 37, 66, 69–70
  setting values, 64
customizing
  notification messages, 288

daily schedules, 254
data files, 229
Data Provider Definition, 155, 296
  configuration objects, 158
  create, 155
  mapping columns, 159
  saving, 162
  select Application View, 157
  validating, 160
Data Provider Definition - Real Time, 162
  create, 163
  defining data sets, 169
  editing data sets, 171
  mapping data set columns, 166
  preview data, 167
  saving, 172
  select Application View, 165
  working with data sets, 168
Data Provider Definition - Real Time Editor, 165
Data Provider Definition Editor, 158
data provider definitions
  real-time, 217
Data Service data source
  add data set, 171
data service data sources, 86, 91–92
data source
  BIRT, 295
  reporting step, 295
data source definitions, 82
Data Source Definitions, 124, 206
data sources, 85
  application server data sources, 86, 91
  data service data sources, 86, 91–92
  JDBC data sources, 86–87, 90
  modifying, 93
  ODBC data sources, 86
data types, 175, 207
databases, 229
date and time functions, 190
`date` function, 187, 192
date range, 23, 25–26
dates
  expiration, 50–51, 53–54
`day` function, 192
`dayofweek` function, 192
`dayofyear` function, 193
`dbl` function, 187
`decimal` function, 188
defining data sets
  Data Provider Definition - Real Time, 169
delete search terms, 29
deleting, 29
  custom properties, 71
  files, 21
  groups, 46
  job history, 262
  job schedule, 262
  job variables, 236
  principals for label, 58
  schedules, 258
  scoring configurations, 226
  search terms, 29
  topics, 76
  users, 46
  versions, 49
delivery failure, 292
deployment, 2
Deployment Manager, 3–4
Deployment Portal, 3–4
deprecated
  commands, 9
  features, 9
deprecated features, 9
description, 61
  column attributes, 140, 151
destination
  credentials, 83
  server, 98
destination name, 94–96

DEVICE, 311, 313, 316
diff function, 193
direction
  column attributes, 151
discriminant models
  scoring, 213
division function, 186
drag and drop, 11
duplicates, 117, 119–120
durable subscriptions, 255
Dynamic notification lists, 285–287

e-mail delivery failure, 292
edit search terms, 27
editing, 27
  custom properties, 37, 69–70
  job variables, 235
  permissions for label, 58
  properties, 35
  schedules, 258
  scoring configurations, 225
  server clusters, 106
  topics, 39
editing data sets
  Data Provider Definition - Real Time, 171
editor
  Application View, 147
  Data Provider Definition, 158
  Data Provider Definition - Real Time, 165
  Enterprise View, 125
editors
  Application View, 123
  Data Provider Definition, 123
  Enterprise View, 123
enabling
  filtering, 271, 273
end date
  for schedules, 254
endsWith function, 203
enter key, 11
Enterprise Server, 101
Enterprise View, 3, 5, 124
  changing, 152
  column attributes, 133
  columns, 131
  credential definitions, 124
  Data Source Definitions, 124, 206
  data types, 207
  editors, 123
  manage keys, 140
  manage links, 142
  merging versions, 153–154
  removing columns, 132
  removing tables, 130
  saving, 146
  tables, 126
Enterprise View Editor, 125
environment
  column attributes, 150
equal to function, 198
error messages
  for schedules, 263
evaluation
  results, 265–267, 269
exclusions
  search terms, 30
executing jobs, 237
execution servers, 5
  remote process, 6
  remote process servers, 310, 318
  SAS, 6, 310
exist function, 180
existing jobs, 232
exiting
  application, 14
  client, 14
  system, 14
expiration
  submitted jobs, 342
expiration date, 61
expiration dates, 342
  exporting, 53
  importing, 53
  modifying, 51
  overview, 50
  reactivating, 54
  restrictions, 53
  searching, 53
  setting, 51
  viewing, 53
export, 278
  external references, 108
  folders, 108–110, 114–115, 117, 119–120
  jobs, 108
exporting, 108, 110, 115, 117, 119–120
  categories, 138
  conflicts, 114–115, 117
  expiration dates, 53
  locks, 33
  resource definitions, 106
  restrictions, 109
exporting locks, 33
expression editor, 172
  expression panel, 173
  functions panel, 174
expression panel, 173
  of expression editor, 173
expression status field, 175
external files
  in SAS steps, 314
external references, 108
  exporting, 108
  importing, 108

F1 help, 12
fail connector, 238
features
  deprecated, 9
  obsolete, 9
field content assist, 12–13
file
  permissions, 246
  subscriptions, 290
  versions, 46, 48–49, 61
file menu, 11
file types
  for folder notifications, 289
FILENAME, 312
fileref, 312
files, 20
  adding, 20
  adding to repository, 108
  copying, 20
  deleting, 21
  downloading, 21, 108
  expired, 53
  expiring, 50–51, 53–54
  exporting, 33
  importing, 33
  locking, 31–32
  moving, 20
  opening, 20
  permissions, 20
  .pes, 108, 110
  reactivating, 54
  unlocking, 33
filtering
  enabling, 271, 273
  job history, 273
  job schedule, 271
  scoring view, 224
filters, 271, 273
`flt` function, 188
folder
  resource definitions, 82–84, 96–101
  search, 22–23, 25–30
  searching, 29
folder options, 281
folders
  child, 45
  content repository, 341–342
  expiration of, 342
  exporting, 108–110, 114–115, 117, 119–120
  importing, 108–109, 111, 114–115, 117, 119–120
  parent, 45
  resource definitions, 15
  restrictions, 341
  submitted jobs, 341–342
functions
  introduction, 175
  types, 175
functions panel, 173, 175
  of expression editor, 174

General job steps, 317
  adding to jobs, 317
  defining properties, 317
  examples, 321, 325
  input files, 318–319
  naming, 317
  output files, 320–321
  working directory, 318, 320–321
general linear models
  scoring, 213
general properties, 35
generalized linear models
  scoring, 213
getting started, 123
global conflict resolution, 115
`greater than` function, 198
`greater than or equal to` function, 199
group search terms, 26
groups, 247
  deleting, 46
  existing, 44
  new, 42
  permissions, 246
GSFNAME, 312
guidelines
  naming, 14

help
  accessing, 12
  F1, 12
history
  job, 261–264, 273
  job step, 264–265
  server status, 275, 277
  updating, 261
hourly schedules, 254
HTML output
  for PASW Statistics steps, 312
HTMLC, 299

`ifnil` function, 195
`implies` function, 200
import, 278
  external references, 108
  folders, 108–109, 111, 114–115, 117, 119–120
  jobs, 108
  security permissions, 109
import order, 109
importing, 108–109, 111, 115, 117, 119–120
  conflicts, 114–115, 117
  expiration dates, 53
  locks, 33

resource definitions, 106
restrictions, 109
importing locks, 33
`in` function, 201
index
columns, 133, 149
individual conflict resolution, 117, 119–120
inheritance
permissions, 45
input files
General job steps, 318–319
`int` function, 189
invalid versions, 117, 119–120
`isnil` function, 201
iterative principals, 247

Java, 343
JAWS, 343
JDBC data source
add data set, 169
JDBC data sources, 86–87
third-party drivers, 90
JMS, 94–96, 333
mapping variables, 257
JMS message domain, 258
job
notifications, 278
subscriptions, 290
Job Editor
accessing, 261
opening job, 261
job history
deleting, 262
filtering, 273
limiting, 273
refreshing, 261
reordering, 262, 273
reorganizing, 262
selecting server, 261
updating, 261
viewing, 261–264, 273
job labels
for schedules, 252
job output, 265
job schedule
deleting, 262
filtering, 271
limiting, 271
refreshing, 261
reordering, 262, 271
reorganizing, 262
selecting server, 261
updating, 261
viewing, 261–262, 271
viewing in Job Editor, 261
job step
notifications, 279
permissions, 246
job step history
viewing, 264–265
job step output, 245
job steps
history, 264–265
results, 264–265
job variables
in schedules, 256–257
logging values, 265
jobs
accessing, 261
adding steps, 232, 236
canceling, 264
components, 229
content explorer, 230, 232
creating, 230
definition, 228
dependencies, 229
editing, 232, 236
existing, 232
exporting, 108
filtering, 271, 273
history, 261–265, 273
importing, 108
job editor, 232, 236–238, 240, 242
launching, 261
logs, 265
new, 230
notifications, 251
opening, 230, 232, 261
overview, 228
prerequisites, 229
process, 230
properties, 232
refreshing, 261
relationships, 237–238, 240, 242
reordering, 262
running, 237, 250–251, 253–254, 262
saving, 243
schedule, 261–262, 271
scheduling, 250, 253–254, 262
searching, 271, 273
selecting, 261
status, 262–265
submitted, 341–342
variables, 234, 251

keyboard shortcuts, 343
keys, 140
keywords, 61
Kohonen models
scoring, 213

labels
filtering, 271, 273

permissions, 57
security, 56
server version, 60
version, 48–49
latency
for scoring configurations, 221
`length` function, 204
`less than` function, 197
`less than or equal to` function, 197
linear regression models
scoring, 213
links, 123, 142
`lng` function, 189
locking objects, 31–33
locks
applying, 31–32
exporting, 33
files, 31–33
importing, 33
objects, 31–33
removing, 33
resource definitions, 31–33
log in
server, 17
log off
server, 18
logging
for scoring, 220
logistic regression models
scoring, 213
logs
job, 265
output, 265
Lotus, 284
`ltrim` function, 204

manage keys
Enterprise View, 140
manage label permission
for labels, 57
manage links
Enterprise View, 142
map columns, 160, 166
mapping columns
Data Provider Definition, 159
mapping data set columns
Data Provider Definition - Real Time, 166
`max` function, 181
menu
file, 11
merging
version conflicts, 154
view versions, 153–154
message customization, 281
body text, 288
errors, 288
event property variables, 288
HTML formatting, 288
message customization, 288
preview, 288
subject, 288
template, 288
message domain, 94–96, 333
message domain attributes, 95–96
message domain name, 94–95
message domain properties, 94
message domains
for schedules, 255
message filters
for schedules, 255
message selector, 333
message template, 338, 340
message text, 333
message-based job, 94
message-based processing example, 258
message-based step, 94
message-driven step, 333
defining properties, 333
metadata, 77–78, 80–81
user-defined, 37, 39, 64–66, 69–71, 73–77
version properties, 61
metrics
for scoring performance, 221
`min` function, 182
min/max
column attributes, 139
miscellaneous functions, 195
model evaluation, 265–266, 281, 335
notifications, 281
return codes, 281
model management, 265–267
modifying
permissions, 41–42, 44
server clusters, 106
`modulo` function, 184
`month` function, 194
monthly schedules, 254
mouse, 11
`multiplication` function, 184

naming conventions, 14
naming factory, 94–96
naming service, 94–96
narrow search, 23, 25–26
narrowing, 23, 25–26
nave Bayes models
scoring, 213
navigating
keyboard shortcuts, 343
navigating through the system, 123
navigation, 10, 123
enter key, 11
mouse, 11

nearest neighbors
  in scoring, 212
`negation` function, 185
Netezza, 90, 124
neural networks
  scoring, 213
new
  custom properties, 64–66
  topics, 73–74
new jobs, 230
nil values, 176
node identifiers
  in scoring, 212
NOSPLASH, 310
NOSTATUSWIN, 310
`not equal to` function, 196
`not` function, 201
notification job step
  adding to a job, 335
  body, 338
  connecting to other job steps, 335
  customizing , 338
  e-mail attachments, 335
  from address, 338
  general information, 336
  iterative consumer, 336
  message template, 338, 340
  Notification tab, 337
  outcome-based, 335
  recipients, 337–338
  subject, 338
  use in model evaluation, 335
notifications, 251, 278
  attachment, 281
  body text, 288
  content notifications, 280
  defining, 280
  disabling, 251
  entering recipient e-mail, 283
  errors, 288
  event property variables, 288
  file types, 289
  folder events, 280
  folder notifications, 280
  folder options, 281
  folder structure changes, 280
  for exported and imported objects, 278
  generating a list of recipients dynamically, 285–287
  HTML formatting, 288
  job, 278
  job step, 279
  message customization, 281, 288
  message subject, 281
  model evaluation, 281
  modifying, 280
  preview, 281, 288
  previewing message, 288
  recipients, 281
  removing, 280–281
  reverting to the default template, 281
  selecting recipients from Lotus Notes, 284
  selecting recipients from MS Outlook, 283
  sender, 281
  subject, 288
  template, 281, 288
notifications delivery failure, 292

object permissions
  cascading, 45
  deleting, 46
  modifying, 41–42, 44
object properties, 34
  editing, 35
  viewing, 34–35, 54, 60
objects
  content, 15, 20–21
  deleting, 21
  files, 20–21
  locking, 31–33
  permissions, 20
  search, 22–23, 25–30
  unlocking, 33
  versions, 61
obsolete features, 9
ODBC, 230
ODBC data sources, 86
ODS, 312
`or` function, 202
ordinal multinomial regression models
  scoring, 213
Outlook, 283
output, 245
  log files, 265
output file permissions, 246
output files
  General job steps, 320–321
output permissions, 246
overview, 123
  content explorer, 15
  jobs, 228, 230
  new features, 9
  resource, 346
  system, 9
  versions, 46

pass connector, 238
passwords
  changing, 19
  modifying, 19
  new, 19
PASW BIRT Report Designer, 3, 6
permission inheritance, 45
permissions
  applying, 45

assigning, 42, 44
cascading, 45
creating, 42, 44
deleting, 46
editing for label, 58
for labels, 57
for reporting results, 299
for SAS results, 312
for visualization results, 306
inheritance, 20
job step, 246
modifying, 41–42, 44–46
object, 41–42, 44, 46
output file, 246
removing, 46
specifying as iterative variables, 247
updating in bulk, 81
when importing, 109
PMML, 212
PMML files, 321, 325
predicted values, 212
predictor, 151
predictor effectiveness, 269
prerequisites
jobs, 229
preview, 281
preview button, 178
preview data
Data Provider Definition - Real Time, 167
preview of selection, 178
previewing notification message, 288
principals
adding for label, 57
deleting for label, 58
modifying label permissions, 58
privileges
access, 64–66, 73–74
administrative, 64–66, 73–74
probabilities
in scoring, 212
properties
bulk updates, 77–78, 80–81
content object, 34
custom, 37, 39, 64–66, 69–71, 74
definition, 34
deleting, 28–29
editing, 27, 35
general, 35
object, 34–35
overview, 34
server, 37, 39, 54–55, 60, 64–66, 69–71, 73–77
topics, 73–77
viewing, 34–35, 54–55, 60
property, 23, 25–26
property types
custom properties, 37
property variables, 12–13
range of recurrence
for schedules, 254
reactivating
expired files, 54
recipient
of notification, 281
recipients, 281, 287
recurrence pattern
for schedules, 254
references
external, 108
refine search, 23, 25–29
relationships
between steps, 237–238, 240
deleting, 242
remote process
execution servers, 6
remote process servers
assigning to job steps, 310, 318
server definition, 100
removing, 281
categories, 139
custom properties, 71
locks, 33
topics, 76
removing columns
Application View, 148
Enterprise View, 132
removing tables
Application View, 148
Enterprise View, 130
renaming
topics, 75
reorder search terms, 28
reordering, 28
search terms, 28
reporting steps, 294, 302
adding to jobs, 294
burst, 296
clean-up, 300
data source, 295
defining general properties, 294
iterative, 296
locale, 299
metadata, 299
naming, 295
notifications, 301
output file format, 299
output file location, 299
output file name, 299
parameters, 298
prompts, 298
rendering options, 299
result permissions, 299
results, 298
single, 296
type, 296

variables, 298
versions, 295
reports
external, 341–342
submitted, 341–342
repository, 3
connecting to, 16–17
content, 15, 20–21
deleting, 21
disconnecting from, 18
exporting, 33
files, 20–21
host, 55
importing, 33
locking objects, 31–33
objects, 31–33
port, 55
search, 22–23, 25–30
SSL, 55
unlocking objects, 33
resolving conflicts
globally, 115
individually, 117, 119–120
resource definitions
credential definitions, 82–84
data source definitions, 82
data sources, 85
exporting, 106
importing, 106
locking, 31–33
server definitions, 82, 96–101
resource definitions folder, 15
resources
additional, 346
overview, 346
restrictions
expiration dates, 53
exporting, 109
importing, 109
submitted jobs, 341
results, 245, 264
analysis, 265–267, 269
champion challenger, 265, 267
job, 263, 265
job step, 264–265
model evaluation, 265–266
predictor effectiveness, 269
scenario, 265–267, 269
search, 29
resuming
scoring configurations, 226
return codes
model evaluation, 281
notifications, 281
reverting to the default template, 281
right-click, 11
rows
filtering, 271, 273
rtrim function, 204
ruleset models
scoring, 213
running jobs, 237, 250–251, 262

SAS
execution server, 6, 99
server definition, 99
SAS execution servers
assigning to job steps, 310
SAS steps, 309
adding to jobs, 309
controlling processing, 310, 316
creating graphs, 311–312, 315
defining properties, 309, 311
example, 313
external files, 314
HTML output, 312
naming, 309
naming results, 311
result locations, 311
result permissions, 312
text output, 312
versions, 309
save
Application View, 151
Data Provider Definition, 162
Data Provider Definition - Real Time, 172
Enterprise View, 146
saving jobs, 243
scenario
results, 265–267, 269
status, 265–267, 269
schedule
job, 261–262, 271
schedules, 251
creating, 252
credentials, 252
deleting, 258
dormant, 258
editing, 258
job labels, 252
job variables, 256–257
message-based, 254
time-based, 253
scheduling jobs, 250, 262
schema browser, 173
scoring
configurations, 214
models, 212
performance, 226
using PMML files, 325
scoring configurations, 214, 224
auditing, 220
batch scoring, 223

cache sizes, 223
configuration names, 215
creating, 214
data, 215–216
deleting, 226
editing, 225
input, 217–218
labels, 215
logging, 220
models, 215
output, 219
resuming, 226
suspending, 225
viewing, 224
scoring functions, 212
scoring graph view, 226
scoring view, 224
deleting configurations, 226
editing configurations, 225
filtering, 224
resuming configurations, 226
suspending configurations, 225
screen elements, 123
screen readers, 343
search, 27–29
accessing, 22–23
advanced, 22–23, 25–30
AND, 23, 25–26
content objects, 22–23, 25–30
date range, 23, 25–26
deleting, 29
dialog, 22–23
expiration dates, 53
files, 22–23, 25–30
group, 26
OR, 23, 25–26
property, 23, 25–26
refining, 23, 25–29
reordering, 28
simple, 22–23, 25–26, 30
time, 23, 25–26
ungroup, 26
`search` function, 204
search terms, 23, 25
deleting, 29
editing, 27
group, 26
properties, 27–29
reordering, 28
stopwords, 30
ungroup, 26
searching
content objects, 29
custom properties, 71
files, 29
for jobs, 271, 273
submitted jobs, 341
topics, 77
security
for labels, 56
security providers, 247
select Application View
Data Provider Definition, 157
Data Provider Definition - Real Time, 165
select data set to add
connection settings, 171
select tables to add
add data set, 170
selecting
servers, 261
versions, 49
selection
drag and drop, 11
file menu, 11
right-click, 11
selection functions, 196
selection values
deleting, 37, 39, 69–70
editing, 37, 39, 69–70
modifying, 37, 39, 69–70
removing, 37, 39, 69–70
sender, 281
of notification, 281
sequential connector, 238
server
change password, 19
connection, 54–55, 60
custom properties, 37, 64–66, 69–71
log in, 16–17, 19
log off, 16, 18
properties, 54–55, 60
topics, 39, 73–76
version label, 60
server clusters, 103
adding servers, 104–105
creating, 103
modifying, 106
name, 103
removing servers, 105
settings, 104
weights, 105
server connection, 16
changing password, 19
ending, 18
existing, 17
new, 16
terminating, 18
server connections
jobs, 229
server definitions, 82, 96
adding, 97
content repository, 99
Enterprise Server, 101
location, 98

modifying, 102
names, 97
remote process, 100
SAS, 99
types, 97
server destination, 98
server status, 275, 277
servers
choosing, 261
remote process servers, 310, 318
SAS, 310
selecting, 261
shortcut keys, 343
shortcuts
keyboard, 343
ShowCase Enterprise Server, 101
simple search, 22–23, 25–26, 30
`size` function, 182
sorting
columns, 132, 148
SSL, 55
standard deviations
in scoring, 212
start date
for schedules, 254
start time
for schedules, 253
starting
application, 10
client, 10
system, 10
`startsWith` function, 205
status
filtering, 271, 273
job, 263
job step, 264–265
refreshing, 261
server, 275, 277
updating, 261
steps
adding, 236
executing, 237
job, 228–230, 232, 236–237
relationships, 237–238, 240, 242
`str` function, 190
`string` function, 205
string functions, 202
subject, 281
submitted jobs
expiration dates, 342
restrictions, 341
searching, 341
subscribing to files, 290
subscription delivery failure, 292
subscriptions
e-mail address for, 291
files, 290
for exported and imported objects, 278
jobs, 290
managing, 292
modifying, 291
removing, 292
subscribing to files, 290
unsubscribing, 291
user preferences, 291
`subtraction` function, 185
`sum` function, 183
support vector machine models
scoring, 213
suspending
scoring configurations, 225
system
content explorer, 15
exiting, 14
help, 12
jobs, 228
launching, 10
naming files, 14
navigating, 10–11
starting, 10, 16

table
search results, 29
select columns, 129
tables
Application View, 147
champion challenger, 265, 267
Enterprise View, 126
filtering, 271, 273
job history, 261–264, 273
job schedule, 261–262, 271
job step history, 263–265
model evaluation, 265–266
predictor effectiveness, 269
refreshing, 261
server status, 275, 277
updating, 261
target, 151
technical support, 346
template, 281
Teradata, 90, 124
terms
deleting, 29
editing, 27
group, 26
reordering, 28
search, 23, 25–26
search exclusions, 30
ungroup, 26
text output
for SAS steps, 312
time, 23, 25–26
timeout, 333
`timestamp` function, 190

toLower function, 206
topics, 72
  accessing, 74
  copying, 75
  creating, 39, 73–74
  deleting, 76
  editing, 39
  existing, 39
  label, 74
  modifying, 39
  new, 74
  property type, 74
  renaming, 75
  searching, 77
  selection values, 39, 74
toUpper function, 206
tree models
  scoring, 213
type
  column attributes, 134, 149
type field, 175

UI, 123
UNC file references, 230
ungroup search terms, 26
unlocking, 33
unlocking objects, 33
unmap columns, 160, 166
uptime
  for scoring configurations, 221
use versions permission
  for labels, 57
user interface, 123
user preferences, 4
  e-mail, 291
users, 247
  deleting, 46
  existing, 44
  new, 42
  permissions, 246

validate
  Data Provider Definition, 160
variables
  adding to jobs, 235
  editing, 235
  in jobs, 234, 251
  removing from jobs, 236
version
  description, 61
  expiration date, 61
  keywords, 61
  metadata, 61
  properties, 61
  values, 61
version label
  filtering, 271, 273
version properties
  updating in bulk, 80
versions
  deleting, 49
  invalid, 119–120
  labeling, 48–49, 60
  overview, 46
  selecting, 49
viewing
  expired files, 53
  job history, 261–264, 273
  job output, 265
  job schedule, 261–262, 271
  job step history, 263–265
  logs, 265
  properties, 34–35, 54–55, 60
  search results, 29
  server status, 275, 277
views
  champion challenger, 265, 267
  model evaluation, 265–266
  model management, 265–267
  predictor effectiveness, 269
  server status, 275, 277
visualization reports
  adding to jobs, 302
  clean-up, 307
  data source, 303
  defining general properties, 302
  dimensions, 306
  iterative, 304
  metadata, 306
  naming, 303
  notifications, 308
  output file location, 306
  parameters, 305
  prompts, 305
  result permissions, 306
  results, 306
  single, 304
  type, 304
  variables, 305
  versions, 303
visualization reports steps
  output file format, 306

weekofyear function, 194
weely schedules, 254
weights
  in server clusters, 105
when function, 195
  handling nil values, 176
working directory
  for General job steps, 318, 320–321
working with data sets
  Data Provider Definition - Real Time, 168

`year` function, 194