

*IBM SPSS Modeler Server 18.5
Administration and Performance Guide*



Note

Before you use this information and the product it supports, read the information in [“Notices” on page 91.](#)

Product Information

This edition applies to version 18, release 4, modification 0 of IBM® SPSS® Modeler and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation .**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- Preface..... vii**

- Chapter 1. About IBM SPSS Modeler 1**
 - IBM SPSS Modeler Products..... 1
 - IBM SPSS Modeler 1
 - IBM SPSS Modeler Server 1
 - IBM SPSS Modeler Administration Console 2
 - IBM SPSS Modeler Batch 2
 - IBM SPSS Modeler Solution Publisher 2
 - IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services 2
 - IBM SPSS Modeler Editions..... 2
 - Documentation..... 3
 - SPSS Modeler Professional Documentation..... 3
 - SPSS Modeler Premium Documentation..... 4
 - Application examples..... 4
 - Demos Folder..... 4
 - License tracking..... 4

- Chapter 2. Architecture and Hardware Recommendations..... 5**
 - IBM SPSS Modeler Architecture..... 5
 - Architecture Description..... 5
 - Hardware Recommendations..... 6
 - Temporary Disk Space and RAM Requirements..... 7
 - Data Access..... 8
 - Referencing Data Files..... 10
 - Importing IBM SPSS Statistics Data Files..... 10
 - Installation Instructions..... 10

- Chapter 3. IBM SPSS Modeler Support..... 11**
 - Connecting to IBM SPSS Modeler Server 11
 - Configuring single sign-on..... 12
 - Adding and Editing the IBM SPSS Modeler Server Connection..... 17
 - Searching for Servers in IBM SPSS Collaboration and Deployment Services 18
 - Data and File Systems..... 18
 - User Authentication..... 18
 - Permissions..... 19
 - File Creation..... 19
 - Differences in Results..... 20

- Chapter 4. IBM SPSS Modeler Administration..... 21**
 - Starting and Stopping IBM SPSS Modeler Server 21
 - To Start, Stop, and Check Status on Windows..... 21
 - To Start, Stop, and Check Status on UNIX..... 21
 - Handling Unresponsive Server Processes (UNIX Systems)..... 22
 - Configuring server profiles..... 23
 - Working with server profiles..... 23
 - Profile structure..... 25
 - Profile scripts..... 27
 - Administration..... 31
 - IBM SPSS Modeler Server Administration..... 31

Starting Modeler Administration Console	31
Restarting the web service.....	31
Configuring Access with Modeler Administration Console	32
SPSS Modeler Server connections.....	32
SPSS Modeler Server Configuration.....	33
SPSS Modeler Server Monitoring.....	40
Using the options.cfg file.....	40
Closing unused database connections.....	41
Using SSL to secure data transfer.....	41
How SSL works.....	41
Securing client/server and server-server communications with SSL.....	42
Cognos SSL connection.....	46
Cognos TM1 SSL connection.....	46
Configuring groups.....	47
Log files.....	50
Chapter 5. Performance Overview.....	53
Server performance and optimization settings.....	53
Client Performance and Optimization Settings.....	53
Database Usage and Optimization.....	54
SQL Optimization.....	55
Stream performance.....	55
Chapter 6. SQL optimization.....	57
How SQL generation works.....	58
SQL Generation Example.....	58
Configuring SQL Optimization.....	60
Previewing Generated SQL.....	60
Viewing SQL for Model Nuggets.....	60
Tips for maximizing SQL generation.....	61
Nodes supporting SQL generation.....	62
CLEM Expressions and Operators Supporting SQL Generation.....	66
Using SQL Functions in CLEM Expressions.....	69
Writing SQL Queries.....	69
Scoring adapter for Teradata - duplicate rows.....	70
Appendix A. Configuring Oracle for UNIX Platforms.....	71
Configuring Oracle for SQL Optimization.....	71
Appendix B. Configuring UNIX Startup Scripts.....	73
Introduction.....	73
Scripts.....	73
Automatically Starting and Stopping IBM SPSS Modeler Server	73
Manually Starting and Stopping IBM SPSS Modeler Server	73
Editing Scripts.....	74
Controlling Permissions on File Creation.....	74
IBM SPSS Modeler Server and the data access pack.....	74
Troubleshooting ODBC configuration.....	77
Library paths.....	79
Appendix C. Configuring and Running SPSS Modeler Server as a Non-Root Process on UNIX.....	81
Introduction.....	81
Configuring as non-root without a private password database.....	81
Configuring as non-root using a private password database.....	82
Running SPSS Modeler Server as a non-root user.....	83

Troubleshooting user authentication failures.....	83
Appendix D. Configuring and Running SPSS Modeler Server with a private password file on Windows.....	85
Introduction.....	85
Configuring a private password database.....	85
Appendix E. Load Balancing with Server Clusters.....	87
Appendix F. LDAP authentication.....	89
Notices.....	91
Trademarks.....	92
Terms and conditions for product documentation.....	92
Index.....	95

Preface

IBM SPSS Modeler is the IBM Corp. enterprise-strength data mining workbench. SPSS Modeler helps organizations to improve customer and citizen relationships through an in-depth understanding of data. Organizations use the insight gained from SPSS Modeler to retain profitable customers, identify cross-selling opportunities, attract new customers, detect fraud, reduce risk, and improve government service delivery.

SPSS Modeler's visual interface invites users to apply their specific business expertise, which leads to more powerful predictive models and shortens time-to-solution. SPSS Modeler offers many modeling techniques, such as prediction, classification, segmentation, and association detection algorithms. Once models are created, IBM SPSS Modeler Solution Publisher enables their delivery enterprise-wide to decision makers or to a database.

About IBM Business Analytics

IBM Business Analytics software delivers complete, consistent and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of business intelligence, predictive analytics, financial performance and strategy management, and analytic applications provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices and professional services, organizations of every size can drive the highest productivity, confidently automate decisions and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government and academic customers worldwide rely on IBM SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises - able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative visit <https://www.ibm.com/mysupport/s/>.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. website at <https://www.ibm.com/mysupport/s/>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Chapter 1. About IBM SPSS Modeler

IBM SPSS Modeler is a set of data mining tools that enable you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data to better business results.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

SPSS Modeler can be purchased as a standalone product, or used as a client in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see <https://www.ibm.com/analytics/us/en/technology/spss/>.

IBM SPSS Modeler Products

The IBM SPSS Modeler family of products and associated software comprises the following.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler is a functionally complete version of the product that you install and run on your personal computer. You can run SPSS Modeler in local mode as a standalone product, or use it in distributed mode along with IBM SPSS Modeler Server for improved performance on large data sets.

With SPSS Modeler, you can build accurate predictive models quickly and intuitively, without programming. Using the unique visual interface, you can easily visualize the data mining process. With the support of the advanced analytics embedded in the product, you can discover previously hidden patterns and trends in your data. You can model outcomes and understand the factors that influence them, enabling you to take advantage of business opportunities and mitigate risks.

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. See the topic “[IBM SPSS Modeler Editions](#)” on page 2 for more information.

IBM SPSS Modeler Server

SPSS Modeler uses a client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger data sets.

SPSS Modeler Server is a separately-licensed product that runs continually in distributed analysis mode on a server host in conjunction with one or more IBM SPSS Modeler installations. In this way, SPSS Modeler Server provides superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. IBM SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation.

IBM SPSS Modeler Administration Console

The Modeler Administration Console is a graphical user interface for managing many of the SPSS Modeler Server configuration options, which are also configurable by means of an options file. The console is included in IBM SPSS Deployment Manager, can be used to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

IBM SPSS Modeler Batch

While data mining is usually an interactive process, it is also possible to run SPSS Modeler from a command line, without the need for the graphical user interface. For example, you might have long-running or repetitive tasks that you want to perform with no user intervention. SPSS Modeler Batch is a special version of the product that provides support for the complete analytical capabilities of SPSS Modeler without access to the regular user interface. SPSS Modeler Server is required to use SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher is a tool that enables you to create a packaged version of an SPSS Modeler stream that can be run by an external runtime engine or embedded in an external application. In this way, you can publish and deploy complete SPSS Modeler streams for use in environments that do not have SPSS Modeler installed. SPSS Modeler Solution Publisher is distributed as part of the IBM SPSS Collaboration and Deployment Services - Scoring service, for which a separate license is required. With this license, you receive SPSS Modeler Solution Publisher Runtime, which enables you to execute the published streams.

For more information about SPSS Modeler Solution Publisher, see the IBM SPSS Collaboration and Deployment Services documentation. The IBM SPSS Collaboration and Deployment Services IBM Documentation contains sections called "IBM SPSS Modeler Solution Publisher" and "IBM SPSS Analytics Toolkit."

IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services

A number of adapters for IBM SPSS Collaboration and Deployment Services are available that enable SPSS Modeler and SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. In this way, an SPSS Modeler stream deployed to the repository can be shared by multiple users, or accessed from the thin-client application IBM SPSS Modeler Advantage. You install the adapter on the system that hosts the repository.

IBM SPSS Modeler Editions

SPSS Modeler is available in the following editions.

SPSS Modeler Professional

SPSS Modeler Professional provides all the tools you need to work with most types of structured data, such as behaviors and interactions tracked in CRM systems, demographics, purchasing behavior and sales data.

SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data and with unstructured text data. SPSS Modeler Premium includes IBM SPSS Modeler Text Analytics:

IBM SPSS Modeler Text Analytics uses advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data, extract and organize the key concepts, and group these concepts into categories. Extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling using the full suite of IBM SPSS Modeler data mining tools to yield better and more focused decisions.

IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription provides all the same predictive analytics capabilities as the traditional IBM SPSS Modeler client. With the Subscription edition, you can download product updates regularly.

Documentation

Documentation is available from the **Help** menu in SPSS Modeler. This opens the online IBM Documentation, which is always available outside the product.

Complete documentation for each product (including installation instructions) is also available in PDF format at <https://www.ibm.com/support/pages/spss-modeler-185-documentation>.

SPSS Modeler Professional Documentation

The SPSS Modeler Professional documentation suite (excluding installation instructions) is as follows.

- **IBM SPSS Modeler User's Guide.** General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services or IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler Source, Process, and Output Nodes.** Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- **IBM SPSS Modeler Modeling Nodes.** Descriptions of all the nodes used to create data mining models. IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Applications Guide.** The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. See the topic [“Application examples”](#) on page 4 for more information.
- **IBM SPSS Modeler Python Scripting and Automation.** Information on automating the system through Python scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running IBM SPSS Modeler streams as steps in processing jobs under IBM SPSS Deployment Manager.
- **IBM SPSS Modeler In-Database Mining Guide.** Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- **IBM SPSS Modeler Server Administration and Performance Guide.** Information on how to configure and administer IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager User Guide.** Information on using the administration console user interface included in the Deployment Manager application for monitoring and configuring IBM SPSS Modeler Server.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.
- **IBM SPSS Modeler Batch User's Guide.** Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments. This guide is available in PDF format only.

SPSS Modeler Premium Documentation

The SPSS Modeler Premium documentation suite (excluding installation instructions) is as follows.

- **SPSS Modeler Text Analytics User's Guide.** Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.

Application examples

While the data mining tools in SPSS Modeler can help solve a wide variety of business and organizational problems, the application examples provide brief, targeted introductions to specific modeling methods and techniques. The data sets used here are much smaller than the enormous data stores managed by some data miners, but the concepts and methods that are involved are scalable to real-world applications.

To access the examples, click **Application Examples** on the Help menu in SPSS Modeler.

The data files and sample streams are installed in the Demos folder under the product installation directory. For more information, see [“Demos Folder”](#) on page 4.

Database modeling examples. See the examples in the *IBM SPSS Modeler In-Database Mining Guide*.

Scripting examples. See the examples in the *IBM SPSS Modeler Scripting and Automation Guide*.

Demos Folder

The data files and sample streams that are used with the application examples are installed in the Demos folder under the product installation directory (for example: C:\Program Files\IBM\SPSS\Modeler\\Demos). This folder can also be accessed from the IBM SPSS Modeler program group on the Windows Start menu, or by clicking Demos on the list of recent directories in the **File > Open Stream** dialog box.

License tracking

When you use SPSS Modeler, license usage is tracked and logged at regular intervals. The license metrics that are logged are *AUTHORIZED_USER* and *CONCURRENT_USER*, and the type of metric that is logged depends on the type of license that you have for SPSS Modeler.

The log files that are produced can be processed by the IBM License Metric Tool, from which you can generate license usage reports.

The license log files are created in the same directory where SPSS Modeler Client log files are recorded (by default, %ALLUSERSPROFILE%/IBM/SPSS/Modeler/<version>/log).

Chapter 2. Architecture and Hardware Recommendations

IBM SPSS Modeler Architecture

This section describes the architecture of IBM SPSS Modeler Server, including the server software, the client software, and the database. It includes information about how IBM SPSS Modeler Server is designed for optimal performance and provides recommendations for maximizing this performance by choosing appropriately sized hardware. It concludes with a section on data access, which describes where to set up the necessary ODBC drivers.

Architecture Description

IBM SPSS Modeler Server uses a three-tier, distributed architecture. Software operations are shared between the client and the server computers. The advantages of installing and using IBM SPSS Modeler Server (versus the standalone IBM SPSS Modeler), especially when dealing with large data sets, are numerous:

- IBM SPSS Modeler Server can run on UNIX, in addition to Windows, allowing more flexibility in deciding where to install it. On any platform, you can dedicate a faster, larger server computer to data mining processes.
- IBM SPSS Modeler Server is optimized for fast performance. When operations cannot be pushed into the database, IBM SPSS Modeler Server stores the intermediate results as temporary files on disk rather than in RAM. Because servers usually have significant disk space available, IBM SPSS Modeler Server can perform sort, merge, and aggregation operations on very large data sets.
- Using the client-server architecture, you can centralize data-mining processes in your organization. Centralization can help to formalize the role of data mining in your business processes.
- Using administrator tools like the IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager) and IBM SPSS Collaboration and Deployment Services (sold separately), you can monitor data mining processes, ensuring that adequate computing resources are available. With IBM SPSS Collaboration and Deployment Services you can automate certain data mining tasks, manage access to data models, and share results across your organization.

The components of IBM SPSS Modeler's distributed architecture are shown in the "[IBM SPSS Modeler Server Architecture](#)" graphic.

- **IBM SPSS Modeler.** The client software is installed on the end user's computer. It provides the user interface and displays the data mining results. The client is a complete installation of IBM SPSS Modeler software, but when it is connected to IBM SPSS Modeler Server for distributed analysis, its execution engine is inactive. The IBM SPSS Modeler runs on Windows operating systems only.
- **IBM SPSS Modeler Server.** The server software installed on a server computer, with network connectivity to both the IBM SPSS Modeler(s) and the database. IBM SPSS Modeler Server runs as a service (on Windows) or a daemon process (on UNIX), waiting for clients to connect. It handles the execution of streams and scripts created using the IBM SPSS Modeler.
- **Database server.** The database server could be a live data warehouse (for example, Oracle on a large UNIX server) or, to reduce impact on other operational systems, a data mart on a local/departmental server (for example, SQL Server on Windows).

IBM SPSS Modeler Server Architecture

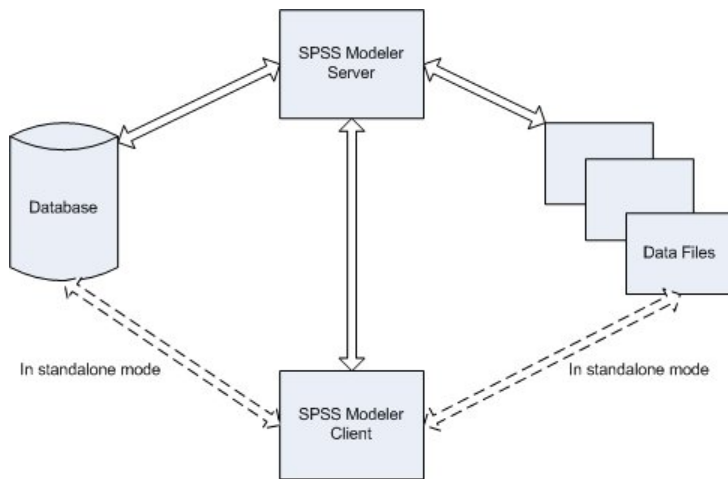


Figure 1. IBM SPSS Modeler Server architecture

With the distributed architecture, most of the processing occurs on the server computer. When the end user executes a stream, IBM SPSS Modeler sends a description of the stream to the server. The server determines which operations can be executed in SQL and creates the appropriate queries. These queries are executed in the database, and the resulting data are passed to the server for any processing that cannot be expressed using SQL. Once the processing is complete, only the relevant results are passed back to the client.

If necessary, IBM SPSS Modeler Server can execute all IBM SPSS Modeler operations outside of the database. It automatically balances its use of RAM and disk memory to hold data for manipulation. This process makes IBM SPSS Modeler Server fully compatible with flat files.

Load balancing is also available by using a cluster of servers for processing. Clustering is available starting in IBM SPSS Collaboration and Deployment Services 3.5 through the Coordinator of Processes plug-in. See the topic [Appendix E, “Load Balancing with Server Clusters,”](#) on page 87 for more information. You can connect to a server or cluster managed in the Coordinator of Processes directly through IBM SPSS Modeler's Server Login dialog. See the topic [“Connecting to IBM SPSS Modeler Server”](#) on page 11 for more information.

Standalone Client

IBM SPSS Modeler may also be configured to run as a self-contained desktop application, shown in the graphic below. See [Chapter 3, “IBM SPSS Modeler Support,”](#) on page 11 for more information.

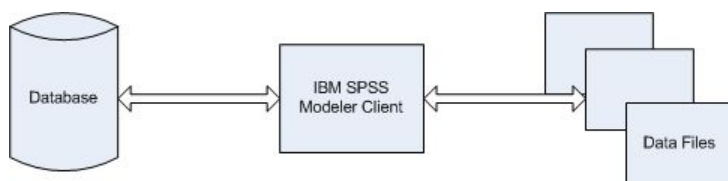


Figure 2. IBM SPSS Modeler standalone

Hardware Recommendations

As you plan your IBM SPSS Modeler Server installation, you should consider the hardware that you will use. Although IBM SPSS Modeler Server is designed to be speedy, you can maximize its efficiency by using hardware that is sized appropriately for your data mining tasks. Upgrading hardware is often the simplest and most economical way to improve performance across the board.

Dedicated server. Install IBM SPSS Modeler Server on a dedicated server machine where it will not compete for resources with other applications, including any databases to which IBM SPSS Modeler Server may be connecting. Model-building operations in particular are resource-intensive and perform much better when not in competition with other applications.

Note: Although installing IBM SPSS Modeler Server on the same computer as the database can reduce data-transfer time between the database and the server by avoiding network overhead, in most cases the best configuration is to have the server and database on separate machines to avoid competition for resources. Provide a fast connection between the two to minimize the cost of data transfer.

Processors. The number of processors on the machine should be no less than the number of concurrent tasks (simultaneously executing streams) you expect to run on a regular basis. In general, the more processors, the better.

- A single instance of IBM SPSS Modeler Server will accept connections from multiple clients (users), and each client connection can initiate multiple stream executions. One server can therefore have several execution tasks in progress at any one time.
- As a rule of thumb, allow one processor for one or two users, two processors for up to four users, and four processors for up to eight users. Add one additional processor for every two to four users beyond that, depending on the mix of work.
- To the extent that some processing may be pushed back to the database through SQL optimization, it may be possible to share a CPU between two or more users with minimal loss in performance.
- Multithreading capabilities make it possible for a single task to take advantage of multiple processors, so adding processors can improve performance even in cases where only one task is running at a time. Generally, multithreading is used for C5.0 model building and certain data preparation operations (sort, aggregate, and merge). Multithreading is also supported for all nodes that run in IBM SPSS Analytic Server (for example: GLE, Linear-AS, Random Forest, LSVM, Tree-AS, Time Series, TCM, Association Rules, and STP).

64-bit platforms. If you plan to process or build models on very large volumes of data, use a 64-bit machine as your IBM SPSS Modeler Server platform, and maximize the amount of RAM for the machine. For larger data sets, the server can quickly exhaust the per-process memory limits imposed by 32-bit platforms, forcing data to be spilled to disk and significantly increasing the running time. 64-bit server implementations can take advantage of additional RAM; a minimum of 8 gigabytes (GB) is recommended.

Future needs. Whenever feasible, make sure that server hardware is expandable in terms of memory and CPUs, both to accommodate increases in usage (for example, increased numbers of simultaneous users or increases in the existing users processing requirements) and increased multithreading capabilities of IBM SPSS Modeler Server in the future.

Temporary Disk Space and RAM Requirements

IBM SPSS Modeler Server uses temporary disk space to process large volumes of data. The amount of temporary space that you need depends on the volume and type of data that you process and the type of operations you perform. The data volume is proportional to both the number of rows *and* the number of columns. The more rows and columns that you process, the more disk space you need.

This section describes the conditions under which temporary disk space and extra RAM are required, and how to estimate the amount required. Note that this section does not discuss the temporary disk space requirements for processes that occur in a database, since these requirements are specific to each database.

Conditions That Require Temporary Disk Space

The powerful SQL optimization feature of IBM SPSS Modeler Server means that processing can occur in the database (rather than on the server) whenever possible. However, when any of the following conditions are true, SQL optimization cannot be used:

- The data to be processed are held in a flat file rather than in a database.
- SQL optimization is turned off.
- The processing operation cannot be optimized using SQL.

When SQL optimization cannot be used, the following data manipulation nodes and CLEM functions create temporary disk copies of some or all of the data. If the streams used at your site contain these processing commands or functions, you may need to set aside additional disk space on your server.

- Aggregate node
- Distinct node
- Binning node
- Merge node when using the merge-by-key option
- Any modeling node
- Sort node
- Table output node
- @OFFSET functions in which the lookup condition uses @THIS.
- Any @ function, such as @MIN, @MAX, and @AVE, in which the offset parameter is calculated.

Calculating the Amount of Temporary Disk Space

In general, IBM SPSS Modeler Server needs to be able to write a temporary file that is at least *three times as large* as the original data set. For example, if the data file is 2GB and SQL generation is not used, IBM SPSS Modeler Server will require 6GB of disk space to process the data. Because each concurrent user account creates its own temporary files, you will need to increase the disk space accordingly for each concurrent user.

If you find that your site frequently uses large temporary files, consider using a separate file system for IBM SPSS Modeler's temporary files, created on a separate disk. For best results, a RAID 0 or striped data set that spans multiple physical disks can be used to speed up disk operations, ideally with each disk in the striped file system on a separate disk controller.

RAM Requirements

For most processing that cannot be performed in the database, IBM SPSS Modeler Server stores the intermediate results as temporary files on disk rather than in memory (RAM). However, for modeling nodes, RAM is used if possible. The Neural Net, Kohonen, and K-Means nodes require large amounts of RAM. If these nodes are frequently used at your site, consider installing more RAM on the server.

In general, the number of bytes of RAM needed can be estimated by

$$(\text{number_of_records} * \text{number_of_cells_per_record}) * \text{number_of_bytes_per_cell}$$

where `number_of_cells_per_record` can become very large when there are nominal fields.

Refer to the system requirements section of the server installation guide for current RAM recommendations. For four or more simultaneous users, even more RAM is recommended. Memory must be shared between concurrent tasks, so scale up accordingly. In general, adding memory is likely to be one of the most cost-effective ways to improve performance across the board.

Data Access

To read or write to a database, you must have an ODBC data source that is installed and configured for the relevant database, with read or write permissions as needed. The IBM SPSS Data Access Pack includes a set of ODBC drivers that can be used for this purpose, and these drivers are available from the download site. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Supported ODBC drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to install drivers

Note: ODBC drivers must be installed and configured on each computer where processing might occur.

- If you are running IBM SPSS Modeler in local (standalone) mode, the drivers must be installed on the local computer.
- If you are running IBM SPSS Modeler in distributed mode against a remote IBM SPSS Modeler Server, the ODBC drivers need to be installed on the computer where IBM SPSS Modeler Server is installed. For IBM SPSS Modeler Server on UNIX systems, see also "Configuring ODBC drivers on UNIX systems" later in this section.
- If you need to access the same data sources from both IBM SPSS Modeler and IBM SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running IBM SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have IBM SPSS Modeler installed.

Configuring ODBC drivers on UNIX systems

By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Server on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands:

```
cd <modeler_server_install_directory>/bin
rm -f libspssodbc.so
```

Then run this command if you want to use the UTF8 driver wrapper:

```
ln -s libspssodbc_datadirect.so libspssodbc.so
```

Or run this command instead if you want to use the UTF16 driver wrapper:

```
ln -s libspssodbc_datadirect_utf16.so libspssodbc.so
```

Doing so removes the default link and creates a link to the DataDirect Driver Manager.

Note: The UTF16 driver wrapper is required to use SAP HANA or IBM Db2 CLI drivers for some databases. DashDB requires the IBM Db2 CLI driver.

To configure SPSS Modeler Server:

1. Configure the SPSS Modeler Server start up script `modelersrv.sh` to source the IBM SPSS Data Access Pack `odbc.sh` environment file by adding the following line to `modelersrv.sh`:

```
. /<pathtoSDAPinstall>/odbc.sh
```

Where `<pathtoSDAPinstall>` is the full path to your IBM SPSS Data Access Pack installation.

2. Restart SPSS Modeler Server.

In addition, for SAP HANA and IBM Db2 only, add the following parameter definition to the DSN in your `odbc.ini` file to avoid buffer overflows during connection:

```
DriverUnicodeType=1
```

Note: The `libspssodbc_datadirect_utf16.so` wrapper is also compatible with the other SPSS Modeler Server supported ODBC drivers.

Note: The above rules apply specifically to accessing data in a database. Other types of file operations, such as opening and saving streams, projects, models, nodes, PMML, output, and script files, are always done on the client and are always specified in terms of the file system of the client computer. In addition, the Set Directory command in SPSS Modeler sets the working directory for *local* client objects (for example, streams) but does not affect the server's working directory.

UNIX and SPSS Statistics

For information about how to configure SPSS Modeler Server on UNIX to work with the IBM SPSS Statistics data access technology, see [Appendix B, “Configuring UNIX Startup Scripts,”](#) on page 73.

Referencing Data Files

Windows. If you store data on the same computer as IBM SPSS Modeler Server, we recommend that you give the path to the data from the perspective of the server computer (for example, `C:\ServerData\Sales 1998.csv`). Performance is faster when the network is not used to locate the file.

If the data is stored on a different host, we recommend using UNC file references (for example, `|mydataserver\ServerData\Sales 1998.csv`). Note that UNC names work only when the path contains the name of a shared network resource. The referencing computer must have permission to read the specified file. If you switch frequently from distributed to local analysis mode, use UNC file references because they work regardless of the mode.

UNIX. To reference data files that reside on a UNIX server, use the full file specification and forward slashes (for example, `/public/data/ServerData/Sales 1998.csv`). Avoid using the backslash character in the UNIX directory and in filenames for data used with IBM SPSS Modeler Server. It does not matter whether a text file uses UNIX or DOS format—both are handled automatically.

Importing IBM SPSS Statistics Data Files

If you are also running IBM SPSS Statistics Server at your site, users may want to import or export IBM SPSS Statistics data while in distributed mode. Recall that when the IBM SPSS Modeler runs in distributed mode, it presents the server's file system. The IBM SPSS Statistics client works in the same way. For importing and exporting to take place between the two applications, both clients must be operating in the same mode. If they are not, their views of the file systems will be different and they will not be able to share files. The IBM SPSS Statistics nodes in IBM SPSS Modeler can automatically start the IBM SPSS Statistics client, but users must first ensure that the IBM SPSS Statistics client is operating in the same mode as IBM SPSS Modeler.

Installation Instructions

For information on installing IBM SPSS Modeler Server, see the installation instructions which are available as PDF files as part of your product download.. Separate documents are available for Windows and UNIX.

For complete information on installing and using the IBM SPSS Modeler client, see the PDF files which are available as part of your product download. Separate installation documents are available, depending on the type of license you have.

Chapter 3. IBM SPSS Modeler Support

This section is intended for administrators and help-desk personnel who support users of IBM SPSS Modeler. It covers the following topics:

- How to log on to IBM SPSS Modeler Server (or run standalone by disconnecting from a Server)
- Data and file systems that users may need
- User accounts and file permissions pertaining to IBM SPSS Modeler Server
- Differences in results that users may see when switching between IBM SPSS Modeler Server and IBM SPSS Modeler

Connecting to IBM SPSS Modeler Server

IBM SPSS Modeler can be run as a standalone application, or as a client connected to IBM SPSS Modeler Server directly or to an IBM SPSS Modeler Server or server cluster through the Coordinator of Processes plug-in from IBM SPSS Collaboration and Deployment Services. The current connection status is displayed at the bottom left of the IBM SPSS Modeler window.

Whenever you want to connect to a server, you can manually enter the server name to which you want to connect or select a name that you have previously defined. However, if you have IBM SPSS Collaboration and Deployment Services, you can search through a list of servers or server clusters from the Server Login dialog box. The ability to browse through the Statistics services running on a network is made available through the Coordinator of Processes.

To Connect to a Server

1. On the Tools menu, click **Server Login**. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
2. Using the dialog box, specify options to connect to the local server computer or select a connection from the table.
 - Click **Add** or **Edit** to add or edit a connection. See the topic [“Adding and Editing the IBM SPSS Modeler Server Connection”](#) on page 17 for more information.
 - Click **Search** to access a server or server cluster in the Coordinator of Processes. See the topic [“Searching for Servers in IBM SPSS Collaboration and Deployment Services ”](#) on page 18 for more information.

Server table. This table contains the set of defined server connections. The table displays the default connection, server name, description, and port number. You can manually add a new connection, as well as select or search for an existing connection. To set a particular server as the default connection, select the check box in the Default column in the table for the connection.

Default data path. Specify a path used for data on the server computer. Click the ellipsis button (...) to browse to the required location.

Set Credentials. Leave this box unchecked to enable the **single sign-on** feature, which attempts to log you in to the server using your local computer username and password details. If single sign-on is not possible, or if you check this box to disable single sign-on (for example, to log in to an administrator account), the following fields are enabled for you to enter your credentials.

User ID. Enter the user name with which to log on to the server.

Password. Enter the password associated with the specified user name.

Domain. Specify the domain used to log on to the server. A domain name is required only when the server computer is in a different Windows domain than the client computer.

3. Click **OK** to complete the connection.

To Disconnect from a Server

1. On the Tools menu, click **Server Login**. The Server Login dialog box opens. Alternatively, double-click the connection status area of the IBM SPSS Modeler window.
2. In the dialog box, select the Local Server and click **OK**.

Configuring single sign-on

You can connect to an IBM SPSS Modeler Server that is running on any supported platform using Single Sign-On. To connect using Single Sign-On, you must first configure your IBM SPSS Modeler server and client machines.

If you are using Single Sign-On to connect to both IBM SPSS Modeler Server and IBM SPSS Collaboration and Deployment Services, you must connect to IBM SPSS Collaboration and Deployment Services before you connect to IBM SPSS Modeler.

IBM SPSS Modeler Server uses Kerberos for Single Sign-On.

Kerberos is a core component of Windows Active Directory, and the following information assumes an Active Directory infrastructure. In particular:

- The client computer is a Windows computer that is joined to an Active Directory domain
- The client user has logged in to the computer using a domain account. The mechanism used to log in is unimportant and may employ a smart card, fingerprint, etc.
- IBM SPSS Modeler Server can validate the client user's credentials by reference to the Active Directory domain controller

This documentation describes how both Windows and UNIX servers can be configured to authenticate this way. Other configurations may be possible but are untested.

To inter-operate with most modern, secure Active Directory installations, you must install the high-strength encryption pack for Java because the required encryption algorithms are not supported by default. You must install the pack for both client and server. An error message such as `Illegal key size` is displayed on the client when a server connection fails because the pack is not installed. See [“Installing unlimited strength encryption”](#) on page 45.

The Service Principal Name

Each server instance must register a unique *service principal name (SPN)* to identify itself, and the client must specify the same SPN when it connects to the server.

An SPN for an instance of SPSS Modeler Server has the form:

```
modelerserver/<host>:<port>
```

For example:

```
modelerserver/jdoemachine.spss.com:28054
```

Note that the host name must be qualified with its DNS domain (`spss.com` in this example), and the domain must map to the Kerberos realm.

The combination of host name and port number makes the SPN unique (because each instance on a given host must listen on a different port). And both client and server already have the host name and port number and so can construct the appropriate SPN for the instance. The additional configuration step required is to register the SPN in the Kerberos database.

Registering the SPN on Windows

If you are using Active Directory as your Kerberos implementation, use the `setspn` command to register the SPN. To run this command, the following conditions must be satisfied:

- You must be logged on to a domain controller
- You must run the command prompt with elevated privileges (run as administrator)
- You must be a member of the Domain Admins group (or have had the appropriate permission delegated to you by a domain administrator)

For more information, refer to the following articles:

- [Setspn Command-Line Reference](#)
- [Delegating Authority to Modify SPNs](#)

For the default instance, listening on the standard port (28054, for example) and running under the Local System account, you must register the SPN against the server computer name. For example:

```
setspn -s modelerserver/jdoemachine.spss.com:28054 jdoemachine
```

For each subsequent (profile) instance, listening on a custom port (for example, 29000) and running under an arbitrary user account (for example, jdoe) with the option `start_process_as_login_user` set to Y, you must register the SPN against the service user account name:

```
setspn -s modelerserver/jdoemachine.spss.com:29000 jdoe
```

Note that in this case (when the service account is other than Local System), registering the SPN is not sufficient to enable a client to connect. Additional configuration steps are described in the next section.

To see which SPNs are registered to the account jdoe:

```
setspn -l jdoe
```

Registering the SPN on UNIX

If you are using Active Directory as your Kerberos implementation, you can use the `setspn` command as described in the previous Windows section; this assumes you have already created the computer or user account in the directory. Or you can use `ktpass`, as illustrated in [“Configuring IBM SPSS Modeler Server on UNIX and Linux” on page 14](#).

If you are using some other Kerberos implementation, then use the Kerberos administration tool to add the service principal to the Kerberos database. To convert the SPN to a Kerberos principal you must append the name of the Kerberos realm. For example:

```
modelerserver/jdoemachine.spss.com:28054@MODELERSSO.COM
```

Add this same principal and password to the server's keytab. The keytab must contain an entry for every instance running on the host.

Configuring IBM SPSS Modeler Server on Windows

In the default scenario where the SPSS Modeler Server service runs under the Local System account, it uses native Windows APIs to authenticate the user's credentials and no additional configuration is required on the server.

In the alternative scenario where the SPSS Modeler Server service runs under a dedicated user account and `start_process_as_login_user` is set to Y, then it uses Java APIs to authenticate the user's credentials and additional configuration is required on the server.

First, verify that the default scenario works. The client should be able to use SSO to connect to the default instance running under the Local System account. This will validate the client-side configuration (that is unchanged). You will need to register the SPN for the default instance as described earlier.

Then perform the following steps:

1. Create the directory `<MODELERSERVER>\config\sso`.

2. Create a file called `krb5.conf` in the `sso` folder you created in step 1. For instructions on how to create this file, see step 3 under [“Configuring IBM SPSS Modeler client”](#) on page 15. The file must be the same on the server and client.
3. Use the following command to create the file `krb5.keytab` in the server SSO directory:

```
<MODELERSERVER>\jre\bin\ktab -a <spn>@<realm> -k krb5.keytab
```

For example:

```
".\jre\bin\ktab.exe" -a modelerserver/  
jdoemachine.spss.com:29000@SPSS.COM  
-k krb5.keytab
```

This will prompt you for a password. The password you enter must be the password of the service account. So if the service account is `jdoe`, for example, you must enter the password for the user `jdoe`.

The service account itself is not mentioned in the keytab, but earlier you registered the SPN to that account using `setspn`. This means that the password for the service principal and the password for the service account are one and the same.

For each new instance (profile) you create, you must register the SPN for that instance (using `setspn`; see [“Configuring server profiles”](#) on page 23 and [“The Service Principal Name”](#) on page 12) and add an entry to the keytab (using `jre\bin\ktab`). There is only one keytab file, and it must contain an entry for every instance that is not running as Local System. The default instance, or any other instance running as Local System, does not need to be in the keytab because it uses Windows APIs to authenticate. Windows APIs do not use the keytab.

To verify that an instance is included in the keytab:

```
ktab.exe -l -e -k krb5.keytab
```

You may see multiple entries for each principal with different encryption types, but this is normal.

Configuring IBM SPSS Modeler Server on UNIX and Linux

Prerequisites

IBM SPSS Modeler Server relies on Windows Active Directory (AD) to enable single sign-on, for which the following prerequisites are essential:

- The SPSS Modeler Client (Windows) computer is a member of an Active Directory (AD) domain.
- The client user logs in to the computer using an AD domain account.
- The SPSS Modeler Server (UNIX) computer is identified by a fully-qualified domain name that is rooted in the AD DNS domain. For example, if the DNS domain is `modelersso.com`, then the server hostname might be `myserver.modelersso.com`.
- The AD DNS domain supports both forward and reverse lookups for the SPSS Modeler Server hostname.

If the SPSS Modeler Server machine is not a member of the AD domain you must create a domain user account to represent the service in the directory. For example, you could create a domain account called `ModelerServer`.

To Configure SPSS Modeler Server on UNIX or Linux

1. In the SPSS Modeler Serverconfig folder, create a subfolder called `sso`.
2. In the `sso` folder, create a keytab file. The keytab file's generation can be done on the AD side; however, there are different requirements depending on whether the SPSS Modeler Server machine is a member of the AD domain:

- If the SPSS Modeler Server machine **is** a member of the AD domain, use the computer account name as the service user name:

```
ktpass -princ <spn>@<realm> -mapUser <domain>\<computer account> -pass <password> -out <output file> -ptype KRB5_NT_PRINCIPAL
```

For example:

```
ktpass -princ modelerserver/myserver.modelersso.com:28054@MODELERSSO.COM -mapUser MODELERSSO\myserver$ -pass Pass1234 -out c:\myserver.keytab -ptype KRB5_NT_PRINCIPAL
```

- If the SPSS Modeler Server machine **is not** a member of the AD domain, specify the domain user account, that you created as a prerequisite, as the service user:

```
ktpass -princ <spn>@<realm> -mapUser <domain>\<user account> -mapOp set -pass <password> -out <output file> -ptype KRB5_NT_PRINCIPAL
```

For example:

```
ktpass -princ modelerserver/myserver.modelersso.com:28054@MODELERSSO.COM -mapUser MODELERSSO\
ModelerServer -mapOp
set -pass Pass1234 -out c:\myserver.keytab -ptype KRB5_NT_PRINCIPAL
```

For more information, see [Ktpass Command-Line Reference](#).

3. Rename the keytab file in the sso folder to `krb5.keytab`.

Note: If you re-join the server machine to the domain, generate a new keytab file.

4. Create a file called `krb5.conf` in the sso folder you created in step 1. For instructions on how to create this file, see step 3 under [“Configuring IBM SPSS Modeler client”](#) on page 15. The file must be the same on the server and client.

Configuring IBM SPSS Modeler client

1. Enable Java to access the TGT session key:

- a. From the **Start** menu, click **Run**.
- b. Enter `regedit` and click **OK** to open the **Registry Editor**.
- c. Navigate to the registry location appropriate to the operating system of the local machine:
 - On Windows XP: My Computer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos
 - On Windows Vista, or Windows 7: My Computer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
- d. Right-click the folder and select **New > DWORD**. The name of the new value should be `allowtgtsessionkey`.
- e. Set the value of `allowtgtsessionkey` to a hexadecimal value of 1, that is 0x0000001.
- f. Close the **Registry Editor**.
- g. Note there is a known issue when the user account is a member of the local Administrators group and User Account Control (UAC) is enabled. In this case, the session key in the retrieved service ticket is empty, which causes SSO authentication to fail. To avoid this issue, perform one of the actions:
 - Run the application as Administrator
 - Disable User Account Control
 - Use an account that is not an Administrator account

2. In the `config` folder of the IBM SPSS Modeler installation location, create a folder called `sso`.

3. In the sso folder, create a krb5.conf file. Instructions for how to create a krb5.conf file can be found at http://web.mit.edu/kerberos/krb5-current/doc/admin/conf_files/krb5_conf.html. An example of a krb5.conf file is provided below:

```
[libdefaults]
    default_realm = MODELERSSO.COM
    dns_lookup_kdc = true
    dns_lookup_realm = true

[realms]
    MODELERSSO.COM = {
        kdc = ad.modelersso.com:88
        admin_server = ad.modelersso.com:749
        default_domain = modelersso.com
    }

[domain_realm]
    .modelersso.com = MODELERSSO.COM
    modelersso.com = MODELERSSO.COM
```

4. Restart the local machine and the server machine.

Getting the SSO user's group membership

When a user logs on to SPSS Modeler Server using SSO and the server is running non-root, then the name of the authenticated user is not associated with an operating system user account. The server cannot obtain the user's operating system group membership. So how is group configuration performed in this case?

We assume the user is registered in an LDAP directory (which could be Active Directory) and we can request the group membership from the LDAP server. SPSS Modeler Server can query the LDAP provider in IBM SPSS Collaboration and Deployment Services for the group membership.

There are two properties in options.cfg on the SPSS Modeler Server that control the server's access to the IBM SPSS Collaboration and Deployment Services Repository:

```
repository_enabled, N
repository_url, ""
```

To enable group lookup, you must set both properties. For example:

```
repository_enabled, Y
repository_url, "http://jdoemachine.spss.ibm.com:9083"
```

The repository connection is only used for SSO group lookup, so you do not need to change these property settings unless you need this feature.

For group lookup to work properly, you must configure your repository first to add an LDAP or Active Directory provider and then to enable SSO using that provider:

1. Start IBM SPSS Deployment Manager client and select **File > New > Administered Server Connection...** to create an administered server connection for your repository (if you do not have one already).
2. Log on to the administered server connection and expand the **Configuration** folder.
3. Right-click **Security Providers**, choose **New > Security provider definition...**, and enter the appropriate values. Click **Help** in the dialog for more information.
4. Expand the **Single Sign-On Providers** folder, right-click **Kerberos SSO Provider**, and select **Open**.
5. Click **Enable**, select your security provider, and then click **Save**. You do not have to fill in any other details here unless you want to use SSO (simply having the provider enabled is sufficient to allow the group lookup).

Important: For group lookup to work properly, the Kerberos provider you configure here must be the same as the provider you configured for SPSS Modeler Server. In particular, they must be working within the same Kerberos realm. So if a user logs on to SPSS Modeler Server using SSO and it identifies

him as `jdoe@SPSS.COM` (where `SPSS.COM` is the realm), it will expect the security provider in IBM SPSS Collaboration and Deployment Services to recognize that user principal name and return the corresponding group membership from the LDAP directory.

Single sign-on for data sources

You can connect to databases from IBM SPSS Modeler using single sign-on. If you want to create a database connection using single sign-on, you must first use your ODBC management software to properly configure a data source and single sign-on token. Then when connecting to a database in IBM SPSS Modeler, IBM SPSS Modeler will use that same single sign-on token and the user will not be prompted to log on to the data source.

However, if the data source was not configured properly for single sign-on, IBM SPSS Modeler will prompt the user to log on to the data source. The user will still be able to access the data source after providing valid credentials.

For complete details about configuring ODBC data sources on your system with single sign-on enabled, see your database vendor documentation. Following is an example of the general steps that may be involved:

1. Configure your database so it can support Kerberos single sign-on.
2. On the IBM SPSS Modeler Server machine, create an ODBC data source and test it. The DSN connection should not require a user ID and password.
3. Connect to IBM SPSS Modeler Server using single sign-on and begin using the ODBC data source created and validate in step 2.

Adding and Editing the IBM SPSS Modeler Server Connection

You can manually edit or add a server connection in the Server Login dialog box. By clicking Add, you can access an empty Add/Edit Server dialog box in which you can enter server connection details. By selecting an existing connection and clicking Edit in the Server Login dialog box, the Add/Edit Server dialog box opens with the details for that connection so that you can make any changes.

Note: You cannot edit a server connection that was added from IBM SPSS Collaboration and Deployment Services, since the name, port, and other details are defined in IBM SPSS Collaboration and Deployment Services. Best practice dictates that the same ports should be used to communicate with both IBM SPSS Collaboration and Deployment Services and SPSS Modeler Client. These can be set as `max_server_port` and `min_server_port` in the `options.cfg` file.

To Add Server Connections

1. On the Tools menu, click **Server Login**. The Server Login dialog box opens.
 2. In this dialog box, click **Add**. The Server Login Add/Edit Server dialog box opens.
 3. Enter the server connection details and click **OK** to save the connection and return to the Server Login dialog box.
- **Server.** Specify an available server or select one from the list. The server computer can be identified by an alphanumeric name (for example, *myserver*) or an IP address assigned to the server computer (for example, 202.123.456.78).
 - **Port.** Give the port number on which the server is listening. If the default does not work, ask your system administrator for the correct port number.
 - **Description.** Enter an optional description for this server connection.
 - **Ensure secure connection (use SSL).** Specifies whether an SSL (**Secure Sockets Layer**) connection should be used. SSL is a commonly used protocol for securing data sent over a network. To use this feature, SSL must be enabled on the server hosting IBM SPSS Modeler Server. If necessary, contact your local administrator for details.

To Edit Server Connections

1. On the Tools menu, click **Server Login**. The Server Login dialog box opens.

2. In this dialog box, select the connection you want to edit and then click **Edit**. The Server Login Add/Edit Server dialog box opens.
3. Change the server connection details and click **OK** to save the changes and return to the Server Login dialog box.

Searching for Servers in IBM SPSS Collaboration and Deployment Services

Instead of entering a server connection manually, you can select a server or server cluster available on the network through the Coordinator of Processes, available in IBM SPSS Collaboration and Deployment Services. A server cluster is a group of servers from which the Coordinator of Processes determines the server best suited to respond to a processing request.

Although you can manually add servers in the Server Login dialog box, searching for available servers lets you connect to servers without requiring that you know the correct server name and port number. This information is automatically provided. However, you still need the correct login information, such as username, domain, and password.

Note: If you do not have access to the Coordinator of Processes capability, you can still manually enter the server name to which you want to connect or select a name that you have previously defined. See the topic [“Adding and Editing the IBM SPSS Modeler Server Connection”](#) on page 17 for more information.

To search for servers and clusters

1. On the Tools menu, click **Server Login**. The Server Login dialog box opens.
2. In this dialog box, click **Search** to open the Search for Servers dialog box. If you are not logged on to IBM SPSS Collaboration and Deployment Services when you attempt to browse the Coordinator of Processes, you will be prompted to do so.
3. Select the server or server cluster from the list.
4. Click **OK** to close the dialog box and add this connection to the table in the Server Login dialog box.

Data and File Systems

Users working with IBM SPSS Modeler Server will probably need to access data files and other data sources on the network, as well as save files on the network. They may need the following information, as applicable:

- **ODBC data source information.** If users need access to ODBC data sources defined on the server computer, they will need the names, descriptions, and login information (including database login IDs and passwords) for the data sources.
- **Data file access.** If users need to access data files on the server computer or elsewhere on the network, they will need the names and locations of the data files.
- **Location for saved files.** When users save data while connected to IBM SPSS Modeler Server, they may attempt to save files on the server computer. However, this is often a write-protected location. If so, let users know where they should save data files. (Typically, the location is the user's home directory.)

User Authentication

IBM SPSS Modeler Server uses the operating system on the server machine to authenticate users who connect to the server. When a user connects to SPSS Modeler Server, all operations that are performed on behalf of the user are performed in the user's security context. Access to database tables is subject to user and/or password privileges in the database itself.

Windows. On Windows, any user with a valid account on the host network can log on. With the default authentication, users must have modify access rights to the `<modeler_server_install>\Tmp` directory. Without these rights, users cannot log on to SPSS Modeler Server from the client using the default authentication on Windows.

UNIX. By default, SPSS Modeler Server is assumed to run as root on UNIX. This allows any user with a valid account on the host network to log on and limits users' file access to their own files and directories.

However, you can configure SPSS Modeler Server to run without root privilege. In this case, you must create a private password database to be used for authentication, and all SPSS Modeler users must share a single UNIX user account (and, consequently, share access to data files). For more information, see [“Configuring as non-root using a private password database”](#) on page 82.

Configuring PAM

On the Linux platform, SPSS Modeler Server uses the Pluggable Authentication Module (PAM) for authentication.

To use PAM authentication the appropriate PAM modules must be correctly configured on the host system; for example, for PAM to interface with LDAP, a PAM LDAP module must exist on the host OS and be correctly configured. Refer to the operating system documentation for further information. This is a prerequisite for SPSS Modeler Server to be able to use PAM.

To configure SPSS Modeler Server to use PAM, edit the SPSS Modeler Server "options.cfg" file and add (or edit) the line **authentication_methods, pam**.

You can use the service name `modelerserver` to provide a specific PAM configuration for SPSS Modeler Server if required. For example, the following steps explain how to configure for Red Hat Linux:

1. Change to the PAM configuration directory. For example: `/etc/pam.d`.
2. Using a text editor, create a new file called "modelerserver".
3. Add the PAM configuration information that you want to use. For example:

```
auth      include      system-auth
account   include      system-auth
password  required     pam_deny.so
session   required     pam_deny.so
```

Note: These lines might vary depending on your particular configuration. For more information, see the Linux documentation.

4. Save the file and restart the Modeler service.

Permissions

Windows. A user connecting to server software that is installed on an *NTFS* drive must login with an account that has the following permissions.

- Read and execute permissions to the server’s installation directory and its subdirectories
- Read, execute, and write permissions to the directory location for temporary files.

In Windows Server 2008 and later, you cannot assume that users have these permission. Be sure to explicitly set permissions as needed.

If the server software is installed on a *FAT* drive, you do not need to set permissions because all files allow users to have full control.

UNIX. If you are not using internal authentication, a user connecting to the server software must login with an account that has the following permissions:

- Read and execute permissions to the server’s installation directory and its subdirectories
- Read, execute, and write permissions to the directory location for temporary files.

File Creation

When IBM SPSS Modeler Server accesses and processes data, it often has to keep a temporary copy of that data on disk. The amount of disk space that will be used for temporary files depends on the size of the data file that the end user is analyzing and the type of analysis that he or she is performing. See the topic [“Temporary Disk Space and RAM Requirements”](#) on page 7 for more information.

UNIX. The UNIX versions of IBM SPSS Modeler Server use the UNIX umask command to set file permissions for the temporary files. You can override the server's default permissions. See the topic [“Controlling Permissions on File Creation”](#) on page 74 for more information.

Differences in Results

Users who run analyses in both modes may see slight differences in the results between IBM SPSS Modeler and IBM SPSS Modeler Server. The discrepancy usually occurs because of record ordering or rounding differences.

Record ordering. Unless a stream explicitly orders records by sorting them, the order in which records are presented may vary between streams executed locally and those executed on the server. There may also be differences in order between operations run within a database and those run in IBM SPSS Modeler Server. These differences are due to the different algorithms used by each system to implement functions that may reorder records, such as aggregation. Also, note that SQL does not specify the order in which records are returned from a database in cases where there is no explicit ordering operation.

Rounding differences. IBM SPSS Modeler running in local mode uses a different internal format for storing floating point values than does IBM SPSS Modeler Server. Due to rounding differences, results might vary slightly between each version.

Chapter 4. IBM SPSS Modeler Administration

This chapter contains information about starting and stopping IBM SPSS Modeler Server, configuring various server options, configuring groups, and interpreting the log file. It describes how to use IBM SPSS Modeler Administration Console, an application that facilitates server configuration and monitoring. For installation instructions for this component, refer to the installation instructions for IBM SPSS Modeler Server, available with that product.

Starting and Stopping IBM SPSS Modeler Server

IBM SPSS Modeler Server runs as a service on Windows or as a daemon process on UNIX.

Scheduling note: Stopping IBM SPSS Modeler Server disconnects end users and terminates their sessions, so try to schedule server restarts during periods of low usage. If this is not possible, be sure to notify users before stopping the server.

To Start, Stop, and Check Status on Windows

On Windows, you control IBM SPSS Modeler Server with the Services dialog box in the Windows Control Panel.

1. **Windows XP.** Open the Windows Start menu. Choose **Settings** and then **Control Panel**. Double-click **Administrative Tools** and then **Services**.

Windows 2003 or 2008. Open the Windows Start menu. Choose **Control Panel**, then **Administrative Tools**, then **Services**.

2. Select the **IBM SPSS Modeler Server <nn.n>** service. You can now check its status, start or stop it, and edit startup parameters, as appropriate.

By default, the service is configured for automatic startup, which means that if you stop it, it will restart automatically when the computer is rebooted. When started this way, the service runs unattended, and the server computer can be logged off without affecting it.

To Start, Stop, and Check Status on UNIX

On UNIX, you start or stop IBM SPSS Modeler Server by running the *modelersrv.sh* script in the IBM SPSS Modeler Server installation directory.

1. Change to the IBM SPSS Modeler Server installation directory. For example, at a UNIX command prompt, type

```
cd /usr/modelersrv
```

where *modelersrv* is the IBM SPSS Modeler Server installation directory.

2. To start the server, at the command prompt, type

```
./modelersrv.sh start
```

3. To stop the server, at the command prompt, type

```
./modelersrv.sh stop
```

4. To check the status of IBM SPSS Modeler Server, at a UNIX command prompt, type

```
./modelersrv.sh list
```

and look at the output, which is similar to what the UNIX `ps` command produces. The first process in the list is the IBM SPSS Modeler Server daemon process, and remaining processes are IBM SPSS Modeler sessions.

The IBM SPSS Modeler Server installation program includes a script (*auto.sh*) that configures your system to start the server daemon automatically at boot time. If you have run that script and then stop the server, the server daemon will restart automatically when the computer is rebooted. See the topic [“Automatically Starting and Stopping IBM SPSS Modeler Server”](#) on page 73 for more information.

UNIX kernel limits

You must ensure that kernel limits on the system are sufficient for the operation of IBM SPSS Modeler Server. The data, memory, file, and processes ulimits are particularly important and should be set to unlimited within the IBM SPSS Modeler Server environment. To do this:

1. Add the following commands to `modelersrv.sh`:

```
ulimit -d unlimited
```

```
ulimit -m unlimited
```

```
ulimit -f unlimited
```

```
ulimit -u unlimited
```

In addition, set the stack limit to the maximum allowed by your system (`ulimit -s XXXX`), for example:

```
ulimit -s 64000
```

2. Restart IBM SPSS Modeler Server.

Handling Unresponsive Server Processes (UNIX Systems)

IBM SPSS Modeler Server processes may become unresponsive for several reasons, including situations where they make a system call or ODBC driver call that becomes blocked (call never returns, or takes a very long time to return). When UNIX processes enter this state, they can be cleaned up using the UNIX `kill` command (interrupts initiated by IBM SPSS Modeler client, or the closing of IBM SPSS Modeler client, will have no effect). A `kill` command is provided as an alternative to the normal `stop` command, and enables an administrator to use *modelersrv.sh* to easily issue the appropriate `kill` command.

On systems which are susceptible to the accumulation of unusable (“zombie”) server processes, we recommend that IBM SPSS Modeler Server is stopped and restarted at regular intervals, using the following sequence of commands:

```
cd modeler_server_install_directory
./modelersrv.sh stop
./modelersrv.sh kill
```

Those IBM SPSS Modeler processes that are ended using the `modelersrv.sh kill` command will leave behind temporary files (from the temporary directory) that will need to be removed manually. Temporary files may be left behind in some other situations too, including application crashes due to resource exhaustion, user interrupts, system crashes, or other reasons. Therefore we recommend that, as part of the process of restarting IBM SPSS Modeler Server at regular intervals, all remaining files are removed from the IBM SPSS Modeler temporary directory.

Once all server processes have been closed and temporary files have been removed, IBM SPSS Modeler Server can be safely restarted.

Configuring server profiles

Server profiles allow you to run multiple, independent instances of SPSS Modeler Server from a single installation. To a client, they will appear to be separate servers located on the same host but listening on different port numbers. Having multiple instances sharing one installation benefits administrators because it simplifies maintenance. Subsequent instances after the first can be created and deleted more quickly than would be required for a full installation and uninstallation, and Fix Packs only need be applied once.

The reason for running multiple server instances on the same host is to be able to configure each instance separately. If all the instances are identical, there is nothing to be gained. In particular, if the instances run *non-root* (so that all sessions share the same user account), each instance can use a different user account to provide data isolation between user groups. For example, a user logging in to an instance A will be allocated a session owned by some particular `User-A` and will have access only to that user's files and folders, whereas a user logging in to instance B will see a different set of files and folders accessible to `User-B`. This can be used in conjunction with group configuration so that logging on to a particular instance is restricted to specific groups, meaning that end users can only log on to the instance (or instances) appropriate to their role. See [“Configuring groups” on page 47](#).

In a standard SPSS Modeler Server installation, the folders `config`, `data`, and `tmp` are specific to a server instance. The purpose of the `config` folder is for the instance to have a private configuration, and the `data` and `tmp` folders support data isolation. Each instance has a private copy of these folders, and everything else is shared.

Note that much of the server configuration can remain common (database settings, for example), so a profile configuration will override the common configuration. The server will look first in the profile configuration and then fall back to the default. The files that are most likely to be changed for a profile are options, groups, and passwords.

See [“Profile structure” on page 25](#) for more information.

For information on how to configure a profile to use SSO, see [“Configuring single sign-on” on page 12](#). This requires you to register a Service Principal Name (SPN), perform some configuration if the Windows Service account is not local, and in some cases enable group lookup.

Working with server profiles

Following are some common use cases for server profiles. Some of these uses are supported via the use of scripts (see [“Profile scripts” on page 27](#)) and may require administrative/root privileges.

Creating a server profile

An SPSS Modeler Server administrator named Jane uses a script to create a new profile:

- Jane must specify a unique name for the profile (it cannot be an existing profile name). If the `profiles` directory does not already exist, it is created for Jane. Then a new sub-directory is created in the `profiles` directory with name Jane specified, containing the directories `config`, `data`, `log`, and `tmp`.
- If Jane chooses, she can also specify the name of an existing profile to use as a template, in which case the content of the `config` folder within the existing profile is copied to the new profile. If she does not specify a template, or if the existing profile does not include an options file even though it should, then an empty options file is created in the new profile.
- Jane may also choose to specify a port number for the profile, in which case the port number is written as the value of the `port_number` property in the profile's options file. If she does not specify a port number, a value is chosen for her and written to the options file.
- Jan can also choose to specify the name of an operating system group that will have exclusive access to the profile in which case group configuration is enabled in the options file. In this case, a `groups` file is created that denies login to all but the specified group.

Configuring a server profile

Server administrator Jane configures a profile either by manually editing the profile configuration files, or by using the IBM SPSS Modeler Administration Console in IBM SPSS Deployment Manager to connect to the profile service.

Creating a Windows service for a serverprofile

On Windows, the administrator uses a script to create a service for a specified profile:

- Jane must specify the name of an existing profile, and then a service instance is created for that profile. The command line for the service will include the `profile` argument. The name of the service will follow a standard pattern including the profile name.
- Jane might need to use the service administration console later and edit the service properties if she needs to change the user name and password for the service (when running non-root).

On UNIX, there are also ways to create "services" that start automatically when the system boots. The administrator may want to create profile services using these mechanisms, but note they are not officially supported by IBM SPSS Modeler.

Managing Windows services for server profiles

Administrators can use a script to perform the following tasks:

- See which server profile services are running
- Start a particular service
- Start all services
- Stop a particular service
- Stop all services

When starting or stopping all services, the list of profiles is obtained by searching the sub-directories of the `profiles` directory.

Deleting a server profile's Windows service

On Windows, administrators can use a script to delete a service for a specified profile (if a service exists for the profile). The name of the profile must be specified.

Removing a server profile

After stopping the profile's service, administrators can remove a profile by deleting its folder from inside the `profiles` directory.

Updating SPSS Modeler Server

When applying a Fix Pack to SPSS Modeler Server, the Fix Pack is applied to all server profiles. On Windows, all profile services are stopped and restarted automatically. On UNIX, you must manually stop and restart them.

Uninstalling SPSS Modeler Server

When SPSS Modeler Server is uninstalled, all server profiles are uninstalled. Note that the `profiles` directory and any profiles it contains are not removed automatically. They must be deleted manually. On Windows, all profile services are uninstalled automatically. On UNIX, you must manually remove them.

Installing a new version of SPSS Modeler Server

When installing a new version of SPSS Modeler Server, any existing server profiles are not migrated automatically. An administrator must manually copy profiles from one installation to the next (and edit the configurations where necessary) to recreate the services.

Profile structure

The profiles directory

Server profiles are stored in a directory location chosen by the server administrator. The default location is a directory named `profiles` in the `[server install path]\config\` directory on the SPSS Modeler Server, but we recommend using a different directory for profile storage for the following reasons:

- Profiles can be shared between nodes in a cluster
- Profiles can be preserved across upgrades
- Administrators and other users who configure profiles do not need to be granted write authority to the SPSS Modeler Server installation directory

The `profiles` directory does not exist after a fresh SPSS Modeler Server installation. It is created when the first profile is created.

The `profiles` directory contains one sub-directory for each profile, and the sub-directory name matches the profile name. Because the directory name and the profile name are the same, the profile name cannot include characters that are not valid in file names. Nor should profile names contain spaces, because they are likely to cause problems in scripts. Also keep in mind that profile names must be unique within a single installation.

The only way to identify all the profiles for an installation is to identify the sub-directories of the `profiles` directory. There is no separate list of profiles maintained anywhere. There is also no limit on the number of profiles that can be created for an installation, aside from what can be tolerated by the host system.

In the `profiles` directory, the sub-directory for any given profile must contain at least one directory called `config`, and within that directory there must be at least one file called `options.cfg` that defines the profile configuration. This file contains a subset of the settings in the standard SPSS Modeler Server `options.cfg` file (located in `[server install path]/config`), as many as are needed for the profile. Settings not present in the profile configuration must be set from the common options file in the installation `config` directory. The profile configuration must contain at least a setting for `port_number` because every profile service must listen on a different port number.

The profile configuration may include other `*.cfg` files normally found in the installation `config` directory, in which case these are read instead of the standard files (only the options file is cumulative). Additional files most likely to be included in a profile configuration are `groups` and `passwords`. Files that are ignored in a profile configuration include the JVM and SSO configuration files which are shared across all profiles.

A profile directory may also contain `data` and `tmp` directories that override the common `data` and `tmp` file locations, unless alternative locations are specified in the profile configuration.

If you use profiles to achieve data isolation, be sure permissions are set appropriately on the relevant directories.

The profiles configuration file

The location of the `profiles` directory is specified in a new configuration file called `[server install path]\config\profiles.cfg`. This shares a common format with other configuration files in the same directory, and the key for setting the `profiles` directory is `profiles_directory`. For example:

```
profiles_directory, "C:\\SPSS\\Modeler\\profiles"
```

A separate file is used for profile configuration (rather than adding settings to the standard options file) for two reasons:

- The profile configuration determines how the options files are read, so there is an intrinsic difficulty in defining one in the other
- The profiles configuration file is designed to be managed automatically, using scripts, so in simple cases users need not be concerned with it at all (but it can still be safely hand-edited to support more complex scenarios)

Aside from the location of the `profiles` directory, the only other entry in `profiles.cfg` is a port number. For example:

```
profile_port, 28501
```

This is the default port number for the next profile to be created, and it is incremented automatically each time a profile is created using a script. The `profiles.cfg` file is created only as needed, so it does not exist in a fresh installation.

Starting a profile

The service executable (`modelerserver.exe`) accepts an additional argument, `profile`, that identifies the profile for the service:

```
modelerserver -server profile=<profile-name>
```

Multiple services can run from the same installation if each service uses a different profile. If the profile argument is omitted, the service uses the common installation defaults without any profile overrides.

When invoked with the `profile` argument, the service:

- Reads `[server install path]\config\profiles.cfg` to obtain the location of the profiles directory
- Reads `[profiles directory]\[profile name]\config\options.cfg` to obtain the profile configuration (in particular, the port number)

If either step fails for any reason, the service prints an error message to the log and stops. If the service is invoked with a profile and it cannot load the profile, then it will not run.

Environment variables

The service defines some additional environment variables so that path names, etc., can be expressed without knowledge of the current profile:

Variable	Value
PROFILE_NAME	The name of the current profile, or the empty string if no profile has been specified.
MODELERPROFILE	The full path to the directory for the current profile (for example, <code>\$MODELERSERVER\profiles\%PROFILE_NAME</code>). If no profile has been specified, the value is the same as the <code>\$MODELERSERVER</code> .
MODELERDATA	The full path to the data directory for the current profile (for example, <code>\$MODELERPROFILE\data</code>). If no profile has been specified, the value will point to the standard data directory <code>\$MODELERSERVER\data</code>

These environment variables are set by the service process, so they are only visible within that process and any child processes it creates. If you set these variables outside the service process, they will be ignored and redefined within the process as described.

Logging

Each profile service expects to have a separate, private folder in which to place its log files. There is one copy of `server_logging.log`, etc., for each profile.

The default `log4cxx.properties` configuration in the installation's `config` directory uses the `PROFILE_NAME` environment variable to identify the log directory for the service:

```
log4j.appender.LoggingAppender.File=${ALLUSERSPROFILE}/IBM/SPSS/Modeler
Server/17/log/${PROFILE_NAME}/server_logging.log
```

You can change the log location for all profiles by changing the line above and including one of the two profile-specific environment variables, either `PROFILE_NAME` or `MODELERPROFILE`. For example, to relocate the log directory within the profile directory:

```
log4j.appender.LoggingAppender.File=${MODELERPROFILE}/log
/server_logging.log
```

Alternatively, you can change the log location for a particular profile by creating and editing a copy of the `log4cxx.properties` file in the profile configuration.

Profile scripts

The scripts described in this section are provided to assist with the creation and management of SPSS Modeler Server profiles. All of the scripts are included in the `scripts/profiles` directory of the SPSS Modeler Server installation directory (for example, `C:\Program Files\IBM\SPSS\ModelerServer\18\scripts\profiles`).

Common script (for all platforms)

The following script helps create and manage profiles. Variants of this script are provided with different extensions for different platforms (`.bat` for Windows and `.sh` for UNIX). The operation is the same in each case.

Creating a profile

create_profile [options] <profile-name>

Creates a new profile with the specified name. The profile name must be appropriate for use as a directory name on the server host (because the script will create a directory with that name) and should not contain spaces. The name must be distinct from any existing profile name.

Options:

-d, --profiles-directory<profiles-directory>

Specifies the `profiles` directory in which this and all subsequent profiles should be created. You must specify this only for the first profile, but a good practice is to specify it every time. If you omit it the first time, a default location will be chosen. If you change the `profiles` directory on a subsequent call, the new profile will be created in the new location, but any existing profiles will be ignored unless they are moved separately to the new location.

-t, --template <profile-name>

Specifies the name of an existing profile to use as a template. The profile configuration is copied from the existing profile to the new profile, and only the port number is changed.

-p, --port-number <port-number>

Specifies the port number for the profile service. The port number must be unique to this profile. If you omit the port number, a default will be chosen.

-g, --group-name <group-name>

Specifies the name of an operating system group that will have exclusive access to this profile. The profile is configured to allow login access only to members of this group.

File system permissions are not changed, so you must perform that action separately.

Examples:

scripts\profiles\create_profile.bat -d C:\Modeler\Profiles comet

Creates a new profile called comet in the directory C:\Modeler\Profiles. The profile will listen on a default port number. To determine the port number, open the options.cfg file that is generated for the profile (in this example, C:\Modeler\Profiles\comet\config\options.cfg).

scripts\profiles\create_profile.bat --template comet --group-name "Meteor Users" --port-number 28510 meteor

Creates a new profile called meteor in the directory C:\Modeler\Profiles (remembered from the previous command). The profile will listen on port 28510 and login access will be allowed only to members of the group Meteor Users. All other configuration options will be copied from the existing profile comet.

Windows scripts

These scripts assist with the creation and management of Windows services for SPSS Modeler Server profiles. They use the Windows Service Control program (SC .EXE) to perform the requested operations, and the script output comes from SC .EXE unless otherwise noted. You must have administrator privileges on the local machine to perform most of these tasks.

See [Microsoft's TechNet documentation](#) about SC .EXE for more information.

Creating a Windows service for a profile

create_windows_service [options] <profile-name>

Creates the Windows service for a specified profile. You must have administrator privileges to create a service. Use the Services Management Console to set additional properties for the service after it is created (for example, to set the account details for the service log on).

Options:

-u, --service-user <account-name>

Specifies the account used for the service log on (passim). This can be a local user account, a domain user account, or the local computer name (standing for the local system account). The default is the local system account. If you specify an account other than the local system account, you must go to the Services Management Console and set the password for the account before the service will start.

-s, --register-spn

Registers a Service Principal Name (SPN) for the service so that clients can connect using Kerberos SSO. You must specify the service login account in this case (-u) so that the SPN can be registered to that account. You must have domain administrator privileges to use this option (or have been delegated the authority to register an SPN).

-H, --service-host <host-name>

Specifies the host name to use in construction of the SPN. This must be the host name clients will connect through, and it must be qualified with a domain name that maps to the Kerberos realm (in a simple active directory configuration, the domain name and the Kerberos realm are one and the same).

Examples:

scripts\profiles\create_windows_service.bat comet

Creates a Windows service for the comet profile. The service is owned by the local system account, and clients are expected to log in with a user name and password.

scripts\profiles\create_windows_service.bat -s -H modelerserver.mycompany.com -u MYCOMPANY\ProjectMeteor meteor

Creates a Windows service for the meteor profile. The service is owned by the ProjectMeteor domain account, and clients can log in using SSO. The service will not start until you go to the Services Management Console and set the password for the ProjectMeteor account. The account will automatically be granted the right to log in as a service.

Deleting a Windows service for a profile

delete_windows_service [options] <profile-names...>

Deletes the Windows services for the specified profiles. You must have administrator privileges to delete a service.

Options:

-s, --summary

Lists the names of the services that were deleted. Services that do not exist or cannot be deleted will not be listed. Without this option, the deletion status of all the specified services will be listed.

-a, --all

Deletes the services for all profiles.

Examples:

scripts\profiles\delete_windows_service.bat comet

Deletes the Windows services for the comet profile.

scripts\profiles\delete_windows_service.bat --all

Deletes the Windows services for all profiles.

Starting a Windows service for a profile

start_windows_service [options] <profile-names...>

Starts the Windows services for the specified profiles. You must have administrator privileges to start a service.

Options:

-s, --summary

Lists the names of the services that were started. Services that are already running or cannot be started will not be listed. Without this option, the status of all the listed services is listed.

-a, --all

Starts the services for all profiles.

Examples:

scripts\profiles\start_windows_service.bat -s comet meteor

Attempts to start the Windows services for the comet and meteor profiles, and lists the names of the services that were successfully started.

Stopping a Windows service for a profile

stop_windows_service [options] <profile-names...>

Stops the Windows services for the specified profiles. You must have administrator privileges to stop a service.

Options:

-s, --summary

Lists the names of the services that were stopped. Services that are already stopped or cannot be stopped will not be listed. Without this option, the status of all the listed services is listed.

-a, --all

Stops the services for all profiles.

Examples:

scripts\profiles\stop_windows_service.bat -a -s

Attempts to stop the Windows services for all profiles, and prints the names of those that were successfully stopped. The set of all profiles is obtained from the `profiles` directory.

Querying the state of a Windows service for a profile

query_windows_service [options] <profile-names...>

Shows the status of the Windows services for the specified profiles. You do not need administrator privileges to query a service.

Options:

-s, --summary

Lists just the names of the services and their current state (RUNNING, STOPPED, etc.). If a service cannot be queried for any reason (for example, if it does not exist), the status is reported as UNKNOWN. Without this option, the full status of all the listed services is listed.

-a, --all

Queries the service status for all profiles.

Examples:

scripts\profiles\query_windows_service.bat -a

Reports the full service status for all profiles.

UNIX script

The existing UNIX script that manages the SPSS Modeler Server service now accepts an additional profile argument so that SPSS Modeler Server profile services can be managed independently.

modelersrv.sh [options] {start|stop|kill|list}

Manages the main SPSS Modeler Server service. See [Chapter 4, “IBM SPSS Modeler Administration,” on page 21](#) for more information.

Options:

-p, --profile <profile-name>

Manages the service instance for the specified profile. When this argument is used, the specified command applies only to the instance for the specified profile. When this argument is absent, the `start` command starts just the default instance (a service with no profile), but the `stop`, `kill`, and `list` commands apply to all active instances.

Examples:

./modelersrv.sh --profile comet start

Starts the service for the comet profile.

./modelersrv.sh --profile meteor start

Starts the service for the meteor profile.

./modelersrv.sh list

Lists the processes for all active services.

./modelersrv.sh --profile comet stop

Stops the service for the comet profile.

./modelersrv.sh stop

Stops all active services

There is currently no supported method for starting SPSS Modeler Server profile services automatically on UNIX. The standard `auto.sh` script is available for configuring the system to start and stop the main SPSS Modeler Server service with the operating system, but this only applies to the default service -- not for any profile service.

Administration

IBM SPSS Modeler Server has a number of configurable options that control its behavior. You can set these options in two ways:

- Use the IBM SPSS Modeler Administration Console, which is available free of charge to current IBM SPSS Modeler customers. See the topic [“IBM SPSS Modeler Server Administration”](#) on page 31 for more information.
- Use the `options.cfg` text file, located in the `[server install path]/config` directory. See the topic [“Using the options.cfg file”](#) on page 40 for more information.

We recommend that you install IBM SPSS Deployment Manager and use its IBM SPSS Modeler Administration Console as your administration tool, rather than editing the `options.cfg` file. Editing the file requires access to the IBM SPSS Modeler Server file system, but IBM SPSS Modeler Administration Console allows you to authorize anyone with a user account to adjust these options. Also, IBM SPSS Modeler Administration Console provides additional information about the server processes, allowing you to monitor usage and performance. Unlike when editing the configuration file, most configuration options can be changed without restarting IBM SPSS Modeler Server.

More information about using IBM SPSS Modeler Administration Console and the `options.cfg` file is provided in the following sections.

IBM SPSS Modeler Server Administration

The Modeler Administration Console in IBM SPSS Deployment Manager provides a console user interface to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

Many of the options available through the Modeler Administration Console can also be specified in the `options.cfg` file, which is located in the SPSS Modeler Server installation directory under `/config`. However, the Modeler Administration Console provides a shared graphical interface that allows you to connect, configure, and monitor multiple servers.

Starting Modeler Administration Console

From the Windows Start menu, choose **[All] Programs**, then **IBM SPSS Collaboration and Deployment Services**, then **Deployment Manager**.

When you first run the application, you see empty Server Administration and Properties panes (unless you already have Deployment Manager installed with an IBM SPSS Collaboration and Deployment Services server connection already set up). After you configure Modeler Administration Console, the Server Administrator pane on the left displays a node for each SPSS Modeler Server that you want to administer. The right-hand pane shows the configuration options for the selected server. You must first [set up a connection](#) for each server that you want to administer.

Restarting the web service

Whenever you make changes to an IBM SPSS Modeler Server in the Administration Console, you must restart the web service.

To restart the web service on Microsoft Windows:

1. On the computer where you installed IBM SPSS Modeler, select **Services** from Administrative Tools on the Control Panel.
2. Locate the server in the list and restart it.
3. Click **OK** to close the dialog box.

To restart the web service on UNIX:

On UNIX, you must restart the IBM SPSS Modeler Server by running the **modelersrv.sh** script in the IBM SPSS Modeler Server installation directory.

1. Change to the IBM SPSS Modeler Server installation directory. For example, at a UNIX command prompt, type:
`cd /usr/<modelersrv>`, where `modelersrv` is the IBM SPSS Modeler Server installation directory.
2. To stop the server, at the command prompt, type
`./modelersrv.sh stop`
3. To restart the server, at the command prompt, type
`./modelersrv.sh start`

Configuring Access with Modeler Administration Console

Administrator access to SPSS Modeler Server through the Modeler Administration Console included with IBM SPSS Deployment Manager is controlled with the `administrators` line in the `options.cfg` file, located in the SPSS Modeler Server installation directory under `/config`. This line is commented out by default, so you must edit this line to allow access to specific people, or use `*` to allow access to all users, as shown in the following examples:

```
administrators, "*"
administrators, "jsmith,mjones,achavez"
```

- The line must begin with `administrators`, and the entries must be contained in quotation marks. Entries are case sensitive.
- Separate multiple user IDs with commas.
- For Windows accounts, do not use domain names.
- Use the asterisk with care. It allows anyone with a valid user account for IBM SPSS Modeler Server (which, in most cases, is anyone on the network) to log in and change the configuration options.

Configuring Access with User Access Control

To use the Modeler Administration Console to make updates to a SPSS Modeler Server configuration installed on a Windows machine that has User Access Control (UAC) enabled, you must have read, write, and execute permissions defined on the `config` directory and on the `options.cfg` file. These (NTFS) permissions must be defined at the specific user level and not at group level, this is due to the way that UAC and NTFS permissions interact.

The Modeler Administration Console is included in IBM SPSS Deployment Manager.

SPSS Modeler Server connections

You must specify a connection to each SPSS Modeler Server on your network that you want to administer. You must then log in to each server. Although the server connection is remembered across Modeler Administration Console sessions in IBM SPSS Deployment Manager, the login credentials are not. You must log in every time you start IBM SPSS Deployment Manager.

To set up a server connection

1. Ensure that the IBM SPSS Modeler Server service is started.
2. From the File menu, choose **New** and then **Administered Server Connection**.
3. On the first page of the wizard, enter a name for your server connection. The name is for your own use and should be something descriptive; for example, *Production Server*. Ensure that Type is set to **Administered IBM SPSS Modeler Server**, then click **Next**.

4. On the second page, enter the hostname or IP address of the server. If you have changed the port from the default, enter the port number. Click **Finish**. The new server connection is shown in the Server Administrator pane.

To perform administration tasks, you must now log in.

To log in to the server

1. In the Server Administrator pane, double-click to select the server to which you want to log in.
2. In the Login dialog box, enter your credentials. (Use your user account for the server host.) Click **OK**.

If the login fails with the message **Unable to obtain administrator rights on server**, the most likely cause is that administrator access has not been configured correctly. See the topic [“Configuring Access with Modeler Administration Console”](#) on page 32 for more information.

If the login fails with the message **Failed to connect to server '<server>'**, make sure that the user ID and password are correct, then make sure that the IBM SPSS Modeler Server service is running. For example, under Windows, go to Control Panel > Administrative Tools > Services and check the entry for IBM SPSS Modeler Server. If the Status column does not show **Started**, select this line on the screen and click **Start**, then retry the login.

Once you log in to your IBM SPSS Modeler Server, two options are shown below the server name, [Configuration](#) and [Monitoring](#). Double-click one of these options.

SPSS Modeler Server Configuration

The Configuration pane shows configuration options for SPSS Modeler Server. Use this pane to change the options as desired. Click **Save** on the toolbar to save the changes. Note that changing any option marked with an asterisk (*) requires a server restart to take effect.

The options are described in the following sections, with the corresponding line in `options.cfg` given in parentheses for each option. Options that are visible only in `options.cfg` are described at the end of this section.

Note: If a non-root user wants to change these options, write permission is required for the SPSS Modeler Server `config` directory.

Connections/Sessions

Modeler port number. (`port_number`) The port number for SPSS Modeler Server to listen on. Change if another application already uses the default. End users must know the port number in order to use SPSS Modeler Server.

Embedded DataView service port number. (`data_view_port_number`) The port number for the DataView service embedded in SPSS Modeler Server to listen on. Change if another application already uses the default.

Maximum number of connections. (`max_sessions`) Maximum number of server sessions at one time. A value of -1 indicates no limit.

Analytic Server connection

Enable Analytic Server SSL (`as_ssl_enabled`). Specify Y to encrypt communications between Analytic Server, and SPSS Modeler otherwise, N.

Host (`as_host`). The IP address of the Analytic Server.

Port Number (`as_port`). The Analytic Server port number.

Context Root (`as_context_root`). The context root of the Analytic Server.

Tenant (`as_tenant`). The tenant that the SPSS Modeler Server installation is a member of.

Realm (`as_realm`). The realm used for this Analytic Server.

Prompt for Password (`as_prompt_for_password`). Specify N if the SPSS Modeler Server is configured with the same authentication system for users and passwords as the system that is used on Analytic Server; for example, when you use Kerberos authentication, otherwise, Y.

Note: If you intend to use Kerberos SSO, you must set extra options in the `options.cfg` file. For more information, see the topic "Options visible in options.cfg" later in this chapter.

Note: To connect SSL enabled Analytic Server, extra steps are required as follows:

1. Use the following command to extract the certificate file `trust.cer` from the JKS file (i.e. `trust.jks`):

```
/bin/keytool -export -alias server-alias -storepass pass4jks -file /home/sslkeys/trust.cer -keystore /home/sslkeys/trust.jks
```

2. Import the `trust.cer` file into `cacerts` in the JRE used by your application server.
3. Import the `trust.cer` file into `caserts` in the JRE used by the SPSS Modeler Server.
4. Restart the SPSS Modeler Server and the IBM SPSS Collaboration and Deployment Services Repository Server.

Data file access

Python executable path for Extension nodes and Custom Dialog Builder.

(`eas_pyspark_python_path`) Full path to the Python executable file, including the name of the file. [`program_files_restricted`] might need to be set to No depending on your Python installation location.

Restrict access to only data file path. (`data_files_restricted`) When set to **yes**, this option restricts data files to the standard data directory and any files that are listed in the **Data File Path**. If you want to use the **View Data** feature while this restriction is enabled, you must set matching temporary file paths in both the `temp_directory` and `data_file_path` parameters.

Data file path. (`data_file_path`) A list of directories to which clients are allowed to read and write data files. This option is ignored unless the **Restrict Access to Only Data File Path** option is turned on. Use forward slashes in all path names. On Windows, specify multiple directories by using semicolons (for example, [`server install path`]/data;c:/data;c:/temp). On Linux and UNIX, use colons (:) instead of semicolons. The data file path must include any path that is specified in the `temp_directory` parameter.

Restrict access to only program files path. (`program_files_restricted`) When set to **yes**, this option restricts program file access to the standard `bin` directory and any files that are listed in the **Program files path**. As of release 17, the only program file to which access is restricted is the Python executable (see the **Python executable path**).

Program files path. (`program_file_path`) A list of directories from which clients are allowed to execute programs. This option is ignored unless the **Restrict Access to Only Program Files Path** option is turned on. Use forward slashes in all path names. Specify multiple directories by using semicolons.

Maximum file size. (`max_file_size`) Maximum size (in bytes) of temporary and exported data files that are created during stream execution (does not apply to SAS and SPSS Statistics data files). A value of `-1` indicates no limit.

Temporary directory. (`temp_directory`) The directory used to store temporary data files (cache files). Ideally, this directory is on a separate high-speed drive or controller because speed of access to this directory can have a significant impact on performance. You can specify multiple temporary directories, separating each with a comma (for example: `temp_directory, "D:/Modeler_temp, C:/Program Files/IBM/SPSS/ModelerServer/<version>/Tmp"`). These directories should be on different disks. The first directory is used most often, and the other directories are used to store temporary work files when certain data preparation operations (such as sort) use parallelism during execution. Allowing each execution thread to use separate disks for temporary storage can improve performance. Use forward slashes in all path specifications.

Note:

- Temporary files are generated in this directory during startup of SPSS Modeler Server. Ensure that you have the necessary access rights to this directory (for example, if the temporary directory is a shared network folder), otherwise SPSS Modeler Server startup fails.
- The `temp_directory` setting does not apply when you run Evaluation streams through IBM SPSS Collaboration and Deployment Services jobs. When you run such a job, a temporary file is created. By default, the file is saved to the IBM SPSS Modeler Server installation directory. You can change the default data folder that the temp files are saved to when you create the IBM SPSS Modeler Server connection in IBM SPSS Modeler.

Python executable path for bulk loading. (`python_exe_path`) Full path to the Python executable including the executable name. If access to program files is restricted, then you must add the directory that contains the Python executable to the `program_file_path` parameter (see **Restrict access to only program files path**).

Path to OPL library of full version of CPLEX. (`cplex_opl_lib_path`) Path to the OPL library for the full version of CPLEX.

Performance/Optimization

Stream rewriting. (`stream_rewriting_enabled`) Allows the server to optimize streams by rewriting them. For example, the server might push data reduction operations closer to the source node to minimize the size of the dataset as early as possible. Disabling this option is normally recommended only if the optimization causes an error or other unexpected results. This setting overrides the corresponding client optimization setting. If this setting is disabled in the server, then the client cannot enable it. But if it is enabled in the server, the client can choose to disable it.

Parallelism. (`max_parallelism`) Describes the number of parallel worker threads that SPSS Modeler is allowed to use when running a stream. Setting this to 0 or any negative number causes IBM SPSS Modeler to match the number of threads to the number of available processors on the computer; the default value for this option is `-1`. To turn off parallel processing (for machines with multiple processors), set this option to 1. To allow limited parallel processing, set it to a number smaller than the number of processors on your machine. Note that a hyperthreaded or dual-core processor is treated as two processors.

Buffer size (bytes). (`io_buffer_size`) Data files transferred from the server to the client are passed through a buffer of this number of bytes.

Cache compression. (`cache_compression`) An integer value in the range 0 to 9 that controls the compression of cache and other files in the server's temporary directory. Compression reduces the amount of disk space used, which can be important when space is limited. Compression increases processor time, but this is almost always made up by the reduction in disk access time. Note that only certain caches, those accessed sequentially, can be compressed. This option does not apply to random-access caches, such as those used by the network training algorithms. A value of 0 disables compression entirely. Values from 1 upward provide increasing degrees of compression but with a corresponding cost in access time. The default value is 1; higher values may be needed where disk space is at a premium.

Memory usage multiplier. (`memory_usage`) Controls the proportion of physical memory allocated for sorting and other in-memory caches. The default is 100, which corresponds to approximately 10% of physical memory. Increase this value to improve sort performance where free memory is available, but be careful of increasing it so high as to cause excessive paging.

Modeling memory limit percentage. (`modelling_memory_limit_percentage`) Controls the proportion of physical memory allocated for training Kohonen and *k*-means models. The default is 25%. Increase this value to improve training performance where free memory is available, but be careful of increasing it so high as to cause excessive paging when data spills onto the disk.

Allow modeling memory override. (`allow_modelling_memory_override`) Enables or disables the **Optimize for Speed** option in certain modeling nodes. The default is enabled. This option allows the modeling algorithm to claim all available memory, bypassing the percentage limit option. You may want to disable this if you need to share memory resources on the server machine.

Maximum and minimum server port. (`max_server_port` and `min_server_port`) Specifies the range of port numbers that can be used for the additional socket connections between client and server that are

required for interactive models and stream execution. These require the server to listen on another port; not restricting the range could cause problems for users on systems with firewalls. Default value for both is -1, meaning "no restriction." Thus, for example, to set the server to listen on port 8000 or above, you would set `min_server_port` to 8000 and `max_server_port` to -1.

Note that you must open additional ports over the main server port to open or execute a stream, and correspondingly more ports if you want to open or execute concurrent streams. This is required in order to capture feedback from the stream execution.

By default, IBM SPSS Modeler will use any open port that is available; if it does not find one (for example, if they are all closed by a firewall), an error is displayed when you execute the stream. To configure the range of ports, IBM SPSS Modeler will need two open ports (in addition to the main server port) available per concurrent stream, plus 3 additional ports for each ODBC connection from within any connected client (2 ports for the ODBC connection for the duration of that ODBC connection, and an additional temporary port for authentication).

Note: An ODBC connection is an entry in the database connections list, and can be shared between multiple database nodes specified with the same database connection.

Note: It is possible that the authentication ports can be shared if the connections are made at different times.

Note: Best practice dictates that the same ports should be used to communicate with both IBM SPSS Collaboration and Deployment Services and SPSS Modeler Client. These can be set as `max_server_port` and `min_server_port`.

Note: If you change these parameters, you need to restart SPSS Modeler Server for the change to take effect.

Array fetch optimization. (`sql_row_array_size`) Controls the way that SPSS Modeler Server fetches data from the ODBC datasource. The default value is 1, which fetches a single row at a time. Increasing this value causes the server to read the information in larger chunks, fetching the specified number of rows into an array. With some operating system/database combinations, this can result in improvements to the performance of SELECT statements.

SQL

Maximum SQL string length. (`max_sql_string_length`) For a string imported from the database with SQL, the maximum number of characters that are guaranteed to be passed successfully. Depending on the operating system, string values longer than this may be truncated on the right without warning. The valid range is between 1 and 65,535 characters. This property also applies to the Database export node.

Note: The default value for this parameter is 2048. If the text you are analyzing is longer than 2048 characters (for example, this may occur if using the SPSS Modeler Text Analytics Web Feed node) we recommend increasing this value if working in native mode otherwise your results may be truncated. If you are using a database and user-defined functions (UDF), this restriction does not occur; this can account for differences in results between native and UDF modes.

Automatic SQL generation. (`sql_generation_enabled`) Allows automatic SQL generation for streams, which may substantially improve performance. The default is enabled. Disabling this option is recommended only if the database is not able to support queries submitted by SPSS Modeler Server. Note that this setting overrides the corresponding client optimization setting; also note that for purposes of scoring, SQL generation must be enabled separately for each modeling node regardless of this setting. If this setting is disabled in the server, then the client cannot enable it. But if it is enabled in the server, the client can choose to disable it.

Default SQL string length. (`default_sql_string_length`). Specifies the default width of string columns that will be created within database cache tables. String fields in database cache tables will be created with a default width of 255 if there is no upstream type information. If you have wider values than this in your data, either instantiate an upstream Type node with those values, or set this parameter to a value that is large enough to accommodate those string values.

Enable Database UDF. (`db_udf_enabled`). If set to Y (default), causes the SQL generation option to generate user-defined function (UDF) SQL instead of pure SPSS Modeler SQL. UDF SQL usually outperforms pure SQL.

SSL

Enable SSL. (`ssl_enabled`) Enables SSL encryption for connections between SPSS Modeler and SPSS Modeler Server.

SSL keystore file. (`ssl_keystore`) The SSL key database file to be loaded when the server starts (either a full path or a relative path to the SPSS Modeler installation directory).

SSL keystore stash file. (`ssl_keystore_stash_file`) The name of the key database password stash file to be loaded when the server starts up (either a full path or a relative path to the SPSS Modeler installation directory). On Linux, if you want to leave this setting blank and be prompted for the password when starting the SPSS Modeler Server, see the following instructions:

- On Linux/UNIX:

1. Make sure the `ssl_keystore_stash_file` setting in `options.cfg` does not have a value.

2. Locate the following line in the `modelersrv.sh` file:

```
if "$INSTALLEDPATH/$SCLEMDNAME" -server $ARGS; then
```

3. Add the `-request_ssl_password` switch as follows:

```
if "$INSTALLEDPATH/$SCLEMDNAME" -request_ssl_password -server $ARGS; then
```

4. Restart the SPSS Modeler Server. You will be prompted for a password. Enter the correct password, click **OK**, and the server will start.

Keystore certificate label. (`ssl_keystore_label`) Label for the specified certificate.

Note: To use the Administration Console with a server setup for SSL, you must import any certificates required by SPSS Modeler Server into the Deployment Manager trust store (under `./jre/lib/security`).

Note: If you change these parameters, you need to restart SPSS Modeler Server for the change to take effect.

Coordinator of Processes configuration

Host. (`cop_host`) The hostname or IP address of the Coordinator of Processes service. The default "spsscops" is a vanity name which administrators can choose to add as an alias for the IBM SPSS Collaboration and Deployment Services host in DNS.

Port number. (`cop_port_number`) The port number of the Coordinator of Processes service. The default, 8080, is the IBM SPSS Collaboration and Deployment Services default.

Context root. (`cop_context_root`) The URL of the Coordinator of Processes service.

Login name. (`cop_username`) The user name for authentication to the Coordinator of Processes service. This is an IBM SPSS Collaboration and Deployment Services login name so may include a security-provider prefix (for example: `ad/jsmith`).

Password. (`cop_password`) The password for authentication to the Coordinator of Processes service.

Note: If you update the `options.cfg` file manually instead of using the Modeler Administration Console in IBM SPSS Deployment Manager, you must manually encode the `cop_password` value that you specify in the file. Plain text passwords are invalid and cause registration with the Coordinator of Processes to fail.

Follow these steps to manually encode the password:

1. Open a Command Prompt, navigate to the SPSS Modeler `./bin` directory, and run the command `pwutil.bat/sh`.

2. When requested, type the user name (the `cop_username` you are specifying in `options.cfg`) and press Enter.

3. When requested, type the password for that user.

The encoded password is displayed between double quotes on the command line as part of the returned string. For example:

```
C:\Program Files\IBM\SPSS\Modeler\18\bin>pwutil
User name: copuser
Password: Pass1234
copuser, "0Tqb4n.ob0w1s"
```

4. Copy the encoded password, without the double quotes, and paste it between the double quotes that already exist for the `cop_password` value in the `options.cfg` file.

Enabled. (`cop_enabled`) Determines whether the server should attempt to register with the Coordinator of Processes. The default is *not* to register because the administrator should choose which services are advertised through the Coordinator of Processes.

SSL Enabled. (`cop_ssl_enabled`) Determines whether SSL is used to connect to the Coordinator or Processes server. If this option is used, you must import the SSL certificate file into the SPSS Modeler Server JRE. To do this, you must obtain the SSL certificate file and its alias name and password. Then run the following command on the SPSS Modeler Server:

```
$JAVA_HOME/bin/keytool -import -trustcacerts -alias $ALIAS_NAME -file
$CERTIFICATE_FILE_PATH -keystore $ModelerServer_Install_Path/jre/lib/security/
cacerts
```

Server name. (`cop_service_name`) The name of this SPSS Modeler Server instance; the default is the host name.

Description. (`cop_service_description`) A description of this instance.

Update interval (min). (`cop_update_interval`) The number of minutes between keep-alive messages; the default is 2.

Weight. (`cop_service_weight`) The weight of this instance, specified as an integer between 1 and 10. A higher weight attracts more connections. The default is 1.

Service host. (`cop_service_host`) The fully-qualified host name of the IBM SPSS Modeler Server host. The default of the host name is derived automatically; the administrator can override the default for multi-homed hosts.

Default data path. (`cop_service_default_data_path`) The default data path for a Coordinator of Processes registered IBM SPSS Modeler Server installation.

Options visible in options.cfg

Most configuration options can be changed using the IBM SPSS Modeler Administration Console included with IBM SPSS Deployment Manager. But there are some exceptions, such as those described in this section. The options in this section must be changed by editing the `options.cfg` file. See [“IBM SPSS Modeler Server Administration” on page 31](#) and [“Using the options.cfg file” on page 40](#) for more information. Note that there may be additional settings in `options.cfg` that are not listed here.

Note: This information only applies to a remote server (IBM SPSS Modeler Server, for example).

administrators. Specify the user names of those users to whom you want to grant administrator access. See the topic [“Configuring Access with Modeler Administration Console” on page 32](#) for more information.

`allow_config_custom_overrides`. Do not modify unless instructed to do so by a technical-support representative.

`data_view_port_number`. You can right-click a data node and select **View Data** to examine and refine your data in interesting ways with advanced data visualizations. This feature uses the port number 28900 by default. Modify the value for this `data_view_port_number` configuration option if you need to use a different port number. We recommend using the default, if possible.

fips_encryption. Enables FIPS compliant encryption. The default is N.

group_configuration. When enabled, IBM SPSS Modeler Server checks the groups.cfg file which controls who can log on to the server.

max_transfer_size. For internal system use only. **Do not modify.**

shell. (UNIX servers only) Overrides the default setting for the UNIX shell, for example shell, "/usr/bin/ksh". By default, IBM SPSS Modeler uses the shell defined in the user profile of the user who is connecting to IBM SPSS Modeler Server.

start_process_as_login_user. Set this to Y if you are running SPSS Modeler Server with a private password database, starting the server service from a non-root account.

use_bigint_for_count. When the number of the records to be counted is larger than what a normal integer ($2^{31}-1$) can hold and is within the range of -2^{63} to $2^{63}-1$, set this option to Y. When this option is set to Y, and a stream is connected to Db2; SQL Server; or a Teradata, Oracle, or Netezza database, a function is used where a record count is needed (for example, the **Record_Count** field generated by the Aggregate node).

When this option is enabled, and if working with either Db2 or SQL Server, SPSS Modeler uses COUNT_BIG() for record counting. If working with Teradata, Oracle, or Netezza, SPSS Modeler will use COUNT(). For all other databases, there is no SQL pushback for the function. The difference is that when use_bigint_for_count is enabled, all record counts are saved as BIG INT (or LONG) type, as compared to normal integer when the options is disabled.

From the point of view of database numeric, see the following table for the results of setting use_bigint_for_count to Y and N for the same data.

Database Numeric	Interpretation for data when use_bigint_for_count is set to Y	Interpretation for data when use_bigint_for_count is set to N
Numeric(18,0)	int	int
Numeric(19,0)	int	string
Numeric(38,0)	int	string

cop_ssl_enabled. Set this option to Y if you are using SSL to connect to the Coordinator of Processes Service. If this option is used, you must import the SSL certificate file into the SPSS Modeler Server JRE. To do this, you must obtain the SSL certificate file and its alias name and password. Then run the following command on the SPSS Modeler Server:

```
$JAVA_HOME/bin/keytool -import -trustcacerts -alias $ALIAS_NAME -file  
$CERTIFICATE_FILE_PATH -keystore $ModelerServer_Install_Path/jre/lib/security/  
cacerts
```

cop_service_default_data_path. You can use this option to set the default data path for a Coordinator of Processes registered IBM SPSS Modeler Server installation.

Users can create their own Analytic Server connections in SPSS Modeler via **Tools > Analytic Server Connections**. Administrators can also define a default Analytic Server connection using the following properties:

as_ssl_enabled. Y or N.

as_host. Specify the Analytic Server host name or IP address.

as_port. Specify the Analytic Server port number.

as_context_root. Specify the Analytic Server context root.

as_tenant. Specify the name of the tenant that the IBM SPSS Modeler Server is a member of

as_prompt_for_password. Y or N.

By default, Analytic Server authentication using the Kerberos method is not enabled. To enable Kerberos authentication, use the three following properties:

as_kerberos_auth_mode. To enable Kerberos authentication, set this option to Y.

as_kerberos_krb5_conf. Specify the path to the Kerberos configuration file that Analytic Server should use; for example, c:\windows\krb5.conf.

as_kerberos_krb5_spn. Specify the Analytic Server Kerberos SPN; for example, HTTP/ashost.mydomain.com@MYDOMAIN.COM.

SPSS Modeler Server Monitoring

The monitoring pane of Modeler Administration Console in IBM SPSS Deployment Manager shows a snapshot of all processes running on the SPSS Modeler Server computer, similar to the Windows Task Manager. To activate the monitoring pane, double-click the Monitoring node beneath the desired server in the Server Administrator pane. This populates the pane with a current snapshot of data from the server. The data refreshes at the rate shown (one minute by default). To refresh the data manually, click the **Refresh** button. To show only SPSS Modeler Server processes in this list, click the **Filter out non-SPSS Modeler processes** button.

Using the options.cfg file

The options.cfg file is located in the [server_install_path]/config directory. Each setting is represented by a comma-separated name-value pair, where the name is the name of the option and the value is the value for the option. Pound (hash) signs (#) indicate comments.

Note: Most configuration options can be changed using IBM SPSS Modeler Administration Console in IBM SPSS Deployment Manager, rather than this configuration file, but there are a few exceptions. See the topic [“Options visible in options.cfg” on page 38](#) for more information.

By using IBM SPSS Modeler Administration Console, you can avoid server restarts for all options except the server port. See the topic [“IBM SPSS Modeler Server Administration” on page 31](#) for more information.

Note: This information only applies to a remote server (IBM SPSS Modeler Server, for example).

Configuration options that can be added to the default file

By default, in-database caching is enabled with IBM SPSS Modeler Server. You can disable this feature by adding the following line to the options.cfg file:

```
enable_database_caching, N
```

Doing so causes temporary files to be created on the server and not in the database.

To view or change the IBM SPSS Modeler Server configuration options:

1. Open the options.cfg file with a text editor.
2. Locate the options of interest. For a full list of options, see [“SPSS Modeler Server Configuration” on page 33](#).
3. Edit the values, as appropriate. Note that all pathname values must use a forward slash (/) rather than a backslash as the pathname separator.
4. Save the file.
5. Stop and restart IBM SPSS Modeler Server so that the changes will take effect.

Closing unused database connections

By default, IBM SPSS Modeler caches at least one connection to a database once that connection has been accessed. The database session is held open even when streams requiring database access are not being executed.

Caching database connections can improve execution times by removing the need for IBM SPSS Modeler to reconnect to the database each time a stream is executed. However, in some environments, it is important for applications to release database resources as quickly as possible. If too many IBM SPSS Modeler sessions maintain connections to the database that are no longer used, database resources may become exhausted.

You can avoid this possibility by turning off the IBM SPSS Modeler option `cache_connection` in a custom database configuration file. Doing so can also make IBM SPSS Modeler more resilient to faults in the database connection (such as timeouts) that can occur when connections are used over a long period of time by an IBM SPSS Modeler session.

To cause unused database connections to be closed:

1. Locate the `[server install path]/config` directory.
2. Add the following file (or open it, if it already exists):

```
odbc-custom-properties.cfg
```

3. Add the following line to the file:

```
cache_connection, N
```

4. Save and close the file.
5. Restart IBM SPSS Modeler Server.

Note:

In-database caches are saved in database as either a regular table or a temporary table, depending on each database's implementation. For example, temporary tables are used for Db2, Oracle, Amazon Redshift, Sybase, and Teradata. For these databases, setting `cache_connection` to N does not work as expected because the temporary table is only valid within a session (it will be cleaned automatically by the database when the database connection is closed).

So when running an SPSS Modeler stream against one of these databases with `cache_connection` set to N, an error such as **Failed to create table for in-database caching. Using file cache instead.** may result. This indicates that SPSS Modeler failed to create the in-database cache. Also, in some cases for an SPSS Modeler-generated SQL query, a temporary table is used but the table is empty.

To work around this issue, you can choose to use a regular database table for in-database caches. To do this, create a custom database property configuration file that contains the following line:

```
table_create_temp_sql, 'CREATE TABLE <table-name> <(table-columns)>'
```

This forces a regular database table to be used for the in-database cache, and the table will be dropped when all connections to the database are closed or when the working stream is closed.

Using SSL to secure data transfer

How SSL works

SSL relies on the server's public and private keys, in addition to a public key certificate that binds the server's identity to its public key.

1. When a client connects to a server, the client authenticates the server with the public key certificate.
2. The client then generates a random number, encrypts the number with the server's public key, and sends the encrypted message back to the server.

3. The server decrypts the random number with its private key.
4. From the random number, both the server and client create the session keys used for encrypting and decrypting subsequent information.

The public key certificate is typically signed by a certificate authority. Certificate authorities, such as VeriSign and Thawte, are organizations that issue, authenticate, and manage security credentials contained in the public key certificates. Essentially, the certificate authority confirms the identity of the server. The certificate authority usually charges a monetary fee for a certificate, but self-signed certificates can also be generated.

Securing client/server and server-server communications with SSL

The main steps in securing client/server and server-server communications with SSL are:

1. Obtain and install the SSL certificate and keys.
2. Enable and configure SSL in the server administration application (IBM SPSS Deployment Manager).
3. If using encryption certificates with a strength greater than 2048 bits, install unlimited strength encryption on the client computers.
4. Instruct users to enable SSL when connecting to the server.

Notes:

- Occasionally a server product acts as a client. An example is IBM SPSS Statistics Server connecting to the IBM SPSS Collaboration and Deployment Services Repository. In this case, IBM SPSS Statistics Server is the *client*.
- On Linux/UNIX systems where both a non-root and SSL configuration are enabled, SSL security will be reduced. Because all user sessions run under the same credential as each other and as the Modeler Server daemon, the SSL certificate data that should be kept secret will instead be exposed to all users. This allows users to easily bypass the normal protections SSL provides to all other users. See [“Introduction” on page 81](#).

Obtaining and installing SSL certificate and keys

The first steps you must follow to configure SSL support are:

1. Obtain an SSL certificate and key file. There are three ways you can do this:
 - Purchase them from a public certificate authority (such as VeriSign, Thawte, or Entrust). The public certificate authority (CA) signs the certificate to verify the server that uses it.
 - Generate the key and certificate files with a third-party certificate authority. If this approach is taken then the third-party CA's root certificate must be imported into the client and server keystore files. See the topic [“Importing a third-party root CA certificate” on page 44](#) for more information.
 - Generate the key and certificate files with an internal self-signed certificate authority. The steps to do this are:
 - a. Prepare a key database. See the topic [“Creating an SSL key database” on page 43](#) for more information.
 - b. Create the self-signed certificate. See the topic [“Creating a self-signed SSL certificate” on page 44](#) for more information.
2. Copy the `.kdb` and `.sth` files created in step 1 into a directory to which the IBM SPSS Modeler Server has access and specify the path to that directory in the `options.cfg` file. .
Note: Use forward slashes as separators in the directory path.
3. Set the following parameters in the `options.cfg` file:
 - `ssl_enabled, Y`
 - `ssl_keystore, "<filename>.kdb"` where `<filename>` is the name of your key database.

- `ssl_keystore_stash_file`, "`<filename>.sth`" where `<filename>` is the name of the key database password stash file.
 - `ssl_keystore_label`, `<label>` where `<label>` is the label of your certificate.
4. For self-signed or third-party certificates install the certificate on client systems. For purchased public CA certificates, this step is not required. Ensure that access permissions deny casual browsing of the directory that contains the certificate. See the topic [“Installing a self-signed SSL certificate” on page 44](#) for more information.

Configuring the environment to run GSKit

The GSKCapiCmd is a non-Java-based command-line tool, and Java™ does not need to be installed on your system to use this tool; it is located in the `<Modeler installation directory>/bin` folder. The process to configure your environment to run IBM Global Security Kit (GSKit) varies depending on the platform in use.

To configure for Linux/Unix, add the shared libraries directory `<Modeler installation directory>/lib` to your environment:

```
$export <Shared library path environment variable>=<modeler_server_install_path>/bin
$export PATH=$PATH:<modeler_server_install_path>/bin
```

The shared library path variable name depends on your platform:

- HP-UX uses the variable name: `SHLIB_PATH`
- Linux uses the variable name: `LD_LIBRARY_PATH`

For example, to set the environment on Linux, use:

```
$export LD_LIBRARY_PATH=/path/to/gskit/bin
$export PATH=$PATH:/path/to/gskit/bin
```

Account access to files

Ensure that you grant the correct permissions for the accounts that will access the SSL files:

1. For all accounts that are used by SPSS Modeler for connection, grant read access to the SSL files.

Note: This also applies to the *Log on as* user that is defined in the SPSS Modeler Server service. On UNIX or Linux, it applies to the user you are starting the server as.
2. For Windows, it is not enough that the accounts are in the Administrators group and that permission is given to that Administrators group when User Access Control (UAC) is enabled. In addition you must take one of the following actions:
 - Give the accounts permission separately.
 - Create a new group, add the accounts into the new group, and give the group permission to access the SSL files.
 - Disable UAC.

Creating an SSL key database

Use the GSKCapiCmd tool to create your key database. Before using the tool, you must configure your environment; see the topic [“Configuring the environment to run GSKit” on page 43](#) for more information

To create the key database, run GSKit and enter the following command:

```
gsk<ver>capiCmd[_64] -keydb -create -populate -db <filename>.kdb -pw <password> -stash
```

where `<ver>` is the GSKit version number, `<filename>` is the name you want to use for the key database file, and `<password>` is the password for the key database.

The `-stash` option creates a stash file at the same path as the key database, with a file extension of `.sth`. GSKit uses the stash file to obtain the password to the key database so that it doesn't have to be entered on the command line each time.

Note: You should use strong file system protection on the `.sth` file.

Creating a self-signed SSL certificate

To generate a self-signed certificate and store it in the key database, use the following command:

```
gsk<ver>capicmd[_64] -cert -create -db <filename>.kdb -stash -dn  
"CN=myserver,OU=mynetwork,O=mycompany,  
C=mycountry" -label <label> -expire <Number of days certificate is valid> -default_cert yes
```

where `<ver>` is the GSKit version number, `<filename>` is the name of the key database file, `<Number of days certificate is valid>` is the physical number of days that the certificate is valid, and `<label>` is a descriptive label to help you identify the file (for example, you could use a label such as: `myselfsigned`).

Installing a self-signed SSL certificate

For the client machines that connect to your server using SSL, you must distribute the public part of the certificate to the clients so that it can be stored in their key databases. To do this, perform the following steps:

1. Extract the public part to a file using the following command:

```
gsk<ver>capicmd[_64] -cert -extract -db <filename>.kdb -stash -label <label>  
-format ascii -target mycert.arm
```

2. Distribute `mycert.arm` to the clients. It should be copied to their `jre/bin` directory.
3. Add the new certificate to the clients' key database using the following command:

```
keytool -import -alias <label> -keystore ..\lib\security\cacerts -file mycert.arm
```

If prompted for a password, use: `changeit`. The `keytool` is located in the `<Modeler installation directory>\jre\bin` directory (or in the `<Modeler installation directory>/SPSSModeler.app/Contents/PlugIns/jre/Contents/Home/bin` directory on Mac).

Importing a third-party root CA certificate

Instead of purchasing a certificate from a well known certificate authority (CA) or creating a self-signed certificate, you can use a third-party certificate authority to sign your server certificates. The client and server must have access to the third-party CA's root certificate to verify the server certificates that are signed by the third-party CA. To do this:

1. Obtain the third-party CA root certificate. The process for this varies depending on the third-party CA's procedures. Third-party CAs often make their root certificates available for download.
2. Add the certificate to the servers' key database using the following command:

```
gsk<ver>capicmd[_64] -cert -add -db <filename>.kdb -stash -label <label> -file  
<ca_certificate>.cert  
-format binary -trust enable
```

3. Add the certificate to the clients' key database using the following command:

On Windows:

```
C:> cd <Modeler Client installation path>\jre\bin  
C:> keytool -import -keystore ..\lib\security\cacerts -file <ca_certificate>.cert -alias  
<label>
```

On Mac:

```
C:> cd <Modeler Client installation path>/SPSSModeler.app/Contents/PlugIns/jre/Contents/
Home/bin
C:> keytool -import -keystore ..\lib\security\cacerts -file <ca_certificate>.crt -alias
<label>
```

If prompted for a password, use: *changeit*. The keytool is located in the <Modeler installation directory>\jre\bin directory (or in the <Modeler installation directory>/SPSSModeler.app/Contents/PlugIns/jre/Contents/Home/bin directory on Mac).

4. Validate the server's key database with the root CA certificate using the following command:

```
gsk<ver>capicmd[_64} -cert -validate -db <filename>.kdb -stashed -label <label>
```

A successful validation is indicated by the returned message: OK.

Note: The commands explained above use a third-party CA root certificate that is in a binary format. If the certificate is in an ASCII format, use the `-format ascii` option.

The `-db` parameter specifies the name of the key database into which you import the third-party CA root certificate.

The `-label` parameter specifies the label to use for the third-party CA root certificate inside the key database file. The label you use here can be anything because it does not have any relation to the labels used in the IBM SPSS Modeler options.cfg file.

The `-file` parameter specifies the file that contains the third-party CA root certificate

Enable and configure SSL in IBM SPSS Deployment Manager

1. If installing a self-signed SSL certificate, copy the cacerts file that you created to the <Deployment Manager installation directory>\jre\lib\security directory. See the topic “[Installing a self-signed SSL certificate](#)” on page 44 for more information.
2. Start the server administration application (IBM SPSS Deployment Manager) and connect to the server.
3. On the configuration page, set **Secure Sockets Layer** to Yes.
4. In **SSL Public Key File**, specify the full path to the public key file.
5. In **SSL Private Key File**, specify the full path to the private key file.

Note: If the public and private keys are stored in one file, specify the same file in **SSL Public Key File** and **SSL Private Key File**.

6. From the menus choose:

File > Save

7. Restart the server service or daemon. When you restart, you will be prompted for the SSL password. On Windows, you can select **Remember this password** to store the password securely. This option eliminates the need to enter the password every time the server is started.

Installing unlimited strength encryption

The Java Runtime Environment shipped with the product has US export-strength encryption enabled. For enhanced security of your data, upgrading to unlimited-strength encryption is recommended. This procedure must be repeated for both client and server installations.

To install unlimited strength encryption

1. Download the [Unrestricted SDK JCE policy files](#) from IBM.com (select the files applicable to Java 8).

Note: You will need to login with your IBMid credentials in order to download the files.

2. Extract the unlimited jurisdiction policy files that are packaged in the compressed file. The compressed file contains a `US_export_policy.jar` file and a `local_policy.jar` file.
3. Back up the existing copies of `US_export_policy.jar` and `local_policy.jar` from the directory `jre/lib/security`.
4. Replace the existing copies of `US_export_policy.jar` and `local_policy.jar` files with the two files that you downloaded and extracted.
5. Restart IBM SPSS Modeler Client or Server as appropriate.

Instructing users to enable SSL

When users connect to the server through a client product, they need to enable SSL in the dialog box for connecting to the server.

Cognos SSL connection

To connect to a Cognos Analytics server with HTTPS and an SSL secured port, you must first change some of the Cognos internal and external dispatcher settings. For details on how to make the required changes, see the Cognos Server Configuration and Administration guide.

After you change the dispatcher settings, import the SSL certification that you created in Cognos into the SPSS Modeler JRE by following these steps:

1. In Cognos configuration, define a password for the IBM Cognos key store:
 - a. In the **Explorer** window, click **Cryptography > Cognos**.
 - b. In the **Properties** window, under **Encryption Key Settings**, set the Encryption key store password.
 - c. From the File menu, select Save.
 - d. From the Actions menu, select Restart.
2. From the command line, go to the `c10_location\bin` directory.
3. Set the `JAVA_HOME` environment variable to the Java™ Runtime Environment location used by the application server that is running Cognos. For example:

```
set JAVA_HOME=c11_location\bin\jre\<version>
```

4. From the command line, run the certificate tool. For example:

```
ThirdPartyCertificateTool.bat -E -T -r ca.cer -k ..\configuration\encryptkeypair\jEncKeystore  
-p <password>
```

5. Copy the `ca.cer` file to the SPSS Modeler Server location.
6. Open a command line and switch to the `<ModelerInstallationLocation>\jre\bin` folder.
7. Run the command to import the certificate. For example:

```
.\keytool -import -alias ca -file <Directory where ca.cer is located>\ca.cer  
-keystore "<ModelerInstallationLocation>\jre\lib\security\cacerts"
```

You can then use HTTPS and the SSL secured dispatcher to connect to Cognos. For example:

```
https://9.119.83.37:9343/p2pd/servlet/dispatch
```

Cognos TM1 SSL connection

To connect to Cognos TM1 with HTTPS and an SSL secured port, follow these steps:

1. Enable SSL on IBM Cognos TM1 Application Server. See the TM1 documentation at <https://www.ibm.com/docs/en/cognos-tm1/10.2.2?topic=configuration-usingssl-data-transmission-security>.
2. Download the `tm1server.pem` certification file. For example, if using Firefox:

- a. Open a browser window and enter your TM1 Server datasource URL such as `https://9.30.204.176:8010/api/v1/`. Then click **View Certificate** to open the server certification.
 - b. In the new Firefox browser window, click **PEM (cert)** to download the `tm1server.pem` file.
3. From the command line, go to the SPSS Modeler `jre\bin` directory and then run the following command to import the `tm1server.pem` file to the SPSS Modeler Server (modify the paths to match your environment, as needed):

```
C:\Program Files\IBM\SPSS\Modeler\18.3\jre\bin>
keytool.exe -import -alias tm1server -file C:
\Users\Administrator\Downloads\tm1server.pem -keystore ..\lib\security\cacerts
```

- a. Enter the default keystore password `changeit` if prompted.
 - b. When prompted whether to trust this certificate, enter `yes`.
4. Restart SPSS Modeler Client and SPSS Modeler Server.

You can now use HTTPS and the SSL secured port number to connect to Cognos TM1.

Configuring groups

An authenticated user typically belongs to one or more security groups, and when group-based configuration is enabled for SPSS Modeler Server, these groups can be used to permit or deny login to the server, or to customize the option settings for the user session.

Group configuration is supported in the following scenarios:

- In a *default* installation where the SPSS Modeler Server service runs under the Local System or root account and the user logs in with explicit credentials or using Single Sign-On (SSO): in this case the groups are the user's operating system security groups used to control file access, etc.
- In a *non-root* installation where the SPSS Modeler Server service runs under a non-privileged account and the user logs in using SSO: in this case the groups are the LDAP groups associated with the SSO principal. These groups are obtained from the LDAP security provider in IBM SPSS Collaboration and Deployment Services, so some additional configuration is required to enable this scenario. See [“Getting the SSO user's group membership”](#) on page 16 for more information.

If neither of these two scenarios apply, then the user's groups are not available and group configuration is not supported. In particular, in a *non-root* installation where the SPSS Modeler Server service runs under a non-privileged account and the user logs in using a user name and password, then the operating system groups are not available to the server and group configuration is not supported.

The principle of group-based configuration is that the option settings applied to a user's session can differ according to the user's group membership. These are the server-side settings normally read from the SPSS Modeler Server `options.cfg` file set identically for all sessions. The `options.cfg` file provides the defaults for all sessions, but there can be group-specific configuration files that override a subset of settings for particular sessions.

Group configuration allows for control of various settings, such as:

- Controlling file and DSN access
- Controlling resource usage

When the `group_configuration` option is enabled in `options.cfg`, IBM SPSS Modeler Server checks the `groups.cfg` file which controls who can log on to the server. The default is `N`. Following is a `groups.cfg` example that denies the `Test` group access to the server and allows the `Fraud` group access with a specified configuration. The asterisk allows all other groups access with the default configuration.

```
Test, DENY
Fraud, "groups/fraud.cfg"
*,
```

A specific group configuration, such as that for Fraud above, might restrict access to particular data sources or change resource settings (relating to SQL push back, memory usage, multi-threading, etc.) to enhance performance for members of that group.

The group configuration mechanism is designed to answer two questions:

1. Is the user allowed to use this instance of IBM SPSS Modeler Server?
2. If they are so allowed, then what configuration options do they get?

Regarding #2, the configuration options are those defined by `options.cfg`, and the default configuration refers to the settings in that file. The group mechanism allows you to override some of the default settings by specifying alternative configuration files for certain groups, where the settings in the group files take precedence over the defaults. The following parameters are supported in group configuration files. Note that there may be other parameters not listed here that can also be used in group configuration files.

```
sql_generation_enabled
db_udf_enabled
stream_rewriting_enabled
io_buffer_size
max_file_size
max_transfer_size
max_sql_string_length
default_sql_string_length
data_files_restricted
data_file_path
program_files_restricted
program_file_path
allow_modelling_memory_override
modelling_memory_limit_percentage
max_parallelism
enable_database_caching
sql_row_array_size
sql_data_sources_restricted
sql_data_source_path
memory_usage
sql_generation
sql_logging
sql_generation_logging
sql_log_native
sql_log_prettyprint
stream_rewriting
stream_rewriting_maximise_sql
date_baseline
date_2digit_baseline
time_rollover
date_format
time_format
decimal_separator
angles_in_radians
record_count_feedback_interval
record_count_suppress_input
decimal_places
column_width
cache_compression
enable_parallelism
database_caching
shell
use_bigint_for_count
trace_extension
```

Regarding #1, when group configuration is disabled, then everyone is allowed to use the server. When group configuration is enabled, then nobody is allowed to use it unless they are explicitly granted access in `groups.cfg`. So an empty `groups.cfg` file makes the server unusable by all. Typically, you add to `groups.cfg` the groups who should have access. For example:

```
A,
B,
C,
```


Optionally, for any group that you allow access, you can also specify a configuration file that overrides the default settings from `options.cfg`:

```
A, "a.cfg"  
B, "b.cfg"  
C, "c.cfg"
```

Any group for which you don't specify a configuration uses the default configuration, which comprises the settings from `options.cfg`.

The DENY option is allowed for more complex cases where a simple enumeration would grant more access than you really want. For example, you allocate a service for Fraud, but there are some developers who are also in the Fraud group who shouldn't have access. So you write:

```
devops, DENY  
fraud,
```

You don't need to specify a default DENY because everyone else is excluded by virtue of not being included.

Note that this mechanism is subsidiary to the O/S logon mechanism (LDAP, etc). The user must always log on first, and if the O/S denies them access then they never get this far. If they can log on, then their O/S group membership is used to determine the group configuration, and they may be denied access at that point.

Controlling DSN access by group

Multi-factor authentication (MFA) requires that users can be restricted in the set of ODBC data source names (DSNs) that they are allowed to access according to their group membership.

The scheme for accomplishing this is similar to the existing scheme for file access. Two configuration settings are available in `options.cfg`:

```
sql_data_sources_restricted, N  
sql_data_source_path, ""
```

If `sql_data_sources_restricted` is set to Y, then the user is limited to the DSNs listed in the associated path. DSNs are separated by the standard path separator character ; (semi-colon) on Windows and : (colon) on UNIX. For example, on Windows:

```
sql_data_sources_restricted, Y  
sql_data_source_path, "Fraud - Analytic;Fraud - Operational"
```

When this restriction is enabled, it has the following results:

- When a user browses for data sources (for example, from the ODBC connection dialog, or when using the PSAPI Session `getServerDataSourceNames` API), instead of being presented with all the DSNs defined on the server system, the user will only see the subset of DSNs that is defined in the `options.cfg` path. Note that the path may contain DSNs that are not defined on the server, and these are ignored -- the user will not see those names.
- If a user constructs an ODBC node (or any node using an ODBC connection) that uses a script or PSAPI and the user specifies a DSN that is not included in the `options.cfg` path, the node will not run and the user will be presented with an error similar to **Access denied to data source: <X>**.

The data source path can include the PATH, GROUP and USER insertions described elsewhere for file paths. The PATH insertion allows the path to be constructed incrementally according to the user's group membership when group-based configuration is used. There might also be circumstances where it makes sense to name a DSN after the group which owns it.

Building on the previous example, if access to the Fraud data sources is allowed only for members of the Fraud Analysts group, then the site can enable group configuration and create a configuration specific to the Fraud Analysts containing at least this line:

```
sql_data_source_path, "${PATH};Fraud - Analytic;Fraud - Operational"
```

The addition of the PATH prefix in this example ensures that the fraud analysts are still able to access other data sources allowed to everyone, or to other groups of which they are members.

Log files

IBM SPSS Modeler Server keeps a record of its important actions in a log file called `server_logging.log`. On UNIX, this file is in the `log` folder in the installation directory. On Windows, this file is in: `%ALLUSERSPROFILE%/IBM/SPSS/Modeler Server/<version>/log`.

The settings that control how logging is carried out in your installation are contained in the `log4cxx.properties` file.

Change the location of the log file

The default location of the log file is set, in the `log4cxx.properties` file, as:

```
log4j.appender.MainLog.File=${app_log_location}/${PROFILE_NAME}/${app_type}logging.log
```

To change the log file location, edit this entry.

Enable tracing

There may be occasions when you require finer detail than just a basic list of information that shows the main actions; for example, this detail might be asked for by your support staff to help identify an issue. In these situations, you can amend the log to provide more detailed trace information.

To enable tracing, in the `log4cxx.properties` file, disable the line `log4j.rootLogger=INFO, MainLog, ConsoleLog` and enable the following line in its place: `log4j.rootLogger=TRACE, MainLog, TraceLog`

To change the location of the trace log, edit the entry:

```
log4j.appender.TraceLog.File=${app_log_location}/${PROFILE_NAME}/${app_type}tracing_${PROCESS_ID}.log
```

Amend logging options

The `log4cxx.properties` file contains the controls that define how various events are logged. These controls are normally set to either **INFO** to record actions in the log file, or **WARN** to notify the user of a potential problem. If you are using the log file to identify potential errors, you can also set some of the controls to **TRACE**.

Control the size of the log file

By default, the log file continues to grow in size every time you use SPSS Modeler Server. To prevent the log becoming too large, you can either set it to be started from scratch every day, or define a size limit for it.

To set the log to be started as a new log every day, in the `log4cxx.properties` file, use the following entries:

```
log4j.appender.MainLog=org.apache.log4j.DailyRollingFileAppender
```

```
log4j.appender.MainLog.DatePattern='.'yyyy-MM-dd
```

Alternatively, to define a size limit for the log (for example 8 Mb), in the `log4cxx.properties` file, use the following entries:

```
log4j.appender.MainLog=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.MainLog.MaxFileSize=8MB
```

Client log file

Note that you can also enable logging for IBM SPSS Modeler client. To do so, open the file `log4j2.xml` in a text editor and change `level="info"` to `level="debug"` in this line:

```
<Logger name="com.spss" additivity="false" level="info">
```

On Mac, the default client log file location is `/Applications/IBM/SPSS/Modeler/18.5/Resources/log/`. On Windows, the default location is `${env.USERPROFILE}/BM/SPSS/Modeler/18.5/log`, where `env.USERPROFILE` is usually `C:\Users%username%` with `%username%` being the proper folder name.

Chapter 5. Performance Overview

Real performance in analyzing data is affected by a number of factors, from server and database configuration to the ordering of individual nodes within a stream. In general, you can obtain the best performance by doing the following:

- Store your data in a DBMS, and use SQL generation and optimization whenever possible.
- Use hardware that meets or exceeds the recommendations given in [Chapter 2, “Architecture and Hardware Recommendations,”](#) on page 5.
- Ensure that the client and server performance and optimization settings are properly configured. Note that when SPSS Modeler is connected to an SPSS Modeler Server installation, the server performance and optimization settings override the client equivalents.
- Design streams for maximum performance.

More information about each of these performance factors is available in the following sections.

Server performance and optimization settings

Certain IBM SPSS Modeler Server settings can be configured to optimize performance. You can adjust these settings using the IBM SPSS Modeler Administration Console interface included in IBM SPSS Deployment Manager. See the topic [“IBM SPSS Modeler Server Administration”](#) on page 31 for more information.

The settings are grouped under the heading **Performance and Optimization** in the IBM SPSS Modeler Administration Console configuration window. The settings are preconfigured for optimal performance for most installations. However, you may need to adjust them depending on your particular hardware, the size of your data sets, and the contents of your streams. See the topic [“Performance/Optimization”](#) on page 35 for more information.

Client Performance and Optimization Settings

The client performance and optimization settings are available from the Options tab of the Stream Properties dialog box. To display these options, choose the following from the client menu.

Tools > Stream Properties > Options > Optimization

You can use the Optimization settings to optimize stream performance. Note that the performance and optimization settings on IBM SPSS Modeler Server (if used) override any equivalent settings in the client. If these settings are disabled in the server, then the client cannot enable them. But if they are enabled in the server, the client can choose to disable them.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > About > Additional Details

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

See [“Connecting to IBM SPSS Modeler Server”](#) on page 11 for more information.

Note: Whether SQL pushback and optimization are supported depends on the type of database in use. For the latest information on which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the corporate Support site at <http://www.ibm.com/support>.

Enable stream rewriting. Select this option to enable stream rewriting in IBM SPSS Modeler. Four types of rewriting are available, and you can select one or more of them. Stream rewriting reorders the nodes in a stream behind the scenes for more efficient operation, without altering stream semantics.

- **Optimize SQL generation.** This option enables nodes to be reordered within the stream so that more operations can be pushed back using SQL generation for execution in the database. When it finds a node that cannot be rendered into SQL, the optimizer will look ahead to see if there are any downstream nodes that can be rendered into SQL and safely moved in front of the problem node without affecting the stream semantics. Not only can the database perform operations more efficiently than IBM SPSS Modeler, but such pushbacks act to reduce the size of the data set that is returned to IBM SPSS Modeler for processing. This, in turn, can reduce network traffic and speed stream operations. Note that the **Generate SQL** check box must be selected for SQL optimization to have any effect.
- **Optimize CLEM expression.** This option enables the optimizer to search for CLEM expressions that can be preprocessed before the stream is run, in order to increase the processing speed. As a simple example, if you have an expression such as *log(salary)*, the optimizer would calculate the actual salary value and pass that on for processing. This can be used both to improve SQL pushback and IBM SPSS Modeler Server performance.
- **Optimize syntax execution.** This method of stream rewriting increases the efficiency of operations that incorporate more than one node containing IBM SPSS Statistics syntax. Optimization is achieved by combining the syntax commands into a single operation, instead of running each as a separate operation.
- **Optimize other execution.** This method of stream rewriting increases the efficiency of operations that cannot be delegated to the database. Optimization is achieved by reducing the amount of data in the stream as early as possible. While maintaining data integrity, the stream is rewritten to push operations closer to the data source, thus reducing data downstream for costly operations, such as joins.

Enable parallel processing. When running on a computer with multiple processors, this option allows the system to balance the load across those processors, which may result in faster performance. Use of multiple nodes or use of the following individual nodes may benefit from parallel processing: C5.0, Merge (by key), Sort, Bin (rank and tile methods), and Aggregate (using one or more key fields).

Generate SQL. Select this option to enable SQL generation, allowing stream operations to be pushed back to the database by using SQL code to generate execution processes, which may improve performance. To further improve performance, **Optimize SQL generation** can also be selected to maximize the number of operations pushed back to the database. When operations for a node have been pushed back to the database, the node will be highlighted in purple when the stream is run.

- **Database caching.** For streams that generate SQL to be executed in the database, data can be cached midstream to a temporary table in the database rather than to the file system. When combined with SQL optimization, this may result in significant gains in performance. For example, the output from a stream that merges multiple tables to create a data mining view may be cached and reused as needed. With database caching enabled, simply right-click any nonterminal node to cache data at that point, and the cache is automatically created directly in the database the next time the stream is run. This allows SQL to be generated for downstream nodes, further improving performance. Alternatively, this option can be disabled if needed, such as when policies or permissions preclude data being written to the database. If database caching or SQL optimization is not enabled, the cache will be written to the file system instead.
- **Use relaxed conversion.** This option enables the conversion of data from either strings to numbers, or numbers to strings, if stored in a suitable format. For example, if the data is kept in the database as a string, but actually contains a meaningful number, the data can be converted for use when the pushback occurs.

Note: Due to minor differences in SQL implementation, streams run in a database may return slightly different results from those returned when run in IBM SPSS Modeler. For similar reasons, these differences may also vary depending on the database vendor.

Database Usage and Optimization

Database server. If possible, create a dedicated database instance for data mining so that the production server is not impacted by IBM SPSS Modeler queries. SQL statements generated by IBM SPSS Modeler can be demanding—multiple tasks on the IBM SPSS Modeler Server machine can be executing SQL in the same database.

In-database mining. Many database vendors provide data mining extensions for their products. These extensions allow data mining activities (such as model-building or scoring) to run within the database server, or within a separate dedicated server. IBM SPSS Modeler’s in-database mining features complement and extend its SQL generation capability, providing a way to drive the vendor-specific database extensions. In some cases, taking this approach avoids the potentially expensive overhead of data transfer between IBM SPSS Modeler and the database. Database caching can further increase the benefits. For more information, see the file *DatabaseMiningGuide.pdf*, which is available as a part of your downloaded eImage.

SQL Optimization

For best performance, you should always try to maximize the amount of SQL generated to exploit the performance and scalability of the database. Only the parts of the stream that cannot be compiled to SQL should be executed within IBM SPSS Modeler Server. For more information, see [Chapter 6, “SQL optimization,”](#) on page 57.

Uploading File-Based Data

Data that is not stored in a database cannot benefit from SQL optimization. If the data you want to analyze is not already in a database, you can upload it using a Database Output node. You can also use this node to store intermediate data sets from data preparation and the results of deployment.

IBM SPSS Modeler can interface with the external loaders for many common database systems. Several scripts are included with the software and are available (with documentation) in the */scripts* subdirectory under your IBM SPSS Modeler installation folder.

The following table shows the potential performance benefit of bulk-loading. The figures show the elapsed time to export 250,000 records and 21 fields to an Oracle database. The external loader was Oracle’s `sqlldr` utility.

Export option	Time (in seconds)
Default (ODBC)	409
Bulk-load via ODBC	52
Bulk-load via external loader	33

Stream performance

Many factors can impact how your SPSS Modeler streams perform.

Keep these general tips in mind:

- Where possible, consider minimizing the size of your data by limiting processing to only those fields that are needed by using Filter nodes and the Filter tab in source nodes.
- Leverage in-database processing capability whenever possible to reduce the amount of data pulled in to SPSS Modeler.
- Minimize the network distance between your IBM SPSS Modeler Server and the source data.
- Certain data sources require more overhead than others. For example, the Excel source node takes longer to access the same data than a CSV file. XML data is inherently wasteful and shouldn't be used for storing large amounts of data.
- If using Python-based nodes or R-based nodes, note that there are internal data transfers that must take place. This can sometimes slow processing.
- Accomplishing your tasks with the fewest number of nodes is usually preferable to more nodes.
- Use Type nodes only when necessary. This is especially true when Hadoop is the data source because each Type node processes the entire data flow.

- Certain statistical modeling nodes might be slow, especially with data sets that have many categorical fields.
- Changing the order of nodes can influence processing speed, so experiment with node order. For example, if you have a stream with nodes that reduce data by subsetting or reducing the number of fields, move them as early in the stream as possible.
- If a modeling node you're using has a corresponding -AS version, use the -AS node instead because it's multi-threaded and can improve processing.

Chapter 6. SQL optimization

One of the most powerful capabilities of IBM SPSS Modeler is the ability to perform many data preparation and mining operations directly in the database. By generating SQL code that can be pushed back to the database for execution, many operations, such as sampling, sorting, deriving new fields, and certain types of graphing, can be performed in the database rather than on the IBM SPSS Modeler or IBM SPSS Modeler Server computer. When you are working with large data sets, these *pushbacks* can dramatically enhance performance in several ways:

- By reducing the size of the result set to be transferred from the DBMS to IBM SPSS Modeler. When large result sets are read through an ODBC driver, network I/O or driver inefficiencies can result. For this reason, the operations that benefit most from SQL optimization are row and column selection and aggregation (Select, Sample, Aggregate nodes). These operations typically reduce the size of the data set to be transferred. Data can also be cached to a temporary table in the database at critical points in the stream (after a Merge or Select node, for example) to further improve performance.
- By using the performance and scalability of the database. Efficiency is increased because a DBMS can often take advantage of parallel processing, more powerful hardware, more sophisticated management of disk storage, and the presence of indexes.

Given these advantages, IBM SPSS Modeler is designed to maximize the amount of SQL generated by each stream so that only those operations that can't be compiled to SQL will be run by IBM SPSS Modeler Server. Because of limitations in what can be expressed in standard SQL (SQL-92), however, certain operations may not be supported. For more information, see [“Tips for maximizing SQL generation”](#) on page 61.

Note: Keep the following information in mind when working with SQL:

- Because of minor differences in SQL implementation, streams that run in a database can return slightly different results when they run in IBM SPSS Modeler. These differences may also vary depending on the database vendor, for similar reasons. For example, depending on the database configuration for case sensitivity in string comparison and string collation, IBM SPSS Modeler streams that run by using SQL pushback might produce different results from those that run without SQL pushback. Contact your database administrator for advice on configuring your database. To maximize compatibility with IBM SPSS Modeler, we recommend making sure database string comparisons are case-sensitive.
- Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity is enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, from the IBM SPSS Modeler menu, go to:

– **Help > About > Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

See [“Connecting to IBM SPSS Modeler Server”](#) on page 11 for more information.

- When using IBM SPSS Modeler to generate SQL, the result that uses SQL push back might not be consistent with IBM SPSS Modeler native on some platforms (Linux/zLinux, for example). The reason is that floating point is handled differently on different platforms.

Note: When you run streams in a Netezza database, date and time details are taken from that database. This behavior might differ from your local or IBM SPSS Modeler Server date and time if, for example, the database is on a machine that is located in a different country or time zone.

Database requirements

For more information about which databases and ODBC drivers are supported and tested for use with IBM SPSS Modeler, see the product compatibility matrices on the corporate Support site at <http://www.ibm.com/support>.

You might gain more performance improvements by using database modeling.

ODBC driver setup

To ensure that time details (such as HH:MM:SS) are processed correctly when using SQL 2012 on Windows 32bit systems, when setting up your ODBC SQL Server Wire Protocol Driver, select both the **Enable Quoted Identifiers** and **Fetch TWFS as Time** options.

How SQL generation works

The initial fragments of a stream leading from the database source nodes are the main targets for SQL generation. When a node is encountered that cannot be compiled to SQL, the data are extracted from the database and subsequent processing is performed by IBM SPSS Modeler Server.

During stream preparation and prior to execution, the SQL generation process happens as follows:

- The server reorders streams to move downstream nodes into the “SQL zone” where it can be proven safe to do so. (This feature can be disabled on the server.)
- Working from the source nodes toward the terminal nodes, SQL expressions are constructed incrementally. This phase stops when a node is encountered that cannot be converted to SQL or when the terminal node (for example, Table node or Graph node) is converted to SQL. At the end of this phase, each node is labeled with an SQL statement if the node and its predecessors have an SQL equivalent.
- Working from the nodes with the most complicated SQL equivalents back toward the source nodes, the SQL is checked for validity. The SQL that was successfully validated is chosen for execution.
- Nodes for which all operations have generated SQL are highlighted in purple on the stream canvas. Based on the results, you may want to further reorganize your stream where appropriate to take full advantage of database execution. See the topic [“Tips for maximizing SQL generation”](#) on page 61 for more information.

Where Improvements Occur

SQL optimization improves performance in a number of data operations:

- **Joins (merge by key).** Join operations can increase optimization within databases.
- **Aggregation.** The Aggregate, Distribution, and Web nodes all use aggregation to produce their results. Summarized data uses considerably less bandwidth than the original data.
- **Selection.** Choosing records based on certain criteria reduces the quantity of records.
- **Sorting.** Sorting records is a resource-intensive activity that is performed more efficiently in a database.
- **Field derivation.** New fields are generated more efficiently in a database.
- **Field projection.** IBM SPSS Modeler Server extracts only fields that are required for subsequent processing from the database, which minimizes bandwidth and memory requirements. The same is also true for superfluous fields in flat files: although the server must read the superfluous fields, it does not allocate any storage for them.
- **Scoring.** SQL can be generated from decision trees, rulesets, linear regression, and factor-generated models.

SQL Generation Example

The following stream joins three database tables by key operations and then performs an aggregation and sort.

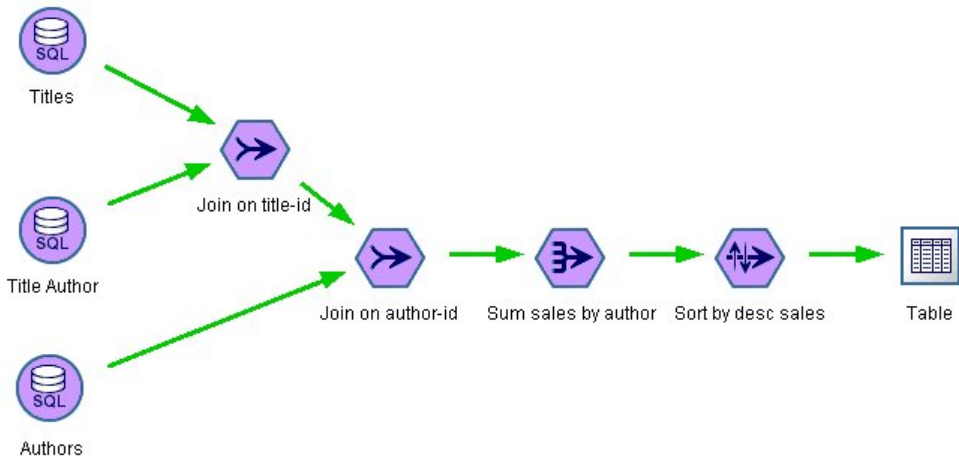


Figure 3. Optimized stream with purple nodes indicating SQL pushbacks (operations performed in database)

Generated SQL

The generated SQL for this stream is as follows:

```

SELECT
    T2. au_lname AS C0,
    T2. au_fname AS C1,
    SUM({fn CONVERT(T0. ytd_sales ,SQL_BIGINT)}) AS C2
FROM
    dbo . titles T0,
    dbo . titleauthor T1,
    dbo . authors T2
WHERE
    (T0. title_id = T1. title_id )
    AND (T1. au_id = T2. au_id )
GROUP BY T2. au_lname ,T2. au_fname
ORDER BY 3 DESC

```

Executing the Stream

When the stream is terminated with a database export node, it is possible for the whole stream to be executed in the database.

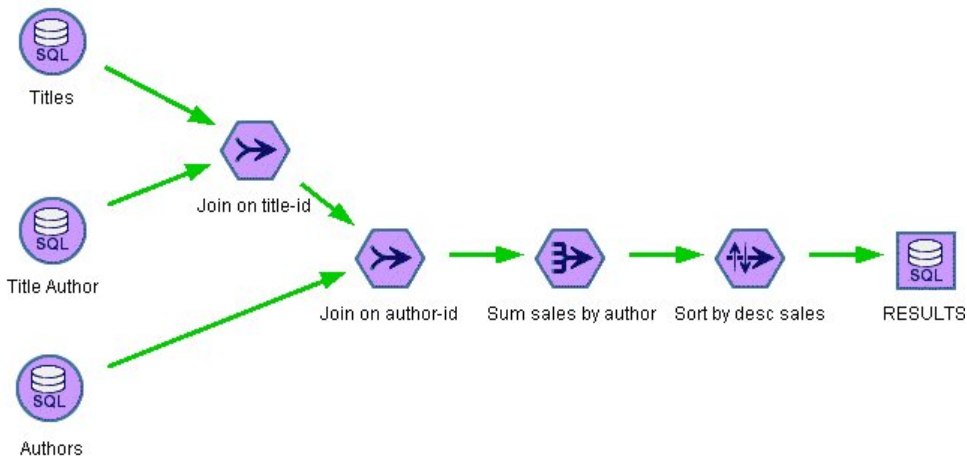


Figure 4. Entire stream executed in database

Configuring SQL Optimization

1. Install an ODBC driver and configure a data source for the database you want to use. See the topic [“Data Access”](#) on page 8 for more information.
2. Create a stream that uses a source node to pull data from that database.
3. Check to be sure that SQL generation is enabled on the client and server, if applicable. It is enabled by default for both.

To Enable SQL Optimization on the Client

1. From the Tools menu, choose **Stream Properties > Options**.
2. Click the Optimization tab. Select **Generate SQL** to enable SQL optimization. Optionally, you can select other settings to improve performance. See the topic [“Client Performance and Optimization Settings”](#) on page 53 for more information.

To Enable SQL Optimization on the Server

Because server settings override any specifications made on the client, the server configuration settings **Stream Rewriting** and **Automatic SQL Generation** must both be turned on. For more information about how to change IBM SPSS Modeler Server settings, see the section [“Performance/Optimization”](#) on page 35. Note that if these settings are disabled in the server, then the client cannot enable them. But if they are enabled in the server, the client can choose to disable them.

To Enable Optimization When Scoring Models

For purposes of scoring, SQL generation must be enabled separately for each modeling node, regardless of any server or client-level settings. This is done because some models generate extremely complex SQL expressions that may not be evaluated effectively within the database. The database may report errors when trying to execute the generated SQL, due to the size or complexity of the SQL.

A certain amount of trial-and-error may be needed to determine whether SQL generation improves performance for a given model. This is done on the Settings tab after adding a generated model to a stream.

Previewing Generated SQL

You can preview generated SQL in the message log before executing it in the database. This may be useful for debugging purposes, and it allows you to export the generated SQL to edit or run in the database at a future date. It also indicates which nodes will be pushed back to the database, which may help you determine whether the stream can be reordered to improve performance.

1. Make sure that **Display SQL in the messages log during stream execution** and **Display SQL generation details in the messages log during stream preparation** are selected in the User Options dialog box. See the topic [“Client Performance and Optimization Settings”](#) on page 53 for more information.
2. On the stream canvas, select the node or stream that you want to preview.
3. Click the **Preview SQL** button on the toolbar.

All nodes for which SQL is generated (and that will be pushed back to the database when the stream is executed) are colored purple on the stream canvas.

4. To preview the generated SQL, from the menus choose:

Tools > Stream Properties > Messages...

Viewing SQL for Model Nuggets

For some models, SQL for the model nugget can be generated, pushing back the model scoring stage to the database. The main use of this feature is not to improve performance, but to allow streams containing these nuggets to have their full SQL pushed back. See the topic [“Nodes supporting SQL generation”](#) on page 62 for more information.

To view the SQL for a model nugget that supports SQL generation:

1. Select the Settings tab on the model nugget.
2. Choose one of the options **Generate with (no) missing value support** or **Generate SQL for this model**, as appropriate.
3. In the model nugget menu, choose:
File > Export SQL
4. Save the file.
5. Open the file to view the SQL.

Tips for maximizing SQL generation

To get the best performance boost from SQL optimization, pay attention to the following items.

Stream order. SQL generation may be halted when the function of the node has no semantic equivalent in SQL because IBM SPSS Modeler’s data-mining functionality is richer than the traditional data-processing operations supported by standard SQL. When this happens, SQL generation is also suppressed for any downstream nodes. Therefore, you may be able to significantly improve performance by reordering nodes to put operations that halt SQL as far downstream as possible. The SQL optimizer can do a certain amount of reordering automatically (just make sure stream rewriting is enabled), but further improvements may be possible. A good candidate for this is the Select node, which can often be brought forward. See the topic [“Nodes supporting SQL generation” on page 62](#) for more information.

CLEM expressions. If a stream cannot be reordered, you may be able to change node options or CLEM expressions or otherwise recast the way the operation is performed, so that it no longer inhibits SQL generation. Derive, Select, and similar nodes can commonly be rendered into SQL, provided that all of the CLEM expression operators have SQL equivalents. Most operators can be rendered, but there are a number of operators that inhibit SQL generation (in particular, the sequence functions [“@ functions”]). Sometimes generation is halted because the generated query has become too complex for the database to handle. See the topic [“CLEM Expressions and Operators Supporting SQL Generation” on page 66](#) for more information.

Multiple source nodes. Where a stream has multiple database source nodes, SQL generation is applied to each input branch independently. If generation is halted on one branch, it can continue on another. Where two branches merge (and both branches can be expressed in SQL up to the merge), the merge itself can often be replaced with a database join, and generation can be continued downstream.

Database algorithms. Model estimation is always performed on IBM SPSS Modeler Server rather than the database, except when using database-native algorithms from Microsoft, IBM, or Oracle.

Scoring models. In-database scoring is supported for some models by rendering the generated model into SQL. However, some models generate extremely complex SQL expressions that aren’t always evaluated effectively within the database. For this reason, SQL generation must be enabled separately for each model node. If you find that a model node is inhibiting SQL generation, go to the Settings tab on the node dialog box and select **Generate SQL for this model** (with some models, you may have additional options controlling generation). Run tests to confirm that the option is beneficial for your application. See the topic [“Nodes supporting SQL generation” on page 62](#) for more information.

When testing modeling nodes to see if SQL generation for models works effectively, we recommend first saving all streams from IBM SPSS Modeler. Some database systems may hang while trying to process the (potentially complex) generated SQL, requiring IBM SPSS Modeler to be closed from the Windows task manager.

Database caching. If you are using a node cache to save data at critical points in the stream (for example, following a Merge or Aggregate node), make sure that database caching is enabled along with SQL optimization. This will allow data to be cached to a temporary table in the database (rather than the file system) in most cases. See the topic [“Configuring SQL Optimization” on page 60](#) for more information.

Vendor-specific SQL. Most of the generated SQL is standards-conforming (SQL-92), but some nonstandard, vendor-specific features are exploited where practical. The degree of SQL optimization can vary, depending on the database source.

SQL configuration option. By default, SPSS Modeler considers SQL queries written in an ODBC source node to be non-replayable, meaning that the query is considered to return different results when being executed multiple times. However, in some cases, this may prevent SPSS Modeler from generating SQL for downstream nodes. You can override this behavior by changing the following value to Y in the `odbc-db2-custom-properties.cfg`. The file is located in the SPSS Modeler config directory.

```
assume_custom_sql_replayable, Y
```

Nodes supporting SQL generation

The following tables show nodes representing data-mining operations that support SQL generation. With the exception of the database modeling nodes, if a node does not appear in these tables, it does not support SQL generation.

You can preview the SQL that is generated before executing it. See the topic [“Previewing Generated SQL”](#) on page 60 for more information.

<i>Table 4. Sources</i>	
Node supporting SQL generation	Notes
Database	This node is used to specify tables and views to be used for further analysis. This node enables entry of SQL queries. Avoid results sets with duplicate column names. See the topic “Writing SQL Queries” on page 69 for more information.

<i>Table 5. Record operations</i>	
Node supporting SQL generation	Notes
Select	Supports generation only if SQL generation for the select expression itself is supported (see expressions below). If any fields have nulls, SQL generation does not give the same results for discard as are given in native IBM SPSS Modeler.
Sample	Simple sampling supports SQL generation to varying degrees depending on the database. See Table 6 on page 63 .
Aggregate	SQL generation support for aggregation depends on the data storage type. See Table 7 on page 64 .
RFM Aggregate	Supports generation except if saving the date of the second or third most recent transactions, or if only including recent transactions. However, including recent transactions does work if the <code>datetime_date(YEAR, MONTH, DAY)</code> function is pushed back.
Sort	

Table 5. Record operations (continued)

Node supporting SQL generation	Notes
Merge	<p>No SQL generated for merge by order.</p> <p>Merge by key with full or partial outer join is only supported if the database/driver supports it. Non-matching input fields can be renamed by means of a Filter node, or the Filter tab of a source node.</p> <p>Supports SQL generation for merge by condition.</p> <p>For all types of merge, SQL_SP_EXISTS is not supported if inputs originate in different databases.</p>
Append	<p>Supports generation if inputs are unsorted.</p> <p>Note: SQL optimization is only possible when your inputs have the same number of columns.</p>
Distinct	<p>A Distinct node with the (default) mode Create a composite record for each group selected does not support SQL optimization.</p>

Table 6. SQL generation support in the Sample node for simple sampling

Mode	Sample	Max size	Seed	Db2 for z/OS	Db2 for OS/400	Db2 for Win/UNIX	Netezza	Oracle	SQL Server	Teradata	
Include	First	n/a		Y	Y	Y	Y	Y	Y	Y	
	1-in-n	off		Y	Y	Y	Y	Y		Y	
		max		Y	Y	Y	Y	Y		Y	
	Random %	off	off	off	Y		Y	Y	Y		Y
			on	Y		Y		Y			
		max	off	off	Y		Y	Y	Y		Y
on			Y	Y		Y		Y			
Discard	First	off					Y	Y			
		max					Y	Y			
	1-in-n	off		Y	Y	Y	Y	Y		Y	
		max		Y	Y	Y	Y	Y		Y	
	Random %	off	off	off	Y		Y	Y	Y		Y
			on	Y		Y		Y			
		max	off	off	Y		Y	Y	Y		Y
			on	Y	Y		Y		Y		

Table 7. SQL generation support in the Aggregate node

Storage	Sum	Mean	Min	Max	SDev	Median	Count	Variance	Percentile
Integer	Y	Y	Y	Y	Y	Y*	Y	Y	Y*
Real	Y	Y	Y	Y	Y	Y*	Y	Y	Y*
Date			Y	Y		Y*	Y		Y*
Time			Y	Y		Y*	Y		Y*
Timestamp			Y	Y		Y*	Y		Y*
String			Y	Y		Y*	Y		Y*

* Median and Percentile are supported on Oracle.

Table 8. Field operations

Node supporting SQL generation	Notes
Type	Supports SQL generation if the Type node is instantiated and no ABORT or WARN type checking is specified.
Filter	
Derive	Supports SQL generation if SQL generated for the derive expression is supported (see expressions below).
Ensemble	Supports SQL generation for Continuous targets. For other targets, supports generation only if the "Highest confidence wins" ensemble method is used.
Filler	Supports SQL generation if the SQL generated for the derive expression is supported (see expressions below).
Anonymize	Supports SQL generation for Continuous targets, and partial SQL generation for Nominal and Flag targets.
Reclassify	
Binning	Supports SQL generation if the "Tiles (equal count)" binning method is used and the "Read from Bin Values tab if available" option is selected. Note: Due to differences in the way that bin boundaries are calculated (this is caused by the nature of the distribution of data in bin fields), you might see differences in the binning output when comparing normal stream execution results and SQL pushback results. To avoid this, use the Record count tiling method, and either Add to next or Keep in current tiles to obtain the closest match between the two methods of stream execution.
RFM Analysis	Supports SQL generation if the "Read from Bin Values tab if available" option is selected, but downstream nodes will not support it.
Partition	Supports SQL generation to assign records to partitions.
SetToFlag	
Restructure	

<i>Table 9. Graphs</i>	
Node supporting SQL generation	Notes
Graphboard	SQL generation is supported for the following graph types: Area, 3-D Area, Bar, 3-D Bar, Bar of Counts, Heat map, Pie, 3-D Pie, Pie of Counts. For Histograms, SQL generation is supported for categorical data only. SQL generation is not supported for Animation in Graphboard.
Distribution	
Web	
Evaluation	

For some models, SQL for the model nugget can be generated, pushing back the model scoring stage to the database. The main use of this feature is not to improve performance, but to allow streams containing these nuggets to have their full SQL pushed back. See the topic [“Viewing SQL for Model Nuggets”](#) on page 60 for more information.

<i>Table 10. Model nuggets</i>	
Model nugget supporting SQL generation	Notes
C&R Tree	Supports SQL generation for the single tree option, but not for the boosting, bagging or large dataset options.
QUEST	
CHAID	
C5.0	
Decision List	
Linear	Supports SQL generation for the standard model option, but not for the boosting, bagging or large dataset options.
Neural Net	Supports SQL generation for the standard model option (Multilayer Perceptron only), but not for the boosting, bagging or large dataset options.
PCA/Factor	
Logistic	Supports SQL generation for Multinomial procedure but not Binomial. For Multinomial, generation is not supported when confidences are selected, unless the target type is Flag.
Generated Rulesets	
Auto Classifier	If a User Defined Function (UDF) scoring adapter is enabled, these nuggets support SQL pushback. In addition, If either SQL generation for Continuous targets, or the "Highest confidence wins" ensemble method are used, these nuggets support further pushback downstream.
Auto Numeric	

<i>Table 11. Output</i>	
Node supporting SQL generation	Notes
Table	Supports generation if SQL generation is supported for highlight expression (see expressions below).

Node supporting SQL generation	Notes
Matrix	Supports generation except if "All numerics" is selected for the Fields option.
Analysis	Supports generation, depending on the options selected.
Transform	
Graphs	SQL Pushback is not supported if animation is enabled in a Graphboard node.
Statistics	Supports generation if the Correlate option is not used.
Report	
Set Globals	

Node supporting SQL generation	Notes
Database	
Publisher	The published stream will contain generated SQL.

CLEM Expressions and Operators Supporting SQL Generation

The following tables show the mathematical operations and expressions that support SQL generation and are often used during data mining. Operations absent from these tables do not support SQL generation in the current release.

Operation supporting SQL generation	Notes
+	
-	
/	
*	
><	Used to concatenate strings.

Operation supporting SQL generation	Notes
=	
/=	Used to specify "not equal."
>	
>=	
<	

Table 14. Relational operators (continued)

Operation supporting SQL generation	Notes
<=	

Table 15. Functions

Operation supporting SQL generation	Notes
abs	
allbutfirst	
allbutlast	
and	
arccos	
arcsin	
arctan	
arctanh	
cos	
div	
exp	
fracof	
hasstartstring	
hassubstring	
integer	
intof	
isaplhacode	
islowercode	
isnumbercode	
isstartstring	
issubstring	
isuppercode	
last	
length	
locchar	
log	
log10	
lowertoupper	
max	

Table 15. Functions (continued)

Operation supporting SQL generation	Notes
member	
min	
negate	
not	
number	
or	
pi	
real	
rem	
round	
sign	
sin	
sqrt	
string	
strmember	
subscrs	
substring	
substring_between	
uppertolower	
to_string	

Table 16. Special functions

Operation supporting SQL generation	Notes
@NULL	
@GLOBAL_AVE	The special global functions are used to retrieve global values computed by the Set Globals node.
@GLOBAL_SUM	
@GLOBAL_MAX	
@GLOBAL_MEAN	
@GLOBAL_MIN	
@GLOBALSDEV	

Table 17. Aggregate functions

Operation supporting SQL generation	Notes
Sum	
Mean	
Min	
Max	
Count	
SDev	

Using SQL Functions in CLEM Expressions

The @SQLFN function can be used to add named SQL functions within CLEM expressions, for purposes of database execution only. This can be useful in special cases where proprietary SQL or other vendor-specific customizations are required.

Use of this function is not covered by the standard IBM SPSS Modeler support agreement, since execution relies on external database components beyond the control of IBM Corp., but may be deployed in special cases, typically as part of a Services engagement. Contact <http://www.ibm.com/software/analytics/spss/services/> for more information if necessary.

Writing SQL Queries

When using the Database node, you should pay special attention to any SQL queries that result in a dataset with duplicate column names. These duplicate names often prevent SQL optimization for any downstream nodes.

IBM SPSS Modeler uses nested SELECT statements to push back SQL for streams that use an SQL query in the Database source node. In other words, the stream nests the query specified in the Database source node inside of one or more SELECT statements generated during the optimization of downstream nodes. Therefore, if the result set of a query contains duplicate column names, the statement cannot be nested by the RDBMS. Nesting difficulties occur most often during a table join where a column with the same name is selected in more than one of the joined tables. For example, consider this query in the source node:

```
SELECT e.ID, e.LAST_NAME, d.*
FROM EMP e RIGHT OUTER JOIN
DEPT d ON e.ID = d.ID;
```

The query will prevent subsequent SQL optimization, since this SELECT statement would result in a dataset with two columns called ID.

In order to allow full SQL optimization, you should be more explicit when writing SQL queries and specify column aliases when a situation with duplicate column names arises. The statement below illustrates a more explicit query:

```
SELECT e.ID AS ID1, e.LAST_NAME, d.*
FROM EMP e RIGHT OUTER JOIN
DEPT d ON e.ID = d.ID;
```

Scoring adapter for Teradata - duplicate rows

The IBM SPSS Modeler Server Scoring Adapter for Teradata expects no identical rows in its input data. Teradata does not allow the existence of two identical rows in a table. However, duplicate rows can happen when joining tables, or when the user uses only part of the fields of a table as input. These duplicate rows will lead to an incorrect number of records after a cartesian join.

Appendix A. Configuring Oracle for UNIX Platforms

Configuring Oracle for SQL Optimization

When running IBM SPSS Modeler Server on UNIX platforms and reading from an Oracle database, consider the following tips to ensure that generated SQL is being fully optimized within the database.

Proper Locale Specification

When running IBM SPSS Modeler Server in a locale other than that shipped with the Connect ODBC drivers, you should reconfigure the machine to enhance SQL optimization. Connect ODBC drivers ship only with the *en_US* locale files. Consequently, if the IBM SPSS Modeler Server machine is running in a different locale or if the shell in which IBM SPSS Modeler Server was started did not have the locale fully defined, generated SQL may not be fully optimized within Oracle. The reasons are as follows:

- IBM SPSS Modeler Server uses the ODBC locale files corresponding to the locale in which it is running to translate the codes returned from the database into text strings. It then uses these text strings to determine which database it is actually connecting to.
- If the locale (as returned to IBM SPSS Modeler Server by the system `$LANG` query) is not *en_US*, IBM SPSS Modeler cannot translate the codes it receives from the ODBC driver into text. In other words, an untranslated code, rather than the string *Oracle*, is returned to IBM SPSS Modeler Server at the start of a database connection. This means that IBM SPSS Modeler is unable to optimize streams for Oracle.

To check and reset locale specifications:

1. In a UNIX shell, run:

```
#locale
```

This will return the locale information for the shell. For example:

```
$ locale
LANG=en_US.ISO8859-15
LC_CTYPE="en_US.ISO8859-15"
LC_NUMERIC="en_US.ISO8859-15"
LC_TIME="en_US.ISO8859-15"
LC_COLLATE="en_US.ISO8859-15"
LC_MONETARY="en_US.ISO8859-15"
LC_MESSAGES="en_US.ISO8859-15"
LC_ALL=en_US.ISO8859-15
```

2. Change to your Connect ODBC/locale directory. (Here you will see a single directory, *en_US*.)
3. Create a soft link to this *en_US* directory, specifying the name of the locale setup in the shell. An example is as follows:

```
#ln -s en_US en_US.ISO8859-15
```

For a non-English locale, such as *fr_FR.ISO8859-1*, you should create the soft link as follows:

```
#ln -s en_US fr_FR.ISO8859-1
```

4. Once you have created the link, restart IBM SPSS Modeler Server from this same shell. (IBM SPSS Modeler Server receives its locale information from the shell from which it is started.)

Notes

When optimizing a UNIX machine for SQL pushbacks to Oracle, consider the following tips:

- The full locale must be specified. In the above example, you must create the link in the form `language_territory.code-page`. The existing `en_US` locale directory is not sufficient.
- To fully optimize in-database mining, both `LANG` and `LC_ALL` must be defined in the shell used to start IBM SPSS Modeler Server. `LANG` may be defined in the shell as you would any other environment variable before restarting IBM SPSS Modeler Server. For example, see the following definition:

```
#LANG=en_US.ISO8859-15; export LANG
```

- Each time you start IBM SPSS Modeler Server you will need to check that the shell locale information is fully specified and that the appropriate soft link exists in the `ODBC/locale` directory.

Appendix B. Configuring UNIX Startup Scripts

Introduction

This appendix describes some of the scripts that ship with the UNIX versions of IBM SPSS Modeler Server, and it explains how to configure the scripts. Scripts are used to:

- Configure IBM SPSS Modeler Server to start automatically when the server computer is restarted.
- Manually stop and restart IBM SPSS Modeler Server.
- Change permissions on files created by IBM SPSS Modeler Server.
- Configure IBM SPSS Modeler Server to work with the ODBC Connect drivers provided with IBM SPSS Modeler Server. See the topic “[IBM SPSS Modeler Server and the data access pack](#)” on page 74 for more information.

Scripts

IBM SPSS Modeler Server uses several scripts, including:

- **modelersrv.sh.** The manual startup script for IBM SPSS Modeler Server is located in the IBM SPSS Modeler Server installation directory. It configures the environment for the server when the server daemon process is *manually* started. Run it when you want to manually start and stop the server. Edit it when you need to change the configuration for manual startup.
- **auto.sh.** This is a script that configures your system to start the server daemon process automatically at boot time. Run it once to configure your system for automatic startup. You do not need to edit it. The script is located in the IBM SPSS Modeler Server installation directory.
- **rc.modeler.** When you run `auto.sh`, this script is created in a location that depends on your server's operating system. It configures the environment for the server when it is *automatically* started. Edit it when you need to change the configuration for automatic startup.

Automatically Starting and Stopping IBM SPSS Modeler Server

IBM SPSS Modeler Server must be started as a daemon process. The installation program includes a script (`auto.sh`) that you can run to configure your system to automatically stop and restart IBM SPSS Modeler Server.

To configure the system for automatic startup and shutdown

1. Log on as root.
2. Change to the IBM SPSS Modeler Server installation directory.
3. Run the script. At the UNIX prompt, type:

```
./auto.sh
```

An automatic startup script, `rc.modeler`, is created in the location shown in the table above. The operating system will use `rc.modeler` to start the IBM SPSS Modeler Server daemon process whenever the server computer is rebooted. The operating system will also use `rc.modeler` to stop the daemon whenever the system is shut down.

Manually Starting and Stopping IBM SPSS Modeler Server

You can manually start and stop IBM SPSS Modeler Server by running the `modelersrv.sh` script.

To manually start and stop IBM SPSS Modeler Server

1. Change to the IBM SPSS Modeler Server installation directory.
2. To start the server, at the UNIX command prompt, type:

```
./modelersrv.sh start
```

3. To stop the server, at the UNIX command prompt type:

```
./modelersrv.sh stop
```

Editing Scripts

If you use both manual and automatic startup, make parallel changes in both *modelersrv.sh* and *rc.modeler*. If you use only manual startup, make changes in *modelersrv.sh*. If you use only automatic startup, make changes in *rc.modeler*.

To edit the scripts

1. Stop IBM SPSS Modeler Server. (See the topic “[Manually Starting and Stopping IBM SPSS Modeler Server](#)” on page 73 for more information.)
2. Locate the appropriate script. (See the topic “[Scripts](#)” on page 73 for more information.)
3. Open the script in a text editor, make changes, and save the file.
4. Start IBM SPSS Modeler Server, either automatically (by restarting the server computer) or manually.

Controlling Permissions on File Creation

IBM SPSS Modeler Server creates temporary files with read, write, and execute permissions for everyone. You can override this default by editing the UMASK setting in the startup script, either in *modelersrv.sh*, *rc.modeler*, or in both. (For more information, see “[Editing Scripts](#)” on page 74 above.) We recommend 077 as the most restrictive UMASK setting to use. Settings that are more restrictive could cause permissions problems for IBM SPSS Modeler Server.

IBM SPSS Modeler Server and the data access pack

If you want to use the ODBC drivers with IBM SPSS Modeler Server, the ODBC environment must be configured by *odbc.sh* when the IBM SPSS Modeler Server process starts. You do this by editing the appropriate IBM SPSS Modeler start up script, either in *modelersrv.sh*, *rc.modeler*, or in both. (See “[Editing Scripts](#)” on page 74 for more information.)

For more information, see the Technical Support web site at <http://www.ibm.com/support>. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

To configure ODBC to start with IBM SPSS Modeler Server

1. Stop the IBM SPSS Modeler Server host if it is running.
2. Download the relevant compressed TAR archive for the platform on which you have IBM SPSS Modeler Server installed. Make sure that you download the correct drivers for your installed version of IBM SPSS Modeler Server. Copy the file to the location where you want to install the ODBC drivers (for example, */usr/spss/odbc*).
3. Extract the TAR archive file using **tar -xvof**.
4. Run the *setodbcpath.sh* script that is extracted from the archive.
5. Edit the script *odbc.sh* to add the definition of ODBCINI to the bottom of this script and export it, for example:

```
ODBCINI=/usr/spss/odbc/odbc.ini; export ODBCINI
```

ODBCINI must point to the full pathname of the `odbc.ini` file that you want IBM SPSS Modeler to read to get a list of the ODBC datasources that you define (a default `odbc.ini` is installed with the drivers).

6. Save `odbc.sh`.
7. (64-bit IBM SPSS Modeler Server installations only; for other installations, continue from the next step) Define and export `LD_LIBRARY_PATH_64` in `odbc.sh`:

```
if [ "$LD_LIBRARY_PATH_64" = "" ]; then
    LD_LIBRARY_PATH_64=<library_path>
else
    LD_LIBRARY_PATH_64=<library_path>:$LD_LIBRARY_PATH_64
fi
export LD_LIBRARY_PATH_64
```

where `library_path` is the same as for the `LD_LIBRARY_PATH` definition already in the script that has been initialized with your installation path (for example, `/usr/spss/odbc/lib`). The easiest way to do this is to copy the `if` and `export` statements for `LD_LIBRARY_PATH` in your `odbc.sh` file, append them to the end of the file, and then replace the "`LD_LIBRARY_PATH`" strings in the newly appended `if` and `export` statements with "`LD_LIBRARY_PATH_64`".

As an example, your final `odbc.sh` file on a 64-bit IBM SPSS Modeler Server installation might look like this:

```
if [ "$LD_LIBRARY_PATH" = "" ]; then
    LD_LIBRARY_PATH=/usr/spss/odbc/lib
else
    LD_LIBRARY_PATH=/usr/spss/odbc/lib:$LD_LIBRARY_PATH
fi
export LD_LIBRARY_PATH
if [ "$LD_LIBRARY_PATH_64" = "" ]; then
    LD_LIBRARY_PATH_64=/usr/spss/odbc/lib
else
    LD_LIBRARY_PATH_64=/usr/spss/odbc/lib:$LD_LIBRARY_PATH_64
fi
export LD_LIBRARY_PATH_64
ODBCINI=/usr/spss/odbc/odbc.ini; export ODBCINI
```

Remember to export `LD_LIBRARY_PATH_64`, as well as defining it with the `if` loop.

8. Edit the `odbc.ini` file that you defined earlier using `$ODBCINI`. Define the data source names that you require (these depend on the database that you are accessing).
9. Save the `odbc.ini` file.
10. Configure IBM SPSS Modeler Server to use these drivers. To do so, edit `modelersrv.sh` and add the following line immediately below the line that defines `SCLEMDNAME`:

```
. <odbc.sh_path>
```

where `odbc.sh_path` is the full path to the `odbc.sh` file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here; be sure to leave a space between the first period and the path to the file.

11. Save `modelersrv.sh`.

Important: For the SDAP driver to work on Db2 on z/OS, you must grant access to `SYSIBM.SYSPACKSTMT`.

To test the connection

1. Restart IBM SPSS Modeler Server.
2. Connect to IBM SPSS Modeler Server from a client.
3. On the client, add a Database source node to the canvas.
4. Open the node and verify that you can see the data source names that you defined in the `odbc.ini` file earlier in the configuration procedure.

If you do not see what you expect here, or you get errors when you try to connect to a data source that you have defined, follow the Troubleshooting procedure. See the topic [“Troubleshooting ODBC configuration”](#) on page 77 for more information.

To configure ODBC to start with IBM SPSS Modeler Solution Publisher Runtime

When you can successfully connect to the database from IBM SPSS Modeler Server, you can configure an IBM SPSS Modeler Solution Publisher Runtime installation on the same server by referencing the same `odbc.sh` script from the startup script of IBM SPSS Modeler Solution Publisher Runtime.

1. Edit the `modelerrun` script in IBM SPSS Modeler Solution Publisher Runtime to add the following line immediately above the last line of the script:

```
. <odbc.sh_path>
```

where `odbc.sh_path` is the full path to the `odbc.sh` file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here. Be sure to leave a space between the first period and the path to the file.

2. Save the `modelerrun` script file.
3. By default, the DataDirect Driver Manager is not configured for IBM SPSS Modeler Solution Publisher Runtime to use ODBC on UNIX systems. To configure UNIX to load the DataDirect Driver Manager, enter the following commands (where `sp_install_dir` is the installation directory of Solution Publisher Runtime):

```
cd sp_install_dir
rm -f libspssodbc.so
ln -s libspssodbc_datadirect.so libspssodbc.so
```

To configure ODBC to start with IBM SPSS Modeler Batch

No configuration of the IBM SPSS Modeler Batch script is necessary for ODBC. This is because you connect to IBM SPSS Modeler Server from IBM SPSS Modeler Batch in order to run streams. Ensure that the IBM SPSS Modeler Server ODBC configuration has been made and is working correctly, as described earlier in this section.

To add or edit a data source name

1. Edit the `odbc.ini` file to include the new or changed name.
2. Test the connection as described earlier in this section.

When the connection with IBM SPSS Modeler Server is working correctly, the new or changed data source should also work correctly with IBM SPSS Modeler Solution Publisher Runtime and IBM SPSS Modeler Batch.

SQL Server support with the Data Access Pack driver

The ODBC configuration for SQL Server must have the `Enable Quoted Identifiers` ODBC connection attribute set to `Yes` (the default for this driver is `No`). On UNIX this attribute is configured in the system information file (`odbc.ini`) using the `QuotedId` option.

Troubleshooting ODBC configuration

No data sources listed, or random text displayed

If you open a Database source node and the list of available data sources is empty or contains unexpected entries, it may be due to a problem with the startup script.

1. Check that `$ODBCINI` is defined within `modelersrv.sh`, either explicitly in the script itself, or in the `odbc.sh` script that is referenced in `modelersrv.sh`.
2. In the latter case, ensure that `ODBCINI` points to the full path to the `odbc.ini` file that you have used to define your ODBC data sources.
3. If the path specification in `ODBCINI` is correct, check the value of `$ODBCINI` that is being used in the IBM SPSS Modeler Server environment by echoing the variable from within `modelersrv.sh`. To do so, add the following line to `modelersrv.sh` after the point where you define `ODBCINI`:

```
echo $ODBCINI
```

4. Save and then execute `modelersrv.sh`. The value of `$ODBCINI` that is being set in the IBM SPSS Modeler Server environment is written to `stdout` for verification.
5. If no value at all is returned to `stdout`, and you are defining `$ODBCINI` in the `odbc.sh` script that you are referencing from `modelersrv.sh`, check that the referencing syntax is correct. It should be:

```
. <odbc.sh_path>
```

where `odbc.sh_path` is the full path to the `odbc.sh` file that you edited near the beginning of this procedure, for example:

```
. /usr/spss/odbc/odbc.sh
```

Note: The syntax is important here; be sure to leave a space between the first period and the path to the file.

When the correct value is echoed to `stdout` on running `modelersrv.sh`, you should be able to see the data source names in the Database source node when you restart IBM SPSS Modeler Server and connect to it from the client.

IBM SPSS Modeler client hangs on clicking Connect in Database Connections dialog box

This behavior can be caused by your library path not being correctly set to include the path to the ODBC libraries. The library path is defined by `$LD_LIBRARY_PATH` (and `$LD_LIBRARY_PATH_64` on 64-bit versions).

To see the value of the library path in the IBM SPSS Modeler Server daemon environment, echo the value of the appropriate environment variable from within `modelersrv.sh`, after the line where you are appending the ODBC library path to the library path, and execute the script. The library path value will be echoed to the terminal when you next execute the script.

If you are referencing `odbc.sh` from `modelersrv.sh` to set up your IBM SPSS Modeler Server ODBC environment, echo your library path value from the line after the one where you reference the `odbc.sh` script. To echo the value, add the following line to the script, then save and execute the script file:

```
echo $<library_path_variable>
```

where `<library_path_variable>` is the appropriate library path variable for your server operating system.

The returned value of your library path must include the path to the *lib* subdirectory of your ODBC installation. If it does not, append this location to the file.

If you are running the 64-bit version of IBM SPSS Modeler Server, `$LD_LIBRARY_PATH_64` will override `$LD_LIBRARY_PATH` if it is set. If you are having this problem on one of these 64-bit platforms, echo `LD_LIBRARY_PATH_64` as well as `$LD_LIBRARY_PATH` from *modelersrv.sh* and, if required, set `$LD_LIBRARY_PATH_64` so that it includes the *lib* subdirectory of your ODBC installation, and export the definition.

Data source name not found and no default driver specified

If you see this error on clicking Connect in the Database Connections dialog box, it usually indicates that your *odbc.ini* file is incorrectly defined. Check that the data source name (DSN) as defined within the [ODBC Data Sources] section at the top of the file matches the string specified between the square brackets further down in *odbc.ini* to define the DSN. If these are different in any way, you will see this error when you try to connect using the DSN from within IBM SPSS Modeler. Following is an example of an *incorrect* specification:

```
[ODBC Data Sources]
Oracle=Oracle Wire Protocol

...
...
[Oracle Driver]
Driver=/usr/ODBC/lib/XEora22.so
Description=SPSS 5.2 Oracle Wire Protocol
AlternateServers=
...

```

You need to change one of the two strings in bold so that they match exactly. Doing so should resolve the error.

Specified driver could not be loaded

This error also indicates that the *odbc.ini* file is incorrectly defined. One possibility is that the Driver parameter within the driver stanza is incorrectly set, for example:

```
[ODBC Data Sources]
Oracle=Oracle Wire Protocol

...
...
[Oracle]
Driver=/nosuchpath/ODBC/lib/XEora22.so
Description=SPSS 5.2 Oracle Wire Protocol
AlternateServers=

```

1. Check that the shared object specified by the Driver parameter exists.
2. Correct the path to the shared object if it is incorrect.
3. If the Driver parameter is specified in this format:

```
Driver=ODBCHOME/lib/XEora22.so
```

this indicates that you have not initialized your ODBC-related scripts. Run the *setodbcpath.sh* script that is installed with the drivers. See the topic [“IBM SPSS Modeler Server and the data access pack” on page 74](#) for more information. When you have run this script, you should see that the string "ODBCHOME" has been substituted with the path to your ODBC installation. This should resolve the issue.

Another cause may be a problem with the driver's library. Use the `ivtestlib` tool provided with ODBC to confirm that the driver can't be loaded. For Connect64, use the `ddtestlib` tool. Correct the problem by setting the library path variable in the startup script.

For example, if the Oracle driver cannot be loaded for a 32-bit installation, follow these steps:

1. Use `ivtestlib` to confirm that the driver cannot be loaded. For example, at the UNIX prompt, type:

```
sh
cd ODBC DIR
. odbc.sh
./bin/ivtestlib MFor815
```

where `ODBC DIR` is replaced by the path to your ODBC installation directory.

2. Read the message to see if there is an error. For example, the message: `Load of MFor815.so failed: ld.so.1: bin/ivtestlib: fatal: libclntsh.so: open failed: No such file or directory` indicates that the Oracle client library, `libclntsh.so`, is missing or that it is not on the library path.
3. Confirm that the library exists. If it doesn't, reinstall the Oracle client. If the library is there, type the following sequence of commands from the UNIX command prompt:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/bigdisk/oracle/product/8.1.6/lib
export LD_LIBRARY_PATH
./bin/ivtestlib MFor815
```

where `/bigdisk/oracle/product/8.1.6/lib` is replaced by the path to `libclntsh.so` and `LD_LIBRARY_PATH` is the library path variable for your operating system.

Note that if you are running IBM SPSS Modeler 64-bit on Linux, the library path variable contains the suffix `_64`. Therefore, the first two lines in the previous example would become:

```
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:/bigdisk/oracle/product/8.1.6/lib
export LD_LIBRARY_PATH_64
```

4. Read the message to confirm that the driver can now be loaded. For example, the message: `Load of MFor815.so successful, qehandle is 0xFF3A1BE4` indicates that the Oracle client library can be loaded.
5. Correct the library path in the IBM SPSS Modeler startup script.
6. Restart the IBM SPSS Modeler Server with the startup script that you edited (`modelersrv.sh` or `rc.modeler`).

Library paths

The name of the library path variable on Linux 64-bit operating system is `LD_LIBRARY_PATH_64`. Use this value to make appropriate substitutions when you're configuring or troubleshooting on your system.

Appendix C. Configuring and Running SPSS Modeler Server as a Non-Root Process on UNIX

Introduction

These instructions provide information on running IBM SPSS Modeler Server as a non-root process on UNIX systems.

Running as root. The default installation of IBM SPSS Modeler Server assumes that the server daemon process will run as root. Running as root allows IBM SPSS Modeler to authenticate reliably each user login and start each user session on the corresponding UNIX user account. This ensures that users have access only to their own files and directories.

Running as non-root. Running IBM SPSS Modeler Server as a non-root process means having the real and effective user IDs of the server daemon process set to an account of your choice. All user sessions started by SPSS Modeler Server will use the same UNIX account and this means that any file data read or written by SPSS Modeler is shared by all SPSS Modeler users. Access to database data is not affected because users have to authenticate themselves independently to each of the database data sources they use. Without root privilege, IBM SPSS Modeler operates in one of two ways:

- **Without a private password database.** With this method, SPSS Modeler uses the existing UNIX password database, NIS, or LDAP server that is normally used for user authentication on the UNIX system. See the topic [“Configuring as non-root without a private password database”](#) on page 81 for more information.
- **With a private password database.** With this method, SPSS Modeler authenticates users against a private password database, distinct from the UNIX password database, NIS, or LDAP server that is normally used for authentication on UNIX. See the topic [“Configuring as non-root using a private password database”](#) on page 82 for more information.

Note: On Linux/UNIX systems where both a non-root and SSL configuration are enabled, SSL security will be reduced. Because all user sessions run under the same credential as each other and as the Modeler Server daemon, the SSL certificate data that should to be kept secret will instead be exposed to all users. This allows users to easily bypass the normal protections SSL provides to all other users. See [“Securing client/server and server-server communications with SSL”](#) on page 42.

Configuring as non-root without a private password database

To configure IBM SPSS Modeler Server to run on a non-root account without the need for a private password database, follow these steps:

1. Open the SPSS Modeler Server *options.cfg* file for editing.
2. Set the option `start_process_as_login_user` to Y.
3. Save and close the *options.cfg* file.

By default, SPSS Modeler Server tries each authentication method until it finds one that works. However, if desired you can use the `authentication_methods` option in *options.cfg* to configure the server to try only one specific authentication method. Possible values for the option are `pasw_modeler`, `gss`, `pam`, `sspi`, `unix`, or `windows`.

Note that running as non-root is likely to require configuration updates. See the topic [“Troubleshooting user authentication failures”](#) on page 83 for more information.



CAUTION: Do not enable the `start_process_as_login_user` setting and then start the IBM SPSS Modeler Server as root. Doing so would mean that, for all users that are connected to the server, their server processes would run as *root*; this is a security risk. Note that the server may stop automatically if you attempt this.

Configuring as non-root using a private password database

If you choose to authenticate users by means of a private password database, all user sessions are started on the same non-root user account.

To configure IBM SPSS Modeler Server to run on a non-root account in this way, follow these steps:

1. Create a group to contain all of your users. You can name this group whatever you'd like, but for this example, let's call it *modelerusers*.
2. Create the user account on which to run IBM SPSS Modeler Server. This account is for the sole use of the IBM SPSS Modeler Server daemon process. For this example, let's call it *modelerserv*.

When creating the account, note that:

- The primary group should be the *<modelerusers>* group created previously.
 - The home directory can be the IBM SPSS Modeler installation directory or any other convenient default (consider using something other than the installation directory if you need the account to survive upgrades).
3. Next, configure the startup scripts to start IBM SPSS Modeler Server using the newly created account. Locate the appropriate startup script and open it in a text editor. See the topic [“Scripts” on page 73](#) for more information.
 - a. Change the umask setting to allow at least group read access on created files:

```
umask 027
```

4. Edit the server options file, *config/options.cfg*, to specify authentication against the private password database by appending the line:

```
authentication_methods, "pasw_modeler"
```

5. Edit the server options file, *config/options.cfg*, to set the option `start_process_as_login_user` to Y.
6. Next, you'll need to create a private password database stored in the file *config/passwords.cfg*. The password file defines the user name/password combinations that are allowed to login to IBM SPSS Modeler. *Note:* These are private to IBM SPSS Modeler and have no connection with the user names and passwords used to login to UNIX. You can use the same user names for convenience, but you cannot use the same passwords.

To create the password file, you will need to use the password utility program, *pwutil*, located in the *bin* directory of the IBM SPSS Modeler Server installation. The synopsis of this program is:

```
pwutil [ username [ password ] ]
```

The program takes a user name and plain-text password and writes the user name and encrypted password to the standard output in a format suitable for inclusion in the password file. For example, to define a user *modeler* with the password “data mining” you would type:

```
bin/pwutil modeler "data mining" > config/passwords.cfg
```

Defining a single user name is sufficient in most cases, where all users log in with the same name and password. However additional users can be created by using the `>>` operator to append each to the file, for example:

```
bin/pwutil modeler "data miner2" >> config/passwords.cfg
```

Note: If a single `>` is used, the contents of *passwords.cfg* will be overwritten each time, replacing any users set previously. Remember that all users share the same UNIX user account regardless.

Note: If you add new users to the private passwords database while SPSS Modeler Server is running, you will need to restart SPSS Modeler Server so that it can recognize the newly defined users. Until you do so, logins will fail for any new users added via `pwutil` since the last restart of SPSS Modeler Server.

7. Recursively change the ownership of the IBM SPSS Modeler installation directory and its contents to be user `<modelerserv>` and group `<modelerservusers>` where the names referenced are those you created earlier. For example:

```
chown -R -h modelerserv:modelerservusers .
```

8. Consider creating subdirectories in the data directory for your IBM SPSS Modeler users so that they have somewhere to store working data without interference. These directories should be group-owned by the `<modelerservusers>` group and have group read, write, and search permissions. For example, to create a working directory for user `bob`:

```
mkdir data/bob
chown bob:modelerservusers data/bob
chmod ug=rwx,o= data/bob
```

Additionally, you can set the set-group-ID bit on the directory so that any data files copied into the directory will be automatically group-owned by `<modelerservusers>`:

```
chmod g+s data/bob
```

Running SPSS Modeler Server as a non-root user

To run SPSS Modeler Server as a non-root user, follow these steps:

1. Log in using the non-root user account created earlier.
2. If you are running with the configuration file option `start_process_as_login_user` enabled, you can start, stop, and check the status of SPSS Modeler Server. See the topic [“To Start, Stop, and Check Status on UNIX”](#) on page 21 for more information.

End users connect to SPSS Modeler Server by logging in from the client software. You must give end users the information that they need to connect, including the IP address or host name of the server machine.

Troubleshooting user authentication failures

Depending on how the operating system is configured to perform authentication, you may experience failures to log on to SPSS Modeler Server when running in a non-root configuration. For example, this may occur if your operating system is configured (using the `/etc/nsswitch.conf` file or similar) to check the local shadow password file, rather than use NIS or LDAP. This occurs because SPSS Modeler Server requires read access to the files used to perform authentication, including the `/etc/shadow` file or its equivalent, which stores secure user account information. However, the operating system file permissions are generally set so that `/etc/shadow` is accessible only by the root user. Under these circumstances a non-root process cannot read `/etc/shadow` to validate user passwords, resulting in an authentication error.

There are several ways to resolve this issue:

- Ask your system administrator to configure the operating system to use NIS or LDAP for authentication.
- Change the file permissions on the protected files, for example by granting read access to the `/etc/shadow` file so that the local user account used to run SPSS Modeler Server can access the file. While this workaround might be deemed unsuitable in production environments, it could be temporarily applied to a test environment to verify whether the authorization failure is linked to the operating system configuration.
- Specify an access control list (ACL) for the `/etc/shadow` file.
- Run SPSS Modeler Server as root, to enable the server processes to read the `/etc/shadow` file.



CAUTION: In this case, ensure that the `options.cfg` file for SPSS Modeler Server contains the option `start_process_as_login_user, N` to avoid the security issue explained earlier.

Appendix D. Configuring and Running SPSS Modeler Server with a private password file on Windows

Introduction

These instructions provide information on running IBM SPSS Modeler Server using a private password file on Windows systems. With this method, IBM SPSS Modeler authenticates users against a private password database, distinct from the system authentication on Windows.

Configuring a private password database

If you choose to authenticate users by using a private password database, all user sessions are started on the same user account.

To configure SPSS Modeler Server in this way, follow these steps:

1. Create the user account on which to run SPSS Modeler Server. This account is for the sole use of the SPSS Modeler Server daemon process. You must start the daemon process as that user account in the Log On tab of the SPSS Modeler Server 18.5.0 Service. For this example, let's call it *modelerserv*.
2. Edit the server options file (`config/options.cfg`) to set the option `start_process_as_login_user` to Y
3. Edit the server options file (`config/options.cfg`) to specify authentication against the private password database by appending the line:

```
authentication_methods, "pasw_modeler"
```

4. Next, you need to create a private password database that is stored in the file `config/passwords.cfg`. The password file defines the user name/password combinations that are allowed to log in to SPSS Modeler. Note that these combinations are private to SPSS Modeler and have no connection with the user names and passwords that are used to log in to Windows. You can use the same user names for convenience, but you cannot use the same passwords.

To create the password file, you need to use the password utility program, `pwutil`, in the `bin` directory of the SPSS Modeler Server installation. The synopsis of this program is:

```
pwutil [ username [ password ] ]
```

The program takes a user name and plain-text password and writes the user name and encrypted password to the standard output in a format suitable for inclusion in the password file. For example, to define a user that is called `modeler` with the password data `mining`, you would use a DOS prompt to navigate to the SPSS Modeler Server installation directory and then type:

```
bin\pwutil modeler "data mining" > config\passwords.cfg
```

Note: Make sure that you have only 1 instance of each user in the file; duplicates prevent SPSS Modeler Server from starting

Defining a single user name is sufficient in most cases, where all users log in with the same name and password. However, more users can be created by using the `>>` operator to append each to the file. For example:

```
bin\pwutil modeler "data miner2" >> config\passwords.cfg
```

Note:

If a single > is used, the contents of `passwords.cfg` are overwritten each time, replacing any users set previously. Remember that all users share the same UNIX user account regardless.

If you add new users to the private passwords database while SPSS Modeler Server is running, you will need to restart SPSS Modeler Server so that it can recognize the newly defined users. Until you do so, logins will fail for any new users added via `pwutil` since the last restart of SPSS Modeler Server.

5. Give the user that was created in step 1 full control over the server options file `config\options.cfg` and the `%ALLUSERSPROFILE%\IBM\SPSS` directory.
6. In the system services, stop the **IBM SPSS Modeler Server** service and change the **Log on** from the **Local System Account** to the user account created in step 1. Then, restart the service.

Appendix E. Load Balancing with Server Clusters

With IBM SPSS Collaboration and Deployment Services, a plug-in called the Coordinator of Processes can be used to manage services on the network. The Coordinator of Processes provides server management capabilities designed to optimize client-server communication and processing.

Services to be managed, such as IBM SPSS Statistics Server or IBM SPSS Modeler Server, register with the Coordinator of Processes upon starting and periodically send updated status messages. Services can also store any necessary configuration files in the IBM SPSS Collaboration and Deployment Services Repository and retrieve them when initializing.

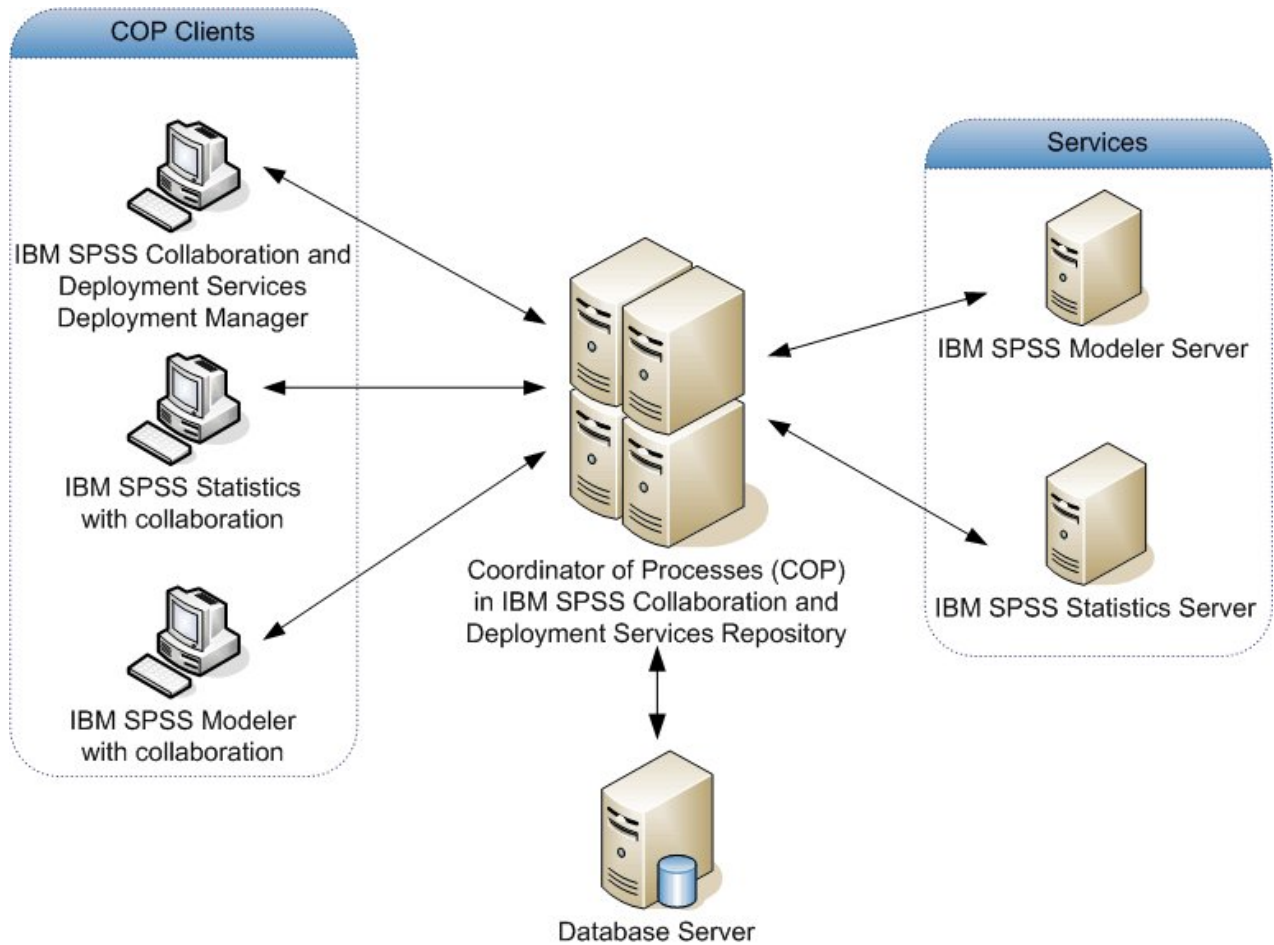


Figure 5. Coordinator of Processes Architecture

Executing your IBM SPSS Modeler streams on a server can increase performance. In some cases, you may have only the choice of one or two servers. In other cases, you might be offered a larger choice of servers because there is a substantive difference between each server, such as owner, access rights, server data, test versus production servers, and so on. In addition, if you have the Coordinator of Processes on your network, you might be offered a server cluster.

A server cluster is a group of servers that are interchangeable in terms of configuration and resources. The Coordinator of Processes determines which server is best suited to respond to a processing request, using an algorithm that will balance the load according to several criteria, including the server weights, user priorities, and current processing loads. For more information, see the *Coordinator of Processes Service Developer's Guide* available in the IBM SPSS Collaboration and Deployment Services documentation suite.

Whenever you connect to a server or server cluster in IBM SPSS Modeler, you can enter a server manually or search for a server or cluster using the Coordinator of Processes. See the topic [“Connecting to IBM SPSS Modeler Server ” on page 11](#) for more information.

Appendix F. LDAP authentication

These instructions provide basic guidelines on how to configure SPSS Modeler Server on UNIX to use LDAP authentication, where the identities of the users who will log in to the server are stored in an LDAP directory.

Note: As a prerequisite, the LDAP client software must be correctly configured on the host operating system. For more information, see the original vendor documentation.

Usually no additional configuration is required and the use of LDAP is not apparent to the server. Examples of where no additional changes are required include the following circumstances:

- The LDAP client and server software are configured according to RFC 2307.
- Access to the passwd (and, where applicable, shadow) database is redirected to LDAP, for example in `nsswitch.conf`.
- Each valid user of SPSS Modeler Server has a passwd (and shadow) entry that is stored in the LDAP directory.
- The SPSS Modeler Server service is started by using the root account.

There are two sets of circumstances when it might become necessary to configure SPSS Modeler Server specifically for LDAP:

- When the service is started by using an account other than root, the service might not have the authority to authenticate by using the default method. Typically this is because access to the shadow database is restricted.
- When users do not have passwd (or shadow) entries that are stored in the directory; that is, they do not have user identities that are valid for login to the host system.

The LDAP authentication procedure uses the PAM subsystem and requires that a PAM LDAP module exists and is correctly configured for the host operating system. For more information, see the original vendor documentation.

Complete the following steps to configure SPSS Modeler Server to use LDAP authentication exclusively.

Note: These steps provide the most basic configuration that can be expected to work. More options or alternative settings might be required depending on your operating system and local security policy. For more information, see the original operating documentation.

1. Edit the service configuration file (`options.cfg`) and add (or edit) the line:
 `authentication_methods, pam`. This line instructs the server to use PAM authentication in preference to the default authentication.
2. Provide a PAM configuration for the SPSS Modeler Server service; which often requires root privileges. The service is identified by the name `modelerserver`.
3. On a Linux type system, which uses `/etc/pam.d`, create a file in that directory with the name `modelerserver` and add content similar to the following example:

```
# IBM SPSS Modeler Server
auth required pam_ldap.so
account required pam_ldap.so
password required pam_deny.so
session required pam_deny.so
```

4. The names of the referenced PAM modules vary by operating system; confirm the modules that are required for your host operating system.

Note: The lines in steps 3 specify that SPSS Modeler Server must refer to the PAM LDAP module for authentication and account management. However, changing of passwords and session management

are not supported so these actions are not allowed. If account management is not required or is inappropriate, change the relevant line to permit all requests, as in the following example:

```
# IBM SPSS Modeler Server
auth required pam_ldap.so
account required pam_permit.so
password required pam_deny.so
session required pam_deny.so
```

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

Special Characters

@SQLFN function [69](#)

Numerics

64-bit operating systems [6](#)

A

adding IBM SPSS Modeler Server connections [17](#), [18](#)

administration

of IBM SPSS Modeler Server [31](#)

administrator access

for IBM SPSS Modeler Server [32](#)

with User Access Control (UAC) [32](#)

allow_modelling_memory_override

options.cfg file [35](#)

application examples [3](#)

architecture

components [5](#)

authentication [18](#)

auto.sh (UNIX)

location of [73](#)

automatic server startup

configuring on UNIX [73](#)

C

cache compression [35](#)

cache_compression

options.cfg file [35](#)

cache_connection option [41](#)

caching, in-database [40](#)

chemsrv.sh (UNIX)

location of [73](#)

CLEM expressions

SQL generation [66](#)

client

single sign-on [15](#)

Cognos SSL connection [46](#)

Cognos TM1 SSL connection [46](#)

configuration options

automatic SQL generation [36](#)

connections and sessions [33](#)

coordinator of processes [37](#)

COP [37](#)

data file access [34](#)

login attempts [33](#)

memory management [35](#)

of IBM SPSS Modeler Server [31](#)

overview [33](#)

parallel processing [35](#)

performance and optimization [35](#)

port number [33](#)

SQL string length [36](#)

configuration options (*continued*)

SSL data encryption [37](#)

stream rewriting [35](#)

temp directory [34](#)

connections

server cluster [18](#)

to IBM SPSS Modeler Server [11](#), [17](#), [18](#)

Coordinator of Processes

load balancing [87](#)

server clusters [87](#)

coordinator of processes configuration

for IBM SPSS Modeler Server [37](#)

COP

load balancing [87](#)

server clusters [87](#)

COP configuration

for IBM SPSS Modeler Server [37](#)

cop_enabled

options.cfg file [37](#)

cop_host

options.cfg file [37](#)

cop_password

options.cfg file [37](#)

cop_port_number

options.cfg file [37](#)

cop_service_description

options.cfg file [37](#)

cop_service_host

options.cfg file [37](#)

cop_service_name

options.cfg file [37](#)

cop_service_weight

options.cfg file [37](#)

cop_update_interval

options.cfg file [37](#)

cop_user_name

options.cfg file [37](#)

D

data access [8](#)

data access pack

and UNIX library paths [79](#)

configuring UNIX for [74](#)

ODBC, configuring on UNIX [74](#)

troubleshooting ODBC on UNIX [77](#)

data files

IBM SPSS Statistics [10](#)

importing and exporting [10](#)

data sources

single sign-on [17](#)

data_file_path

options.cfg file [34](#)

data_files_restricted

options.cfg file [34](#)

database caching

controlling from options.cfg [40](#)

- database caching (*continued*)
 - SQL generation [61](#)
- database connections
 - closing [41](#)
- database servers [54](#)
- databases
 - accessing [8](#)
- Db2
 - SQL optimization [57](#), [58](#)
- disk space
 - calculating [8](#)
- documentation [3](#)
- domain name (Windows)
 - IBM SPSS Modeler Server [11](#)

E

- encryption
 - FIPS [38](#)
 - SSL [41](#)
- error on stream execution [35](#)
- examples
 - Applications Guide [3](#)
 - overview [4](#)

F

- file permissions
 - configuring on UNIX [74](#)
 - on IBM SPSS Modeler Server [19](#)
- filenames
 - UNIX [10](#)
 - Windows [10](#)
- FIPS encryption [38](#)
- firewall settings
 - options.cfg file [35](#)

G

- group_configuration [38](#)

H

- hard drives [8](#)
- hardware recommendations
 - for IBM SPSS Modeler Server [6](#)
- host name
 - IBM SPSS Modeler Server [11](#), [17](#)

I

- IBM SPSS Analytic Server
 - configuration options [33](#)
- IBM SPSS Modeler
 - documentation [3](#)
- IBM SPSS Modeler Administration Console
 - administrator access [32](#)
 - User Access Control access [32](#)
- IBM SPSS Modeler client
 - single sign-on [15](#)
- IBM SPSS Modeler Server
 - administration of [31](#)
 - administration options [31](#)

- IBM SPSS Modeler Server (*continued*)
 - administrator access [32](#)
 - configuration options [33](#)
 - coordinator of processes configuration [37](#)
 - COP configuration [37](#)
 - different results than client [20](#)
 - domain name (Windows) [11](#)
 - file creation [19](#)
 - host name [11](#), [17](#)
 - information for end users [18](#)
 - monitoring usage [40](#)
 - password [11](#)
 - permissions [19](#)
 - port number [11](#), [17](#), [33](#)
 - server processes [40](#)
 - single sign-on [12](#), [14](#)
 - single sign-on for data sources [17](#)
 - temp directory [34](#)
 - unresponsive processes [22](#)
 - User Access Control access [32](#)
 - user accounts [18](#)
 - user authentication [18](#)
 - user ID [11](#)
- IBM SPSS Statistics data access technology [8](#)
- IBM SPSS Statistics data files
 - importing and exporting [10](#)
- in-database caching [40](#)
- in-database mining [54](#)
- io_buffer_size
 - options.cfg file [35](#)

K

- Kerberos [38](#)
- kernel limits on UNIX [21](#)

L

- LDAP
 - authentication [89](#)
- Linux
 - single sign-on [14](#)
- log files
 - displaying generated SQL [60](#)
 - for IBM SPSS Modeler Server [50](#)
- logging in to IBM SPSS Modeler Server [11](#)

M

- max_file_size
 - options.cfg file [34](#)
- max_login_attempts
 - options.cfg file [33](#)
- max_parallelism
 - options.cfg file [35](#)
- max_sessions
 - options.cfg file [33](#)
- max_sql_string_length
 - options.cfg file [36](#)
- memory [8](#)
- memory management
 - administration options [35](#)
- memory_usage

- memory_usage (*continued*)
 - options.cfg file [35](#)
- messages
 - displaying generated SQL [60](#)
- Microsoft SQL Server
 - SQL optimization [57](#), [58](#)
- model nuggets
 - viewing SQL for [60](#)
- modeling
 - memory management [35](#)
- modelling_memory_limit_percentage
 - options.cfg file [35](#)
- multiple stream execution [35](#)

N

- node caching
 - SQL generation [61](#)
 - writing to database [61](#)
- nodes
 - that support SQL generation [62](#), [70](#)

O

- ODBC
 - configuring on UNIX [74](#)
- ODBC data sources
 - and UNIX [74](#)
 - ODBC and UNIX scripts [74](#)
- ODBC driver setup [57](#)
- operating systems
 - 64-bit [6](#)
- operators
 - SQL generation [66](#)
- optimization
 - SQL generation [57](#), [58](#), [60](#)
- options.cfg [38](#)
- options.cfg file [40](#)
- Oracle
 - SQL optimization [57](#), [58](#), [71](#)

P

- PAM
 - authentication [89](#)
- parallel processing
 - controlling [35](#)
- password
 - IBM SPSS Modeler Server [11](#)
- paths [10](#)
- performance
 - of IBM SPSS Modeler Server [53](#)
- permissions [19](#)
- port number
 - IBM SPSS Modeler Server [11](#), [17](#), [33](#)
- port settings
 - options.cfg file [35](#)
- port_number
 - options.cfg file [33](#)
- preview
 - SQL generation [60](#)
- processes, unresponsive [22](#)
- processors

- processors (*continued*)
 - multiple [35](#)
- program_file_path
 - options.cfg file [34](#)
- program_files_restricted
 - options.cfg file [34](#)
- purple nodes
 - SQL optimization [58](#)
- pushbacks
 - CLEM expressions [66](#)

R

- RAM [8](#)
- rc.modeler (UNIX)
 - location of [73](#)
- restarting the web service [31](#)
- results
 - differences between Client and Server [20](#)
 - record order [20](#)
 - rounding of [20](#)

S

- searching COP for connections [18](#)
- Secure Sockets Layer [41](#)
- security
 - configuring file creation on UNIX [74](#)
 - file creation [19](#)
 - SSL [41](#)
- server
 - adding connections [17](#)
 - logging in [11](#)
 - searching COP for servers [18](#)
 - single sign-on [12](#), [14](#)
- server port settings
 - options.cfg file [35](#)
- server_logging.log [50](#)
- single sign-on [11](#)
- SQL
 - duplicate column names [69](#)
 - optimizing Oracle [71](#)
 - previewing generated [60](#)
 - queries [69](#)
 - viewing for model nuggets [60](#)
- SQL generation
 - CLEM expressions [61](#), [66](#)
 - enabling [60](#)
 - enabling for IBM SPSS Modeler Server [36](#)
 - logging [60](#)
 - previewing [60](#)
 - stream rewriting [61](#)
 - tips [61](#)
 - viewing for model nuggets [60](#)
- SQL pushback. See also SQL generation [57](#)
- SQL Server
 - SQL optimization [57](#), [58](#)
- sql_generation_enabled
 - options.cfg file [36](#)
- SSL
 - Cognos connection [46](#)
 - Cognos TM1 connection [46](#)
 - overview [41](#)

- SSL (*continued*)
 - securing communications [42](#)
- SSL data encryption
 - enabling for IBM SPSS Modeler Server [37](#)
- ssl_certificate_file
 - options.cfg file [37](#)
- ssl_enabled
 - options.cfg file [37](#)
- ssl_private_key_file
 - options.cfg file [37](#)
- ssl_private_key_password
 - options.cfg file [37](#)
- starting IBM SPSS Modeler Server
 - on UNIX [21](#)
 - on Windows [21](#)
- status
 - of IBM SPSS Modeler Server on UNIX [21](#)
 - of IBM SPSS Modeler Server on Windows [21](#)
- stopping IBM SPSS Modeler Server
 - on UNIX [21](#)
 - on Windows [21](#)
- stream rewriting [61](#)
- stream_rewriting_enabled
 - options.cfg file [35](#)

Z

- zombie processes, IBM SPSS Modeler Server [22](#)

T

- temp directory
 - for IBM SPSS Modeler Server [34](#)
- temp_directory
 - options.cfg file [34](#)
- temporary files
 - permissions for (IBM SPSS Modeler Server) [19](#)

U

- UNC filenames [10](#)
- UNIX
 - configuring file permissions [74](#)
 - library paths [79](#)
 - permissions [19](#)
 - restarting the web service [31](#)
 - single sign-on [14](#)
 - user authentication [18](#)
- UNIX kernel limits [21](#)
- UNIX scripts
 - auto.sh [73](#)
 - editing [74](#)
 - modelersrv.sh [73](#)
 - rc.modeler [73](#)
- UNIX shell [38](#)
- user accounts
 - IBM SPSS Modeler Server [18](#)
 - permissions [19](#)
- user authentication [18](#)
- user ID
 - IBM SPSS Modeler Server [11](#)

W

- web service - restarting [31](#)
- Windows
 - restarting the web service [31](#)

