

**IBM SPSS Modeler Server 14.2
Administration and Performance Guide**



Note: Before using this information and the product it supports, read the general information under Notices on p. .

This edition applies to IBM SPSS Modeler 14 and to all subsequent releases and modifications until otherwise indicated in new editions.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Licensed Materials - Property of IBM

© Copyright IBM Corporation 1994, 2011.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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 <http://www.ibm.com/spss>.

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. web site at <http://www.ibm.com/support>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Contents

1	About IBM SPSS Modeler	1
	IBM SPSS Modeler Server	1
	IBM SPSS Modeler Options	1
	IBM SPSS Text Analytics	2
	IBM SPSS Modeler Documentation	2
	Application Examples	3
	Demos Folder	4
2	Architecture and Hardware Recommendations	5
	IBM SPSS Modeler Architecture	5
	Architecture Description	5
	Hardware Recommendations	7
	Temporary Disk Space and RAM Requirements	8
	Data Access	9
	Referencing Data Files.	11
	Importing IBM SPSS Statistics Data Files.	11
	Installation Instructions	11
3	IBM SPSS Modeler Support	12
	Connecting to IBM SPSS Modeler Server	12
	Adding and Editing the IBM SPSS Modeler Server Connection	14
	Searching for Servers in IBM SPSS Collaboration and Deployment Services	15
	Data and File Systems.	16
	User Authentication	16
	File Creation	17
	Differences in Results.	17
4	IBM SPSS Modeler Server Administration	18
	Starting and Stopping IBM SPSS Modeler Server	18
	To Start, Stop, and Check Status on Windows	18
	To Start, Stop, and Check Status on UNIX	18
	Handling Unresponsive Server Processes (UNIX Systems)	19

Administration	20
Using IBM SPSS Modeler Administration Console	20
Starting Modeler Administration Console	21
Configuring Access with Modeler Administration Console	21
SPSS Modeler Server Connections	22
SPSS Modeler Server Configuration	23
SPSS Modeler Server Monitoring	29
Using the options.cfg File	30
Closing Unused Database Connections	31
IBM SPSS Statistics License Location	31
Using SSL to secure data transfer	32
How SSL works	32
Securing client-server and server-server communications with SSL	32
URL prefix configuration	36
Securing LDAP with SSL	36
Server Log	37

5 Performance Overview 39

Server Performance and Optimization Settings	39
Client Performance and Optimization Settings	39
Database Usage and Optimization	42
SQL Optimization	42

6 SQL Optimization 44

How SQL Generation Works	45
SQL Generation Example	46
Configuring SQL Optimization	47
Previewing Generated SQL	48
Viewing SQL for Model Nuggets	49
Tips for Maximizing SQL Generation	50
Nodes Supporting SQL Generation	51
CLEM Expressions and Operators Supporting SQL Generation	54
Using SQL Functions in CLEM Expressions	56
Writing SQL Queries	57

Appendices

A Configuring Oracle for UNIX Platforms **58**

Configuring Oracle for SQL Optimization 58

B Configuring UNIX Startup Scripts **60**

Introduction 60

Scripts 60

Automatically Starting and Stopping IBM SPSS Modeler Server 61

Manually Starting and Stopping IBM SPSS Modeler Server 61

Editing Scripts 61

Controlling Permissions on File Creation 62

IBM SPSS Modeler Server and the Data Access Pack 62

 Troubleshooting Connect for ODBC Configuration 65

 Library Paths 68

Configuring a Driver for the Enterprise View Node 68

C Running As a Non-Root Process on UNIX **69**

Introduction 69

Configuring IBM SPSS Modeler Server As a Non-Root Process 69

D Load Balancing with Server Clusters **72**

E Notices **74**

Index **77**

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, SPSS Modeler supports the entire data mining process, from data to better business results.

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 in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see <http://www.ibm.com/software/analytics/spss/products/modeler/>.

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. Additional products or updates beyond those listed here may also be available. For more information, see <http://www.ibm.com/software/analytics/spss/products/modeler/>.

SPSS Modeler. SPSS Modeler is a functionally complete version of the product that is installed and run on the user's desktop computer. It can be run in local mode as a standalone product or in distributed mode along with IBM® SPSS® Modeler Server for improved performance on large data sets.

SPSS Modeler Server. SPSS Modeler Server runs continually in distributed analysis mode together with one or more IBM® SPSS® Modeler installations, providing superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation. At least one SPSS Modeler installation must be present to run an analysis.

IBM SPSS Modeler Options

The following components and features can be separately purchased and licensed for use with SPSS Modeler. Note that additional products or updates may also become available. For more information, see <http://www.ibm.com/software/analytics/spss/products/modeler/>.

- SPSS Modeler Server access, providing improved scalability and performance on large data sets, as well as support for SQL optimization and in-database modeling capabilities.

- SPSS Modeler Solution Publisher, for real-time or automated scoring outside the SPSS Modeler environment. [For more information, see the topic IBM SPSS Modeler Solution Publisher in Chapter 2 in *IBM SPSS Modeler 14.2 Solution Publisher*.](#)
- Adapters to enable deployment to IBM SPSS Collaboration and Deployment Services or the thin-client application IBM SPSS Modeler Advantage. [For more information, see the topic Storing and Deploying IBM SPSS Collaboration and Deployment Services Repository Objects in Chapter 9 in *IBM SPSS Modeler 14.2 User's Guide*.](#)

IBM SPSS Text Analytics

IBM® SPSS® Text Analytics is a fully integrated add-on for SPSS Modeler that 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.

- The Text Mining node offers concept and category modeling, as well as an interactive workbench where you can perform advanced exploration of text links and clusters, create your own categories, and refine the linguistic resource templates.
- A number of import formats are supported, including blogs and other web-based sources.
- Custom templates, libraries, and dictionaries for specific domains, such as CRM and genomics, are also included.

Note: A separate license is required to access this component. For more information, see <http://www.ibm.com/software/analytics/spss/products/modeler/>.

IBM SPSS Modeler Documentation

Complete documentation in online help format is available from the Help menu of SPSS Modeler. This includes documentation for SPSS Modeler, SPSS Modeler Server, and SPSS Modeler Solution Publisher, as well as the Applications Guide and other supporting materials.

Complete documentation for each product in PDF format is available under the *Documentation* folder on each product DVD.

- **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, Predictive Applications, 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. [For more information, see the topic Overview of Modeling Nodes in Chapter 3 in *IBM SPSS Modeler 14.2 Modeling Nodes*.](#)

- **IBM SPSS Modeler Algorithms Guide.** Descriptions of the mathematical foundations of the modeling methods used in SPSS Modeler.
- **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. [For more information, see the topic Application Examples in IBM SPSS Modeler 14.2 User's Guide.](#)
- **IBM SPSS Modeler Scripting and Automation.** Information on automating the system through scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running SPSS Modeler streams and scenarios as steps in processing jobs under IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **IBM SPSS Modeler CLEF Developer's Guide.** CLEF provides the ability to integrate third-party programs such as data processing routines or modeling algorithms as nodes in SPSS Modeler.
- **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 and Performance Guide.** Information on how to configure and administer IBM® SPSS® Modeler Server.
- **IBM SPSS Modeler Administration Console User Guide.** Information on installing and using the console user interface for monitoring and configuring SPSS Modeler Server. The console is implemented as a plug-in to the Deployment Manager application.
- **IBM SPSS Modeler Solution Publisher Guide.** SPSS Modeler Solution Publisher is an add-on component that enables organizations to publish streams for use outside of the standard SPSS Modeler environment.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.

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 involved should be scalable to real-world applications.

You can access the examples by clicking 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 the topic Demos Folder in IBM SPSS Modeler 14.2 User's Guide.](#)

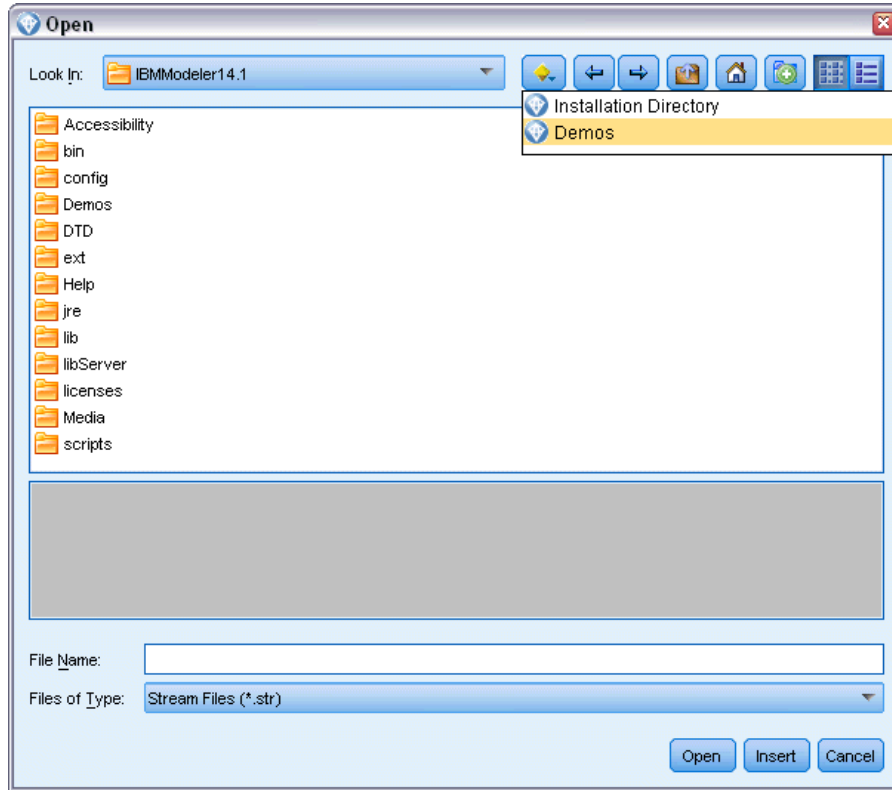
Database modeling examples. See the examples in the *IBM SPSS Modeler In-Database Mining Guide*.

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

Demos Folder

The data files and sample streams used with the application examples are installed in the *Demos* folder under the product installation directory. This folder can also be accessed from the IBM SPSS Modeler 14.2 program group on the Windows Start menu, or by clicking *Demos* on the list of recent directories in the File Open dialog box.

Figure 1-1
Selecting the Demos folder from the list of recently-used directories



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 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 SPSS Modeler Server (versus the standalone IBM® SPSS® Modeler), especially when dealing with large data sets, are numerous:

- 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.
- SPSS Modeler Server is optimized for fast performance. When operations cannot be pushed into the database, SPSS Modeler Server stores the intermediate results as temporary files on disk rather than in RAM. Because servers usually have significant disk space available, 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® Collaboration and Deployment Services Deployment Manager (included with SPSS Modeler Server) and the IBM® SPSS® Collaboration and Deployment Services and IBM® SPSS® Collaboration and Deployment Services Repository (sold separately), you can monitor data mining processes, ensuring that adequate computing resources are available. You can automate certain data mining tasks, manage access to data models, and share results across your organization.

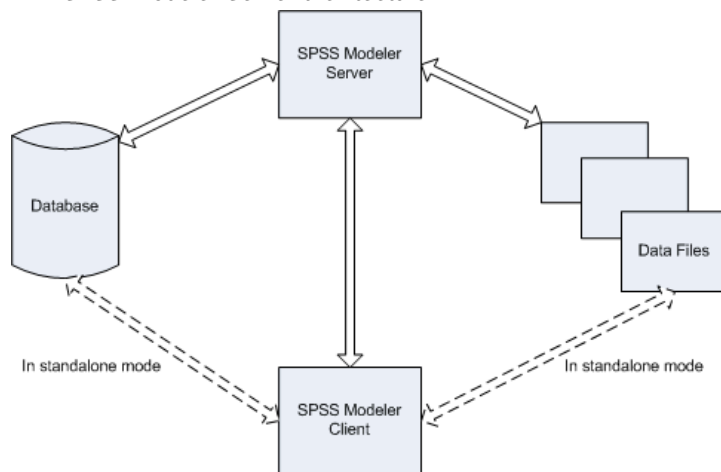
The components of IBM® SPSS® Modeler's distributed architecture are shown in the "IBM SPSS Modeler Server architecture" graphic on p. 6.

- **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 SPSS Modeler software, but when it is connected to SPSS Modeler Server for distributed analysis, its execution engine is inactive. The SPSS Modeler runs on Windows operating systems only.

- **SPSS Modeler Server.** The server software installed on a server computer, with network connectivity to both the SPSS Modeler(s) and the database. 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 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 2-1
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, 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.

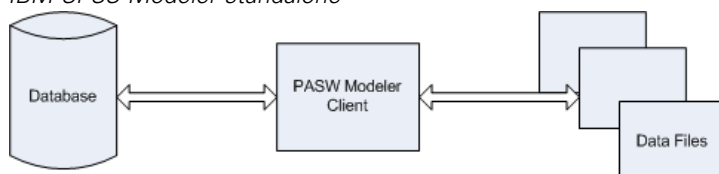
If necessary, SPSS Modeler Server can execute all SPSS Modeler operations outside of the database. It automatically balances its use of RAM and disk memory to hold data for manipulation. This process makes SPSS Modeler Server fully compatible with flat files.

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. [For more information, see the topic Load Balancing with Server Clusters in Appendix D on p. 72.](#) You can connect to a server or cluster managed in the Coordinator of Processes directly through SPSS Modeler's Server Login dialog. [For more information, see the topic Connecting to IBM SPSS Modeler Server in Chapter 3 in *IBM SPSS Modeler 14.2 User's Guide*.](#)

Standalone Client

SPSS Modeler may also be configured to run as a self-contained desktop application, shown in the “IBM SPSS Modeler standalone” graphic below. [For more information, see the topic IBM SPSS Modeler Support in Chapter 3 on p. 12.](#)

Figure 2-2
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 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 SPSS Modeler Server on a dedicated server machine where it will not compete for resources with other applications, including any databases to which 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 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 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).

64-bit platforms. If you plan to process or build models on very large volumes of data, use 64-bit Solaris, Windows, or Linux as your 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. 64-bit support is available for Solaris, Windows, and Linux platforms.

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 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

IBM® SPSS® Modeler Server's powerful SQL optimization feature allows processing to 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, 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 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 corporate Web site at <http://www.spss.com/drivers/clientCLEM.htm>. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

Database support in IBM® SPSS® Modeler is classified into three tiers, each representing a different level of support for SQL pushback and optimization, depending on the database vendor. The different levels of support are implemented by means of a number of system settings, which can be customized as part of a Services engagement.

The three tiers of database support are:

Table 2-1
Database support tiers

Support tier	Description
Tier 1	All possible SQL pushback is available, with database-specific SQL optimization.
Tier 2	Most SQL pushback is available, with non-database-specific SQL optimization.
Tier 3	No SQL pushback or optimization—only reading data from, and writing data to, the database.

Supported ODBC Drivers

For the latest information on which databases and ODBC drivers are supported and tested for use with SPSS Modeler 14.2, see the product compatibility matrices on the corporate Support site (<http://www.ibm.com/support>).

Where to Install Drivers

Note that ODBC drivers must be installed and configured on each computer where processing may 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 SPSS Modeler in distributed mode against a remote IBM® SPSS® Modeler Server, the ODBC drivers need to be installed on the computer where SPSS Modeler Server is installed.
- If you need to access the same data sources from both SPSS Modeler and SPSS Modeler Server, the ODBC drivers must be installed on both computers.
- If you are running SPSS Modeler over Terminal Services, the ODBC drivers need to be installed on the Terminal Services server on which you have SPSS Modeler installed.
- If you are using the IBM® SPSS® Modeler Solution Publisher Runtime to run published streams on a separate computer, you also need to install and configure ODBC drivers on that computer.

Note: If you are using SPSS Modeler Server on UNIX to access a Teradata database you must use the ODBC Driver Manager that is installed with the Teradata ODBC driver. In order to make this change to SPSS Modeler Server please specify a value for ODBC_DRIVER_MANAGER_PATH near the top of the modelersrv.sh script where indicated by the comments. This environment variable needs to be set to the location of the ODBC Driver Manager that is shipped with the Teradata ODBC driver (/usr/odbc/lib in a Teradata ODBC driver default installation). You must restart SPSS Modeler Server for the change to take effect. For details of the SPSS Modeler Server platforms that offer support for Teradata access, and the Teradata ODBC driver version that is supported, see the corporate Support site at <http://www.ibm.com/support>.

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. For information about how to configure SPSS Modeler Server on UNIX to work with the IBM® SPSS® Statistics data access technology, see [Configuring UNIX Startup Scripts on p. 60](#).

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 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 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 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. IBM® SPSS® Modeler's SPSS Statistics nodes can automatically start the SPSS Statistics client, but users must first ensure that the SPSS Statistics client is operating in the same mode as SPSS Modeler.

Installation Instructions

For information on installing IBM® SPSS® Modeler Server, see the instructions in the *\documentation\installation* folder on the Server CD. Separate documents are available for Windows and UNIX.

For complete information on installing and using the IBM® SPSS® Modeler, see the client CD.

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:

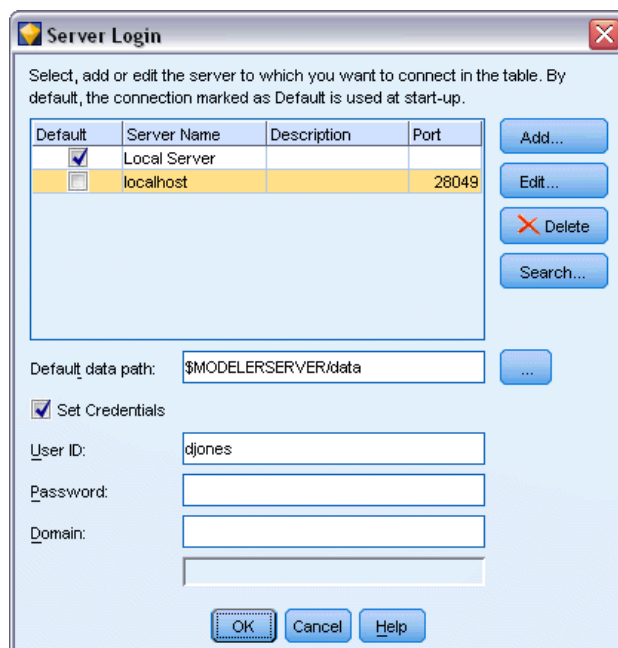
- How to log on to IBM® SPSS® Modeler Server (or run standalone by disconnecting from a Server)
- Data and file systems that users may need
- User accounts and file permissions pertaining to SPSS Modeler Server
- Differences in results that users may see when switching between SPSS Modeler Server and 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 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 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. [For more information, see the topic Load Balancing with Server Clusters in Appendix D on p. 72.](#)

Figure 3-1
Server Login dialog box



To Connect to a Server

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the SPSS Modeler window.
- ▶ 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. [For more information, see the topic Adding and Editing the IBM SPSS Modeler Server Connection in IBM SPSS Modeler 14.2 User's Guide.](#)
 - Click Search to access a server or server cluster in the Coordinator of Processes. [For more information, see the topic Searching for Servers in IBM SPSS Collaboration and Deployment Services in IBM SPSS Modeler 14.2 User's Guide.](#)

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.

- ▶ Click OK to complete the connection.

To Disconnect from a Server

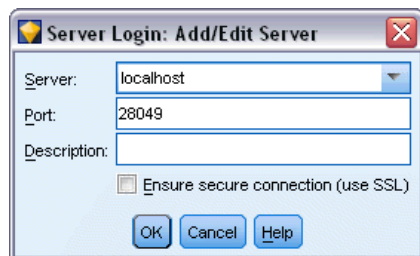
- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the SPSS Modeler window.
- ▶ In the dialog box, select the Local Server and click OK.

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.

Figure 3-2
Server Login Add/Edit Server dialog box



To Add Server Connections

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ▶ In this dialog box, click Add. The Server Login Add/Edit Server dialog box opens.
- ▶ 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

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ▶ In this dialog box, select the connection you want to edit and then click Edit. The Server Login Add/Edit Server dialog box opens.
- ▶ 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. [For more information, see the topic Load Balancing with Server Clusters in Appendix D on p. 72.](#)

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 logon 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. [For more information, see the topic Adding and Editing the IBM SPSS Modeler Server Connection in IBM SPSS Modeler 14.2 User's Guide.](#)

Figure 3-3
Search for Servers dialog box



To search for servers and clusters

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ▶ 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. [For more information, see the topic *Connecting to the IBM SPSS Collaboration and Deployment Services Repository* in Chapter 9 in *IBM SPSS Modeler 14.2 User's Guide*.](#)
- ▶ Select the server or server cluster from the list.
- ▶ 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 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 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 may 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 IBM® SPSS® Modeler users must share a single UNIX user account (and, consequently, share access to data files). [For more information, see the topic *Configuring IBM SPSS Modeler Server As a Non-Root Process* in Appendix C on p. 69.](#)

On Solaris, HP-UX, Linux and AIX platforms, SPSS Modeler Server uses PAM for authentication. You can use the service name `modelerserver` to configure the PAM modules for SPSS Modeler Server if required.

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. [For more information, see the topic Temporary Disk Space and RAM Requirements in Chapter 2 on p. 8.](#)

UNIX. The UNIX versions of SPSS Modeler Server use the UNIX `umask` command to set file permissions for the temporary files. You can override the server's default permissions. [For more information, see the topic Controlling Permissions on File Creation in Appendix B on p. 62.](#)

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 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 SPSS Modeler Server. Due to rounding differences, results might vary slightly between each version.

IBM SPSS Modeler Server Administration

This chapter contains information about starting and stopping IBM® SPSS® Modeler Server, configuring various server options, and interpreting the log file. It describes how to use the IBM® SPSS® Collaboration and Deployment Services Deployment Manager, an application that facilitates server configuration and monitoring. For installation instructions for this component, please refer to the installation instructions for SPSS Modeler Server, available on your product CD.

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 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.

- ▶ **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.

- ▶ 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 SPSS Modeler Server installation directory.

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

```
cd /usr/modelersrv
```


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

- ▶ To start the server, at the command prompt, type

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

- ▶ To stop the server, at the command prompt, type

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

- ▶ To check the status of 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 SPSS Modeler Server daemon process, and remaining processes are IBM® SPSS® Modeler sessions.

The 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. [For more information, see the topic Automatically Starting and Stopping IBM SPSS Modeler Server in Appendix B on p. 61.](#)

UNIX kernel limits

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

- ▶ Add the following commands to *modelersrv.sh*:

```
ulimit -d unlimited
```

```
ulimit -m unlimited
```

```
ulimit -f unlimited
```

- ▶ Restart 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 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 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 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 SPSS Modeler Server at regular intervals, all remaining files are removed from the SPSS Modeler temporary directory.

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

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® Collaboration and Deployment Services Deployment Manager application, which is available free of charge to current IBM® SPSS® Modeler customers. [For more information, see the topic Using IBM SPSS Modeler Administration Console on p. 20.](#)
- Use the `options.cfg` text file, located in the `[server install path]/config` directory. [For more information, see the topic Using the options.cfg File on p. 30.](#)

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

More information about using Deployment Manager and the `options.cfg` file is provided in the following sections.

Using IBM SPSS Modeler Administration Console

The Modeler Administration Console application 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.

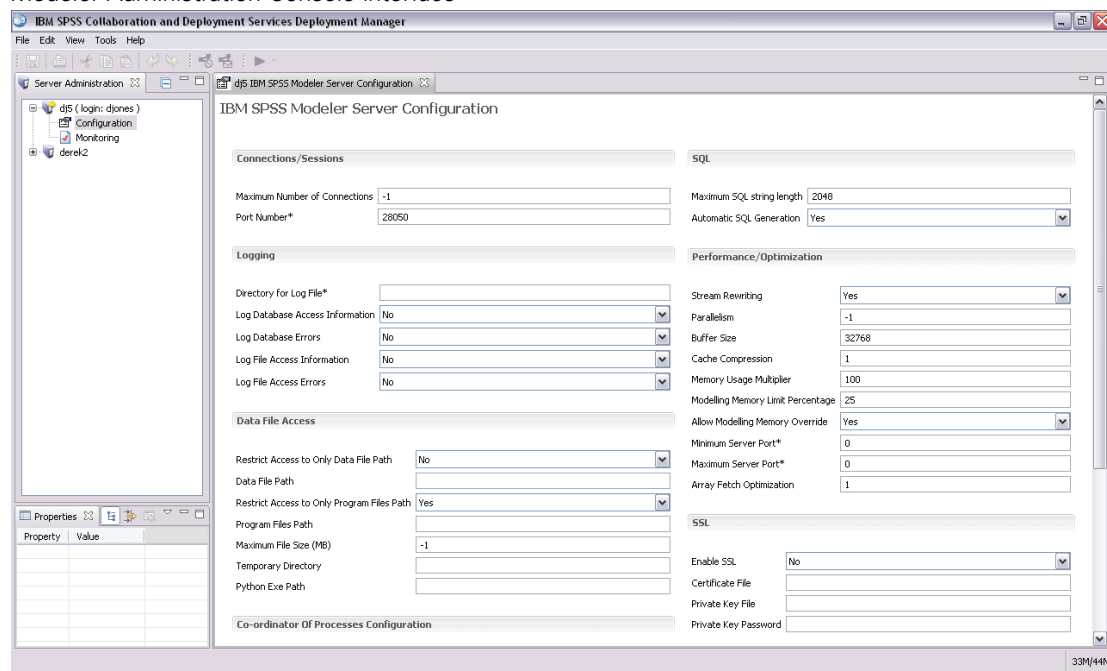
Many of the options available through Modeler Administration Console can also be specified in the *options.cfg* file, which is located in the SPSS Modeler Server installation directory under */config*. However, the Modeler Administration Console application provides a shared graphical interface that 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, then IBM SPSS Modeler Administration Console <nn.n>.

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.

Figure 4-1
Modeler Administration Console interface



Configuring Access with Modeler Administration Console

Administrator access to SPSS Modeler Server through Modeler Administration Console 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.

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, the login credentials are not. You must log in every time you start Modeler Administration Console.

To Set Up a Server Connection

- ▶ Ensure that the IBM® SPSS® Modeler Server service is started.
- ▶ From the File menu, choose New and then Administered Server Connection.
- ▶ 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.
- ▶ 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

- ▶ In the Server Administrator pane, double-click to select the server to which you want to log in.
- ▶ 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. [For more information, see the topic Configuring Access with Modeler Administration Console on p. 21.](#)

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 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 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 the port number requires a server restart in order 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.

Connections/Sessions

Figure 4-2
Connections/Sessions settings

Connections/Sessions	
Maximum Number of Connections	-1
Port Number*	28047

Maximum number of connections. (max_sessions) Maximum number of users connected at one time. A value of -1 indicates no limit.

Port number. (port_number) The port number for SPSS Modeler Server to listen on. Change if another application already uses the default (28047). End users must know the port number in order to use SPSS Modeler Server.

Logging

Figure 4-3
Logging settings

Logging	
Directory for Log File*	
Log Database Access Information	No
Log Database Errors	No
Log File Access Information	No
Log File Access Errors	No

Directory for log file. (log_directory) Location of the *messages.log* file. A blank string means *[server install path]/log*. Note that you must use forward slashes. For information about the contents of this log file, see [Server Log on p. 37](#).

Log database access information. (log_database_access) Add database access information to the log file.

Log database errors. (log_database_errors) Add database errors to the log file.

Log file access information. (log_file_access) Add file access information to the log file.

Log file access errors. (log_file_errors) Add file access errors to the log file.

Data File Access

Figure 4-4
Data File Access settings

Data File Access	
Restrict Access to Only Data File Path	No <input type="button" value="v"/>
Data File Path	<input type="text"/>
Restrict Access to Only Program Files Path	Yes <input type="button" value="v"/>
Program Files Path	<input type="text"/>
Maximum File Size (MB)	-1
Temporary Directory	<input type="text"/>
Python Exe Path	<input type="text"/>

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 those listed in the Data File Path below.

Data file path. (data_file_path) A list of additional 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. Note that you should use forward slashes in all pathnames. Specify multiple directories using semicolons (for example, *[server install path]/data;c:/data;c:/temp*).

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 those listed in the Program Files Path below.

Program files path. (program_file_path) A list of additional 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. Note that you should use forward slashes in all pathnames. Specify multiple directories using semicolons.

Maximum file size (MB). (max_file_size) Maximum size of temporary and exported data files 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 should be on a separate high-speed drive or controller because speed of access to this directory may have a significant impact on performance. You may specify

multiple temporary directories, separating each with a comma. These should be located on different disks; the first directory is used most often, and additional 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 will fail.

Python Exe Path. (`python_exe_path`) Full path to the python executable including the executable name. Note that Restrict access to only program files path may need to be set to *No* depending on the location where python is installed.

Performance/Optimization

Figure 4-5
Performance/Optimization settings

Performance/Optimization	
Stream Rewriting	Yes
Parallelism	-1
Buffer Size	32768
Cache Compression	1
Memory Usage Multiplier	100
Modelling Memory Limit Percentage	25
Allow Modelling Memory Override	Yes
Minimum Server Port	0
Maximum Server Port	0
Array Fetch Optimization	1

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.

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, and improves performance by reducing the amount of disk activity needed to create and read the cache. 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 are rarely needed except 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 neural net, 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, 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, SPSS Modeler will need two open ports (in addition to the main server port) available per concurrent stream, plus an additional port for each connected client that has an ODBC connection. To execute more concurrent streams, you can widen the range of port numbers and open up more ports on the firewall accordingly.

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

Figure 4-6
SQL settings

SQL

Maximum SQL string length

Automatic SQL Generation

Maximum SQL string length. (`max_sql_string_length`) Maximum length of a string imported from the database with SQL. String values longer than this are truncated on the right without warning. The valid range is between 1 and 65,535 characters.

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.

SSL

Figure 4-7
SSL settings

SSL

Enable SSL

Certificate File

Private Key File

Private Key Password

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

Certificate file. (`ssl_certificate_file`) The public SSL certificate file to be loaded when the server starts.

Private key file. (`ssl_private_key_file`) The private SSL key file to be loaded when the server starts up.

Private key password. (`ssl_private_key_password`) Password key for the specified certificate. Note this key must have been signed by at least one of the Certification Authorities listed in the configured certificate file.

Coordinator of Processes Configuration

Figure 4-8
Coordinator of Processes configuration settings

Co-ordinator Of Processes Configuration	
Host*	spsscop
Port Number*	8080
Login Name*	
Password*	
Enabled*	No
Server Name*	
Description*	
Update Interval (min)*	2
Weight	1
Service Host*	

Items marked (*) require a server restart to take effect.

Host. (`cop_host`) The hostname or IP address of the Coordinator of Processes service. The default “spsscop” 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.

Login name. (`cop_user_name`) 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 encrypted password for authentication to the Coordinator of Processes service.

To create the password file, you will need to use the password utility program, *pwutil*, located in the *bin* directory of the SPSS Modeler Server installation.

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.

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.

Options Visible Only in options.cfg

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.

max_transfer_size. For internal system use only; do not modify.

start_process_as_login_user. Set this to true if using single sign-on and you are connected to a IBM® SPSS® Collaboration and Deployment Services server that does not use the same authentication method as the IBM® SPSS® Modeler Server host. Setting this option to true enables child processes to use the SPSS Modeler Server login details.

administrators. Specify the usernames of those users to whom you want to grant administrator access. [For more information, see the topic Configuring Access with Modeler Administration Console on p. 21.](#)

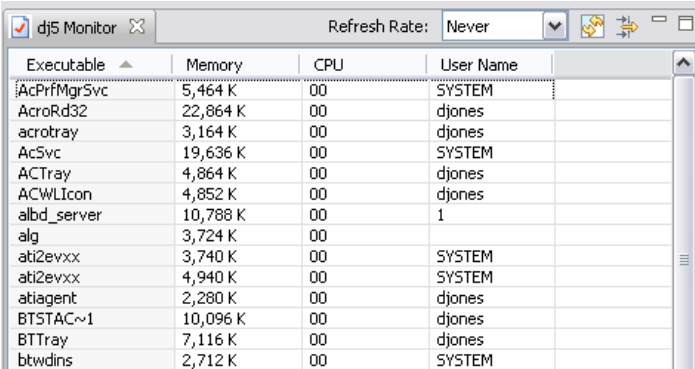
supported_sql_type_logging. For debug purposes only; do not modify unless instructed to do so by a technical-support representative.

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

SPSS Modeler Server Monitoring

The monitoring pane of Modeler Administration Console 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.

Figure 4-9
IBM SPSS Modeler Administration Console monitoring pane



Executable	Memory	CPU	User Name
AcPrfMgrSvc	5,464 K	00	SYSTEM
AcroRd32	22,864 K	00	djones
acrotray	3,164 K	00	djones
AcSvc	19,636 K	00	SYSTEM
ACTray	4,864 K	00	djones
ACWLIcon	4,852 K	00	djones
albd_server	10,788 K	00	1
alg	3,724 K	00	
ati2evxx	3,740 K	00	SYSTEM
ati2evxx	4,940 K	00	SYSTEM
atiagent	2,280 K	00	djones
BTSTAC~1	10,096 K	00	djones
BTTray	7,116 K	00	djones
btwdins	2,712 K	00	SYSTEM

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: With the exception of the following, all configuration options can be changed using IBM® SPSS® Collaboration and Deployment Services Deployment Manager, rather than this configuration file:

- administrators
- allow_custom_config_overrides
- max_transfer_size
- start_process_as_login_user
- supported_sql_type_logging

By using Deployment Manager, you can avoid server restarts for all options except the server port. [For more information, see the topic Using IBM SPSS Modeler Administration Console on p. 20.](#)

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

- ▶ Open the *options.cfg* file with a text editor.
- ▶ Locate the options of interest. For a full list of options, see [SPSS Modeler Server Configuration on p. 23.](#)

- ▶ Edit the values, as appropriate. Note that all pathname values must use a forward slash (/) rather than a backslash as the pathname separator.
- ▶ Save the file.
- ▶ Stop and restart SPSS Modeler Server so that the changes will take effect. [For more information, see the topic Starting and Stopping IBM SPSS Modeler Server on p. 18.](#)

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 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 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 SPSS Modeler option `cache_connection` in a custom database configuration file. Doing so can also make 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 SPSS Modeler session.

To cause unused database connections to be closed:

- ▶ Locate the *[server install path]/config* directory.
- ▶ Add the following file (or open it, if it already exists):
odbc-custom-properties.cfg
- ▶ Add the following line to the file:
`cache_connection, N`
- ▶ Save and close the file.
- ▶ Restart IBM® SPSS® Modeler Server to cause the changes to take effect.

IBM SPSS Statistics License Location

To enable IBM® SPSS® Modeler to use the Statistics Transform, Statistics Model, and Statistics Output nodes, you must have a copy of IBM® SPSS® Statistics installed and licensed on the computer where the stream is executed. When running against a server, the license configuration must be done on that server.

Windows

- ▶ From the command prompt, change to the IBM® SPSS® Modeler Server bin directory and run:
statisticsutility -location =<path to IBM SPSS Statistics Server license file>

Where <path to SPSS Statistics Server license file> is the installation directory of a licensed SPSS Statistics server.

UNIX

- ▶ From the command prompt, change to the SPSS Modeler Server bin directory and run:
./statisticsutility -location =<path to IBM SPSS Statistics Server license file>

Where <path to SPSS Statistics Server license file> is the installation directory of a licensed SPSS Statistics server.

Using SSL to secure data transfer

Security Sockets Layer (SSL) is a protocol for encrypting data transferred between two computers. SSL ensures that communication between the computers is secure. SSL can encrypt the authentication of a username/password and the contents of an exchange between a server and client.

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.

- ▶ When a client connects to a server, the client authenticates the server with the public key certificate.
- ▶ 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.
- ▶ The server decrypts the random number with its private key.
- ▶ 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:

- ▶ Install OpenSSL on the server computer.

- ▶ Obtain and install the SSL certificate and keys.
- ▶ Enable and configure SSL in the server administration application (IBM® SPSS® Collaboration and Deployment Services Deployment Manager).
- ▶ If desired, install unlimited strength encryption on the client computers.
- ▶ If using a self-signed certificate, copy the certificate on the client computer.
- ▶ Add the certificate to the client keystore.
- ▶ Instruct end users to enable SSL when connecting to the server.

Note: 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, SPSS Statistics Server is the *client*.

Install OpenSSL

If OpenSSL is not already available on the server, you must install it.

- ▶ Download OpenSSL from <http://www.openssl.org/>. Be sure to use the version of OpenSSL appropriate for the server version:

Table 4-1
OpenSSL versions

Server Product	Compatible OpenSSL Version
IBM® SPSS® Statistics 17-19 (<i>not</i> Linux® on System z®) IBM® SPSS® Modeler 14.2 (<i>not</i> Linux® on System z®)	0.9.8 and its subversions (0.9.8a, 0.9.8b, and so on)
SPSS Statistics 19 (Linux® on System z®) SPSS Modeler 14.2 (Linux® on System z®)	1.0.0

Note: IBM® SPSS® Modeler Server for UNIX, when configured for SSL, looks to link with two libraries with names such as (the *.so* may be platform-specific):

- *libssl.so.0.9.8*
- *libcrypto.so.0.9.8*

When using a subversion of SSL (0.9.8a, 0.9.8b, and so on), you must create symbolic links for these libraries with the *xxx.0.9.8* versions that SPSS Modeler Server expects.

- ▶ Follow the instructions for installing and configuring the software. If you build OpenSSL yourself, be aware of the following guidelines:
 - Windows.** OpenSSL should be built with DLLs (which are multithreaded by default).
 - UNIX.** OpenSSL should support multiple threads (which is not always by default) and shared libraries.
- ▶ Ensure that the system can locate the SSL libraries:
 - Windows.** Make sure the OpenSSL modules are included on the system path.

UNIX. Append the SSL libraries to the end of the load library path defined in *modelersrv.sh*. For example, on Solaris, add this line to the server startup script:

```
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:/usr/local/openssl; export LD_LIBRARY_PATH_64
```

Obtaining and installing SSL certificate and keys

- ▶ Obtain an SSL certificate and key file. There are two ways you can do this:
 - Purchase them from a public certificate authority (such as VeriSign or Thawte). The public certificate authority signs the certificate to verify the server that uses it.
 - Generate the key and certificate files with an internal self-signed certificate authority. OpenSSL provides a certificate management tool for this purpose, or you can search the Internet for instructions on creating a self-signed SSL certificate.
- ▶ Copy the certificate and key file or files to a local directory on the server. The public and private keys can be stored in one file. Ensure that access permissions deny casual browsing of this directory.
- ▶ Install the SSL certificate and keys on the application server. For additional information on how the keys and certificate interoperate with a specific application server, see the original vendor's documentation. Note that you may be required to add the certificate and keys to the Java keystore.

Enable and Configure SSL in the IBM SPSS Collaboration and Deployment Services Deployment Manager

- ▶ Start the server administration application (IBM® SPSS® Collaboration and Deployment Services Deployment Manager) and connect to the server.
- ▶ On the configuration page, set Secure Sockets Layer to Yes.
- ▶ In SSL Public Key File, specify the full path to the public key file.
- ▶ 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.

- ▶ From the menus choose:
File > Save
- ▶ 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.

Set Up SSL in IBM SPSS Modeler Server (UNIX systems only)

If a password is required to access the private SSL key file, on UNIX systems you can supply the password in one of the following ways:

- Specify it in the *options.cfg* file by using the `ssl_private_key_password` option. [For more information, see the topic SSL on p. 27.](#)
- Add a switch to the *modelersrv.sh* file to prompt for the password on startup.

If using the latter option, locate the following line in the *modelersrv.sh* file:

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

Add the `-request_ssl_password` switch as follows:

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

Installing unlimited strength encryption

The Java Runtime Environment shipped with the product has US export-strength encryption enabled. For enhanced security of your data, we recommend that this is upgraded to unlimited-strength encryption.

- ▶ Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0 from <http://java.sun.com/javase/downloads/index.jsp>.
- ▶ Unzip the downloaded file.
- ▶ Copy the two *.jar* files *local_policy.jar* and *US_export_policy.jar* into *<installation folder>/jre/lib/security*, where *<installation folder>* is the folder in which you installed the product.

Adding the certificate to client keystore (for connections to the repository or IBM SPSS Modeler Server)

Note: Skip this step if you are using a certificate that is signed by a certificate authority.

If you are using SSL to connect to an IBM® SPSS® Collaboration and Deployment Services repository or IBM® SPSS® Modeler Server and you are using self-signed certificates, you need to add the certificate to the client's Java keystore. The following steps are completed on the *client* computer.

Be aware that a server computer may also act as a client. An example is IBM® SPSS® Statistics Server connecting to the IBM® SPSS® Collaboration and Deployment Services Repository. In this case, SPSS Statistics Server is the *client*, and therefore you need to copy the certificate for the IBM SPSS Collaboration and Deployment Services Repository server to the SPSS Statistics Server.

- ▶ Copy the certificate (*.cer*) file from the SSL server to the *<install_dir>/jre/lib/security* directory on the client machine.

- ▶ In the same directory, right-click on the *cacerts* file and select Properties.
- ▶ Clear the Read-only check box.
- ▶ Open a command prompt and change directories to the following location, where *<product install dir>* is the directory in which you installed the product:

```
<product install dir>/jre/bin
```

- ▶ Enter the following command:

```
keytool -import -alias <alias name> -file <path to cert> -keystore <path to key store>
```

Where *<alias name>* is an arbitrary alias for the certificate, *<path to cert>* is the full path to the certificate, and *<path to key store>* is the full path to the Java keystore, which may be *<product install dir>/lib/security/jssecacerts* or *<product install dir>/lib/security/cacerts*.

- ▶ When prompted, enter the keystore password, which is `changeit` by default.
- ▶ When prompted about trusting the certificate, enter `yes`.

Instructing end users to enable SSL

When end users connect to the server through a client product, they need to enable SSL in the dialog box for connecting to the server. Be sure to tell your users to select the appropriate check box. For example in IBM® SPSS® Modeler, users should select the Ensure secure connection check box on the Repository: Server dialog box, accessed from the Repository option on the Tools menu.

URL prefix configuration

If IBM® SPSS® Collaboration and Deployment Services Repository is set up for SSL access, the value of the URL Prefix configuration setting must be modified as follows:

1. Log into the repository using browser-based console.
2. Open *URL Prefix* configuration option.
Configuration > Setup > URL Prefix
3. Set the value of the prefix to `https` instead of `http` and set the port value to the SSL port number.
For example:

```
[default]
http://<hostname>:<port>
[SSL-enabled]
https://<hostname>:<SSLport>
```

Securing LDAP with SSL

Lightweight Directory Access Protocol (LDAP) is an Internet Engineering Task Force (IETF) standard for exchanging information between network directories and databases containing any level of information. For systems requiring additional security, LDAP providers, such as

Microsoft's Active Directory, can operate over Secure Socket Layer (SSL), provided that the Web or application server supports LDAP over SSL. Using SSL in conjunction with LDAP can ensure that login passwords, application information, and other sensitive data are not hijacked, compromised, or stolen.

The following example illustrates how to enable LDAPS using Microsoft's Active Directory as a security provider. For more specific information on any of the steps or to find details that address a particular release of the security provider, see the original vendor documentation.

1. Verify that Active Directory and the Enterprise Certificate Authority are installed and functioning.
2. Use the certificate authority to generate a certificate, and import the certificate into the certificate store of the IBM® SPSS® Collaboration and Deployment Services Deployment Manager installation. This allows the LDAPS connection to be established between the IBM® SPSS® Collaboration and Deployment Services Repository and an Active Directory server.

To configure Deployment Manager for secure Active Directory connections, verify that a connection exists to the repository.

3. Launch the IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
4. From the Tools menu, choose Server Administration.
5. Log in to a previously defined administered server.
6. Double-click the Configuration icon for the server to expand the hierarchy.
7. Double-click the Security Providers icon to expand the hierarchy.
8. Double-click the Active Directory security provider.
9. Enter configuration values for the instance of Active Directory with security certificates installed.
10. Select the Use SSL check box.
11. Note the name in the Domain User field. Subsequent logins using Active Directory are authenticated using SSL.

For additional information about installing, configuring, and implementing LDAPS on a particular application server, see the original vendor's documentation.

Server Log

IBM® SPSS® Modeler Server keeps a record of its important actions in a log file called *messages.log*. By default, this file is located in *[server install path]/log*, but the location is configurable. When SPSS Modeler Server is started, the log file is updated to indicate the time of the start and the port number it is listening on. From that point forward, the log file is updated with the result whenever a client attempts to connect to the server.

The server writes the following events to the log file by default:

- Server started
- Server stopped

- Connection accepted
- Connection refused
- Session started
- Session stopped
- System error
- Login succeeded
- Login failed

If you want to record information about database access events, change Log Database Access Information and Log Database Errors to *yes* in the SPSS Modeler Server configuration. The following database access events will be logged:

- Database login succeeded
- Database login failed
- Database logout

If you want to record information about file access events, change Log File Access Information and Log File Access Errors to *yes* in the SPSS Modeler Server configuration. The following file access events will be logged:

- File open/create
- File close

Here is an example of a log file (line numbers have been added):

1. 2006/06/14 08:41:23 [167]: IBM SPSS Modeler Server Version 0.1.0 started
2. 2006/06/14 08:41:23 [167]: Listening on port 28028
3. 2006/06/14 08:49:28 [167]: Accepted connection from localhost:1827
4. 2006/06/14 08:49:28 [167]: Session 105 (localhost:1827) started
5. 2006/06/14 08:49:29 [167-105]: Login failed for user: SPSS\awilson
6. 2006/06/14 09:27:27 [167]: Session 105 ended
7. 2006/06/14 09:53:39 [167]: Accepted connection from localhost:1837
8. 2006/06/14 09:53:39 [167]: Session 63 (localhost:1837) started
9. 2006/06/14 09:53:40 [167-63]: Login succeeded for user: SPSS\awilson
10. 2006/06/14 10:27:31 [167]: Session 63 ended

All of the items in the log file contain a timestamp. Lines 1 and 2 were added to the log file when SPSS Modeler Server was started. Line 3 was added when a user on the machine *localhost* on port number *1827* attempted to connect to the server. Line 4 indicates that a server session was started with this user. Line 5 shows that the user failed to log in correctly. Some time later, the user ended the session. Half an hour later, another connection was attempted from *localhost* and a session was started. This time, the login was a success. After another half hour, the session ended, and the connection was closed.

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 optimization whenever possible.
- Use hardware that meets or exceeds the recommendations given in [Chapter 2](#).
- Be sure that your server's performance and optimization settings are properly configured. Some of these settings are also available in the client, but they are only applicable when using IBM® SPSS® Modeler in standalone mode. When IBM® SPSS® Modeler is connected to a server, it automatically uses the server's performance and optimization settings.
- Optimize streams for maximum performance. [For more information, see the topic Performance Considerations for Streams and Nodes in Chapter 13 in IBM SPSS Modeler 14.2 User's Guide.](#)

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® Collaboration and Deployment Services Deployment Manager application, which is included with SPSS Modeler Server. [For more information, see the topic Using IBM SPSS Modeler Administration Console in Chapter 4 on p. 20.](#)

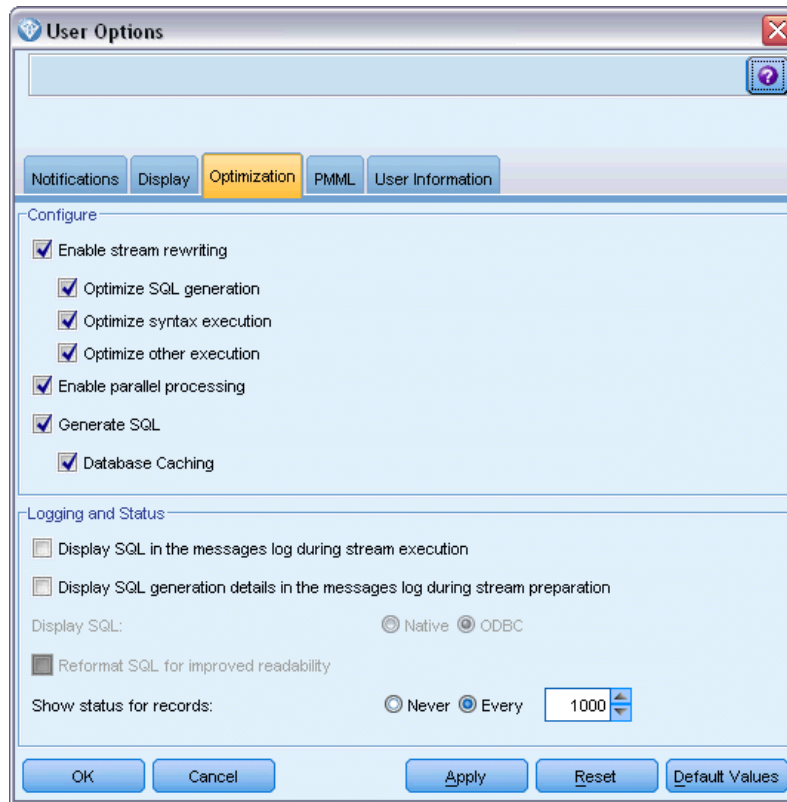
The settings are grouped under the heading Performance and Optimization in the Deployment Manager configuration window. The settings are preconfigured for optimal performance for most installations. However, you may need to adjust them depending on your particular hardware, the size of your data sets, and the contents of your streams. [For more information, see the topic Performance/Optimization in Chapter 4 on p. 25.](#)

Client Performance and Optimization Settings

The Optimization tab on the User Options dialog box allows you to optimize stream performance. Note that the performance and optimization settings on IBM® SPSS® Modeler Server (if used) override any settings in the client. *This tab is available only if SPSS Modeler Server connectivity has been enabled on the IBM® SPSS® Modeler computer.*

Note: Database modeling and SQL optimization require that SPSS Modeler Server connectivity be enabled on the SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler. [For more information, see the topic Connecting to IBM SPSS Modeler Server in Chapter 3 in IBM SPSS Modeler 14.2 User's Guide.](#)

Figure 5-1
User Options dialog box, Optimization tab



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 14.2, see the product compatibility matrices on the corporate Support site at <http://www.ibm.com/support>.

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

- **Optimize SQL generation.** This option allows 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 SPSS Modeler, but such pushbacks act to reduce the size of the data set that is returned to 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. [For more information, see the topic SQL Optimization in Chapter 6 on p. 44.](#)

- **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. [For more information, see the topic Caching Options for Nodes in Chapter 5 in IBM SPSS Modeler 14.2 User's Guide.](#)

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

Display SQL in the messages log during stream execution. Specifies whether SQL generated while running the stream is passed to the message log.

Display SQL generation details in the messages log during stream preparation. During stream preview, specifies whether a preview of the SQL that would be generated is passed to the messages log.

Display SQL. Specifies whether any SQL that is displayed in the log should contain native SQL functions or standard ODBC functions of the form {fn FUNC(...)}, as generated by SPSS Modeler. The former relies on ODBC driver functionality that may not be implemented. For example, this control would have no effect for SQL Server.

Reformat SQL for improved readability. Specifies whether SQL displayed in the log should be formatted for readability.

Show status for records. Specifies when records should be reported as they arrive at terminal nodes. Specify a number that is used for updating the status every N records.

Click Default Values to revert to the system default settings for this tab.

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 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. 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 SPSS Modeler and the database. Database caching can further increase the benefits. For more information, see the file *DatabaseMiningGuide.pdf*, available under the *\Documentation* folder on the IBM® SPSS® Modeler installation disk.

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](#).

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. [For more information, see the topic Database Export Advanced Options in Chapter 7 in IBM SPSS Modeler 14.2 Source, Process, and Output Nodes.](#)

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 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.

Table 5-1
Performance benefit of bulk-loading

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

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 datasets, these **pushbacks** can dramatically enhance performance in several ways:

- By reducing the size of the result set to be transferred from the DBMS to SPSS Modeler. When large result sets are read through an ODBC driver, network I/O or driver inefficiencies may result. For this reason, the operations that benefit most from SQL optimization are row and column selection and aggregation (Select, Sample, Aggregate nodes), which typically reduce the size of the dataset 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 making use of 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, SPSS Modeler is designed to maximize the amount of SQL generated by each stream so that only those operations that cannot be compiled to SQL are executed by 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 the topic Tips for Maximizing SQL Generation on p. 50.](#)

Note: Because of minor differences in SQL implementation, streams executed in a database may return slightly different results when executed in 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, SPSS Modeler streams executed using SQL pushback may produce different results from those executed without SQL pushback. Contact your database administrator for advice on configuring your database. To maximize compatibility with SPSS Modeler, database string comparisons should be case sensitive.

Note: Database modeling and SQL optimization require that SPSS Modeler Server connectivity be enabled on the SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler. [For more information, see the topic Connecting to IBM SPSS Modeler Server in Chapter 3 in IBM SPSS Modeler 14.2 User's Guide.](#)

Database Requirements

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

Note that you may gain additional performance improvements by using database modeling. For more information, see the topic [Database Modeling Overview in Chapter 2 in IBM SPSS Modeler 14.2 In-Database Mining Guide](#).

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. For more information, see the topic [Tips for Maximizing SQL Generation on p. 50](#).

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.

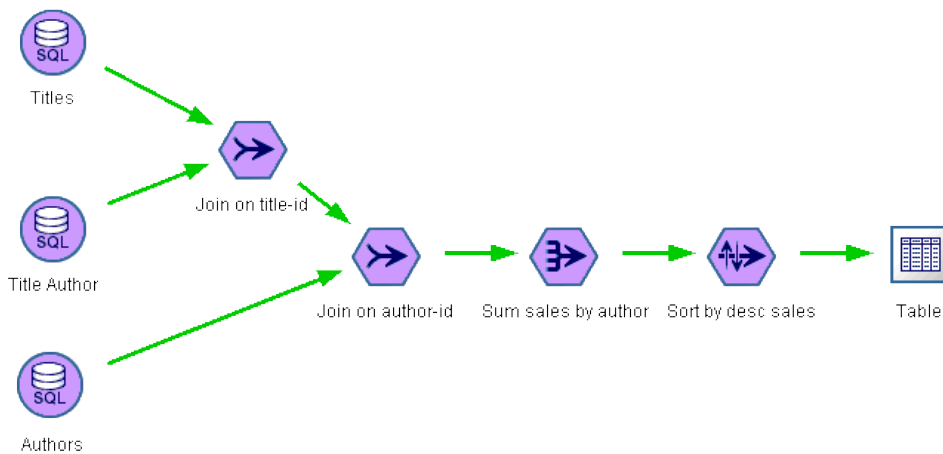
- **Field projection.** SPSS Modeler Server extracts only fields that are required for subsequent processing from the database, which minimizes bandwidth and memory requirements. The same is also true for superfluous fields in flat files: although the server must read the superfluous fields, it does 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 6-1

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



Generated SQL

The generated SQL for this stream is as follows:

```

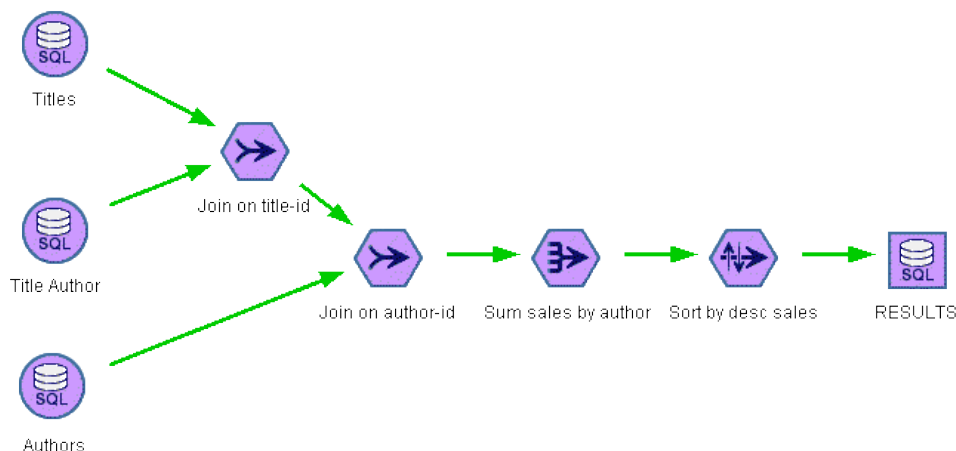
SELECT
  T2.au_lname AS C0,
  T2.au_fname AS C1,
  SUM({fn CONVERT(T0.ytd_sales,SQL_BIGINT)}) AS C2
FROM
  dbo.titles T0,
  dbo.titleauthor T1,
  dbo.authors T2
WHERE
  (T0.title_id = T1.title_id )
  AND (T1.au_id = T2.au_id )
GROUP BY T2.au_lname ,T2.au_fname
ORDER BY 3 DESC

```

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 6-2
Entire stream executed in database



Configuring SQL Optimization

- ▶ Install an ODBC driver and configure a data source for the database you want to use. [For more information, see the topic Data Access in Chapter 2 on p. 9.](#)
- ▶ Create a stream that uses a source node to pull data from that database.
- ▶ 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

- ▶ From the Tools menu, choose User Options.
- ▶ Click the Optimization tab. Select Generate SQL to enable SQL optimization. Optionally, you can select other settings to improve performance. [For more information, see the topic Client Performance and Optimization Settings in Chapter 5 on p. 39.](#)

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 in Chapter 4.](#)

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. [For more information, see the topic Using Model Nuggets in Streams in Chapter 3 in IBM SPSS Modeler 14.2 Modeling Nodes.](#)

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.

- ▶ 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. [For more information, see the topic Client Performance and Optimization Settings in Chapter 5 on p. 39.](#)
- ▶ On the stream canvas, select the node or stream that you want to preview.
- ▶ Click the Preview SQL button on the toolbar.

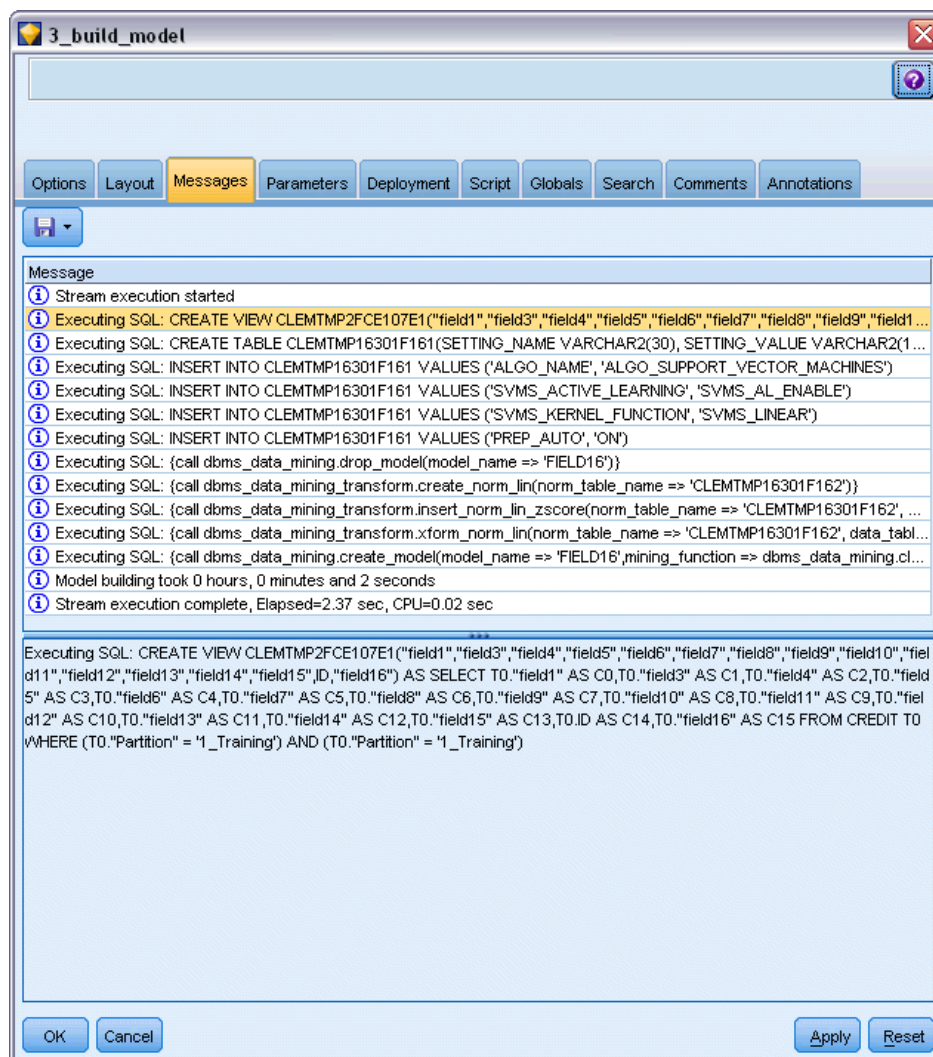
All nodes for which SQL is generated (and that will be pushed back to the database when the stream is executed) are colored purple on the stream canvas.

Figure 6-3
Preview SQL button



- ▶ To preview the generated SQL, from the menus choose:
Tools > Stream Properties > Messages...

Figure 6-4
Generated SQL displayed in message log



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. [For more information, see the topic Nodes Supporting SQL Generation on p. 51.](#)

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

- ▶ Select the Settings tab on the model nugget.
- ▶ Choose one of the options Generate with (no) missing value support or Generate SQL for this model, as appropriate.

- ▶ In the model nugget menu, choose:
File > Export SQL
- ▶ Save the file.
- ▶ 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. [For more information, see the topic Nodes Supporting SQL Generation on p. 51.](#)

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. [For more information, see the topic CLEM Expressions and Operators Supporting SQL Generation on p. 54.](#)

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. [For more information, see the topic Database Modeling Overview in Chapter 2 in IBM SPSS Modeler 14.2 In-Database Mining Guide.](#)

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. [For more information, see the topic Nodes Supporting SQL Generation on p. 51.](#)

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 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. [For more information, see the topic Configuring SQL Optimization on p. 47.](#) Also, see [Caching Options for Nodes](#) 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.

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. [For more information, see the topic Previewing Generated SQL on p. 48.](#)

Sources

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. For more information, see the topic Writing SQL Queries on p. 57.

Record Operations

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 as shown in the next table. Complex sampling does not support SQL generation.
Aggregate	
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	

Node supporting SQL generation	Notes
Merge	No SQL generated for merge by order. 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. For all types of merge, SQL_SP_EXISTS is not supported if inputs originate in different databases.
Append	Supports generation if inputs are unsorted.
Distinct	

Support for SQL generation in the Sample node when using Simple sampling is shown in the next table.

Mode	Sample	Max size	Seed	DB2 OS/Z	DB2 OS/400	DB2 Windows/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			Y	Y	Y		Y
			on			Y		Y		
		max	off			Y	Y	Y		Y
on					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			Y	Y	Y		Y
			on			Y		Y		
		max	off			Y	Y	Y		Y
			on			Y		Y		

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	

Node supporting SQL generation	Notes
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.
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	

Graphs

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.
Distribution	
Web	
Evaluation	

Generated Model Nodes

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. [For more information, see the topic Viewing SQL for Model Nuggets on p. 49.](#)

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	

Output

Node supporting SQL generation	Notes
Table	Supports generation if SQL generation is supported for highlight expression (see expressions below).
Matrix	Supports generation except if “All numerics” is selected for the Fields option.
Analysis	Supports generation, depending on the options selected.
Transform	
Statistics	Supports generation if the Correlate option is not used.
Report	
Set Globals	

Export

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.

Operators

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

Relational Operators

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

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	
member	
min	
negate	
not	
number	
or	
pi	
real	
rem	
round	
sign	
sin	
sqrt	

Operation supporting SQL generation	Notes
string	
strmember	
subscrs	
substring	
substring_between	
uppertolower	
to_string	

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	

Aggregate Functions

Operation supporting SQL generation	Notes
Sum	
Mean	
Min	
Max	
Count	
SDev	

Using SQL Functions in CLEM Expressions

The @SQLFNfunction 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 in 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;
```

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 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 SPSS Modeler Server machine is running in a different locale or if the shell in which 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:

- 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 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 SPSS Modeler Server at the start of a database connection. This means that SPSS Modeler is unable to optimize streams for Oracle.

To check and reset locale specifications:

- ▶ 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
```

- ▶ Change to your Connect ODBC/locale directory. (Here you will see a single directory, *en_US*.)

- ▶ 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
```

- ▶ Once you have created the link, restart SPSS Modeler Server from this same shell. (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 SPSS Modeler Server. *LANG* may be defined in the shell as you would any other environment variable before restarting SPSS Modeler Server. For example, see the following definition:

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

- Each time you start 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.

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 SPSS Modeler Server to start automatically when the server computer is restarted.
- Manually stop and restart SPSS Modeler Server.
- Change permissions on files created by SPSS Modeler Server.
- Configure SPSS Modeler Server to work with the ODBC Connect drivers provided with SPSS Modeler Server. [For more information, see the topic IBM SPSS Modeler Server and the Data Access Pack on p. 62.](#)
- Configure SPSS Modeler Server to use the IBM® SPSS® Collaboration and Deployment Services Enterprise View Driver when reading data with an Enterprise View node. [For more information, see the topic Configuring a Driver for the Enterprise View Node on p. 68.](#)

Scripts

IBM® SPSS® Modeler Server uses several scripts, including:

- **modelersrv.sh.** The manual startup script for SPSS Modeler Server is located in the 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 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.

Table B-1
Location of *rc.modeler* by operating system

Operating system	Location
AIX	<i>/etc/rc.modeler</i>
HP-UX	<i>/sbin/init.d/rc.modeler</i>
Solaris	<i>/etc/init.d/rc.modeler</i>

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 SPSS Modeler Server.

To configure the system for automatic startup and shutdown

- ▶ Log on as root.
- ▶ Change to the SPSS Modeler Server installation directory.
- ▶ 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 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

- ▶ Change to the SPSS Modeler Server installation directory.
- ▶ To start the server, at the UNIX command prompt, type:

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

- ▶ 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

- ▶ Stop IBM® SPSS® Modeler Server. (For more information, see the topic [Manually Starting and Stopping IBM SPSS Modeler Server on p. 61.](#))
- ▶ Locate the appropriate script. (For more information, see the topic [Scripts on p. 60.](#))
- ▶ Open the script in a text editor, make changes, and save the file.

- ▶ Start 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](#) above.) We recommend **077** as the most restrictive UMASK setting to use. Settings that are more restrictive could cause permissions problems for SPSS Modeler Server.

IBM SPSS Modeler Server and the Data Access Pack

If you want to use the Connect for ODBC drivers with IBM® SPSS® Modeler Server, the Connect for ODBC environment must be configured by *odbc.sh* when the SPSS Modeler Server process starts. You do this by editing the appropriate IBM® SPSS® Modeler startup script, either in *modelersrv.sh*, *rc.modeler*, or in both. (For more information, see the topic [Editing Scripts on p. 61](#).)

For more information, see the corporate Web site at <http://www.spss.com/drivers/>. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

To configure Connect for ODBC to start with IBM SPSS Modeler Server

- ▶ From the corporate Web site at <http://www.spss.com/drivers/clientCLEM.htm>, download the relevant compressed tar archive for the platform on which you have SPSS Modeler Server installed. Make sure you download the drivers for the correct version of SPSS Modeler Server that you have installed. Copy the file to the location where you want to install the ODBC drivers (for example, */usr/spss/odbc*).
- ▶ Extract the archive.
- ▶ Run the *setodbcpath.sh* script that is extracted from the archive.
- ▶ 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 SPSS Modeler to read to get a list of the ODBC datasources that you define (a default *odbc.ini* is installed with the drivers).

- ▶ Save *odbc.sh*.
- ▶ (64-bit 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”.

Thus your final *odbc.sh* file on a 64-bit 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.

- ▶ 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).
- ▶ Save the *odbc.ini* file.
- ▶ Configure 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.

- ▶ Save *modelersrv.sh*.

To test the connection

- ▶ Restart SPSS Modeler Server.
- ▶ Connect to SPSS Modeler Server from a client.

- ▶ On the client, add a Database source node to the canvas.
- ▶ 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. [For more information, see the topic Troubleshooting Connect for ODBC Configuration on p. 65.](#)

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

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

- ▶ Edit the *modelerrun* script in 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.

- ▶ Save the *modelerrun* script file.

To configure Connect for 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 SPSS Modeler Server from SPSS Modeler Batch in order to run streams. Ensure that the 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

- ▶ Edit the *odbc.ini* file to include the new or changed name.
- ▶ Test the connection as described earlier in this section.

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

Troubleshooting Connect for 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.

- ▶ 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*.
- ▶ 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.
- ▶ 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
```

- ▶ Save and then execute *modelersrv.sh*. The value of \$ODBCINI that is being set in the SPSS Modeler Server environment is written to *stdout* for verification.
- ▶ 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 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 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 SPSS Modeler ServerODBC 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 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=
```

- ▶ Check that the shared object specified by the Driver parameter exists.
- ▶ Correct the path to the shared object if it is incorrect.
- ▶ 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. [For more information, see the topic IBM SPSS Modeler Server and the Data Access Pack on p. 62.](#) 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 Connect for 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:

- ▶ Use *ivtestlib* to confirm that the driver cannot be loaded. For example, at the UNIX prompt, type:

```
sh
cd ODBCDIR
. odbc.sh
./bin/ivtestlib MFor815
```

where *ODBCDIR* is replaced by the path to your Connect for ODBC installation directory.

- ▶ 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 (for example, not on *LD_LIBRARY_PATH* on Solaris).
- ▶ 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 SPSS Modeler 64-bit on Linux or Solaris, 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
```

- ▶ 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.
- ▶ Correct the library path in the SPSS Modeler startup script.
- ▶ Restart the SPSS Modeler Server with the startup script that you edited (*modelersrv.sh* or *rc.modeler*).

Library Paths

The name of the library path variable varies depending on your operating system. The following table is included as a guide to help you make appropriate substitutions when you are configuring or troubleshooting on your system.

Table B-2
Library path by operating system

Operating system	Library path variable name
AIX	LIBPATH
HP-UX	SHLIB_PATH
Solaris or Linux 32-bit	LD_LIBRARY_PATH
Solaris or Linux 64-bit	LD_LIBRARY_PATH_64

Configuring a Driver for the Enterprise View Node

The Enterprise View node enables you to create and maintain a connection between an IBM® SPSS® Modeler session and an IBM SPSS Collaboration and Deployment Services Enterprise View in a shared IBM® SPSS® Collaboration and Deployment Services Repository. [For more information, see the topic Storing and Deploying IBM SPSS Collaboration and Deployment Services Repository Objects in Chapter 9 in IBM SPSS Modeler 14.2 User's Guide.](#)

To use the Enterprise View node with IBM® SPSS® Modeler Server, an IBM® SPSS® Collaboration and Deployment Services Enterprise View Driver must be installed and configured. For details on installing the driver, contact your local administrator. After installation, a reference to *pev.sh* must be added in the startup script—either in *modelersrv.sh*, *rc.modeler*, or in both—following the format:

```
./usr/odbc/pev.sh
```

Running 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 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. It is possible, however, to run the daemon process on a non-root account by following the instructions below.

Running as non-root. Without root privilege, SPSS Modeler authenticates users against a private password database, distinct from the UNIX password database, and starts all user sessions on the same UNIX user account. 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.

Configuring IBM SPSS Modeler Server As a Non-Root Process

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. You must still login as root, or su to root, to start and stop the server and to perform general administration.

To configure SPSS Modeler Server to run on a non-root account, 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 SPSS Modeler Server. This account is for the sole use of the SPSS Modeler Server daemon process. In other words, you cannot use this account to start and stop the server or to perform general IBM® SPSS® Modeler administration. For this example, let's call it *modelerserv*.

When creating the account, note that:

- The account does not need login capability.
- The primary group should be the *<modelerusers>* group created previously.
- The home directory can be the 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 SPSS Modeler Server using the newly created account. Locate the appropriate startup script and open it in a text editor. [For more information, see the topic Scripts in Appendix B on p. 60.](#)

- ▶ In the appropriate script, change the line that sets the `MODELERUSER` variable to reference the name of the user account created to run SPSS Modeler Server. For example:

```
MODELERUSER=modelerserv
```

- ▶ 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. 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 SPSS Modeler. *Note:* These are private to 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 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.

6. Recursively change the ownership of the SPSS Modeler installation directory and its contents to be user `<modelerserv>` and group `<modelerusers>` where the names referenced are those you created earlier. For example:

```
chown -R -h modelerserv:modelerusers .
```

7. Consider creating subdirectories in the data directory for your SPSS Modeler users so that they have somewhere to store working data without interference. These directories should be

group-owned by the *<modelerusers>* group and have group read, write, and search permissions. For example, to create a working directory for user *bob*:

```
mkdir data/bob
chown bob:modelerusers 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 *<modelerusers>*:

```
chmod g+s data/bob
```

Running IBM SPSS Modeler Server as a non-root user on AIX

When starting SPSS Modeler Server as a non-root user on AIX, there is the chance that the error *lsattr not found* will be printed to the console. This is caused because *lsattr* is located in the */usr/sbin* directory, but when SPSS Modeler Server logs in a new user (to run as non-root) it resets the path to include just */usr/bin* and */bin*.

If this error occurs, the solution is to make the following link under */usr/bin*:

```
su
cd /usr/bin
ln -s /usr/sbin/lsattr lsattr
```

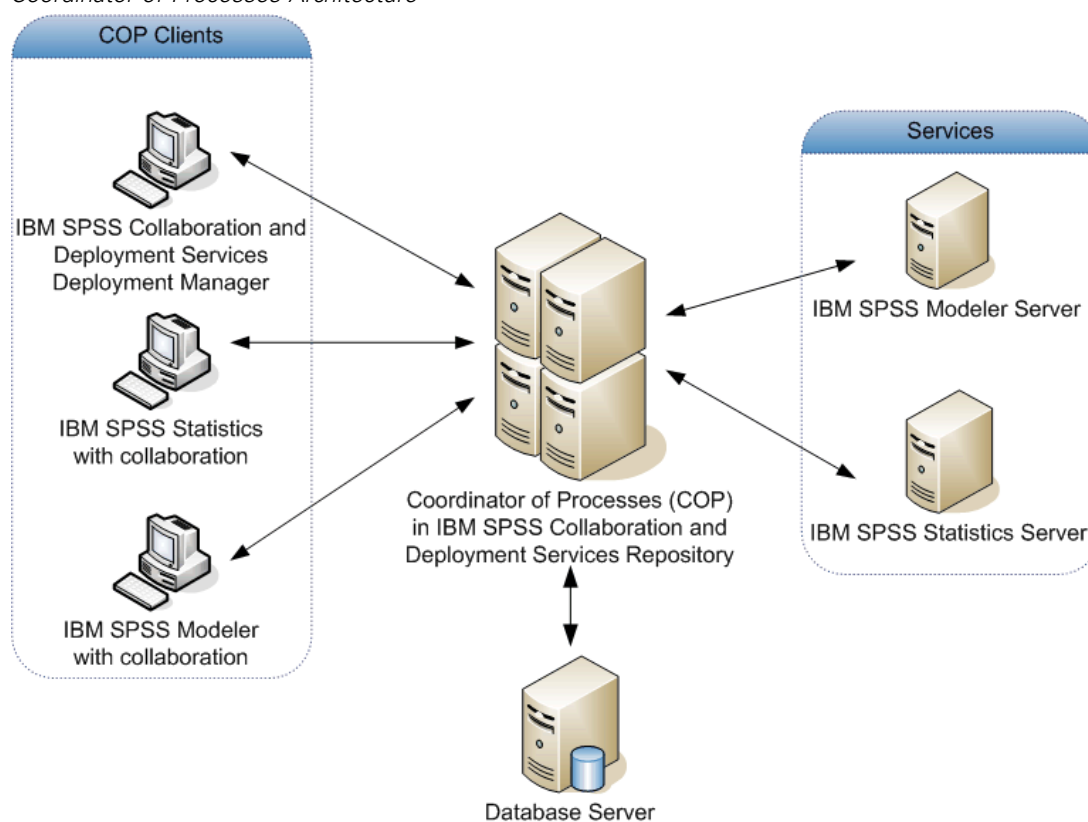
After making the link, restart SPSS Modeler Server.

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 D-1
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 on request from IBM Corp.

Whenever you connect to a server or server cluster in SPSS Modeler, you can enter a server manually or search for a server or cluster using the Coordinator of Processes. [For more information, see the topic Connecting to IBM SPSS Modeler Server in Chapter 3 in *IBM SPSS Modeler 14.2 User's Guide*.](#)

Notices

This information was developed for products and services offered worldwide.

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 Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (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. 1623-14, Shimotsuruma, Yamato-shi Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Software Group Attention: Licensing 233 S. Wacker Dr. Chicago, IL 60606 USA

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All 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 the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, and SPSS are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.html>.

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.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other product and service names might be trademarks of IBM or other companies.

Index

- 64-bit operating systems, 8
 - adding IBM SPSS Modeler Server connections, 14–15
 - administration
 - of IBM SPSS Modeler Server, 20
 - of IBM SPSS Statistics license location, 31
 - administrator access
 - for IBM SPSS Modeler Server, 21
 - allow_modelling_memory_override
 - options.cfg* file, 25
 - application examples, 2
 - architecture
 - components, 5
 - authentication, 16
 - auto.sh* (UNIX)
 - location of, 60
 - automatic server startup
 - configuring on UNIX, 61
 - cache compression, 25
 - cache_compression
 - options.cfg* file, 25
 - cache_connection option, 31
 - caching, in-database, 30
 - chemsrv.sh* (UNIX)
 - location of, 60
 - CLEM expressions
 - SQL generation, 54
 - configuration options
 - automatic SQL generation, 27
 - connections and sessions, 23
 - coordinator of processes, 28
 - COP, 28
 - data file access, 24
 - logging, 23
 - login attempts, 23
 - memory management, 25
 - of IBM SPSS Modeler Server, 20
 - overview, 23
 - parallel processing, 25
 - performance and optimization, 25
 - port number, 23
 - SQL string length, 27
 - SSL data encryption, 27
 - stream rewriting, 25
 - temp directory, 24
 - Connect for ODBC
 - configuring on UNIX, 62
 - connections
 - server cluster, 15
 - to IBM SPSS Modeler Server, 12, 14–15
 - Coordinator of Processes, 15
 - load balancing, 72
 - server clusters, 72
 - coordinator of processes configuration
 - for IBM SPSS Modeler Server, 28
 - COP, 15
 - load balancing, 72
 - server clusters, 72
 - COP configuration
 - for IBM SPSS Modeler Server, 28
 - cop_enabled
 - options.cfg* file, 28
 - cop_host
 - options.cfg* file, 28
 - cop_password
 - options.cfg* file, 28
 - cop_port_number
 - options.cfg* file, 28
 - cop_service_description
 - options.cfg* file, 28
 - cop_service_host
 - options.cfg* file, 28
 - cop_service_name
 - options.cfg* file, 28
 - cop_service_weight
 - options.cfg* file, 28
 - cop_update_interval
 - options.cfg* file, 28
 - cop_user_name
 - options.cfg* file, 28
- data access, 9
- data access pack
 - and UNIX library paths, 68
 - configuring UNIX for, 62
 - Connect for ODBC, configuring on UNIX, 62
 - troubleshooting Connect for ODBC on UNIX, 65
- data files
 - IBM SPSS Statistics, 11
 - importing and exporting, 11
- data_file_path
 - options.cfg* file, 24
- data_files_restricted
 - options.cfg* file, 24
- database
 - support tiers, 10
- database caching
 - controlling from *options.cfg*, 30
 - SQL generation, 51
- database connections
 - closing, 31
- database servers, 42
- databases
 - accessing, 9
- DB2
 - SQL optimization, 44–45
- disk space, 8
 - calculating, 9
- documentation, 2

- domain name (Windows)
 - IBM SPSS Modeler Server, 12
- encryption
 - SSL, 32
- error on stream execution, 26
- examples
 - Applications Guide, 2
 - overview, 3
- file permissions
 - configuring on UNIX, 62
 - on IBM SPSS Modeler Server, 17
- filenames
 - UNIX, 11
 - Windows, 11
- firewall settings
 - options.cfg* file, 26
- hard drives, 9
- hardware recommendations
 - for IBM SPSS Modeler Server, 7
- host name
 - IBM SPSS Modeler Server, 12, 14
- IBM SPSS Modeler, 1
 - documentation, 2
- IBM SPSS Modeler Administration Console, 20
 - administrator access, 21
- IBM SPSS Modeler Server
 - administration of, 20
 - administration options, 20
 - administrator access, 21
 - configuration options, 23
 - coordinator of processes configuration, 28
 - COP configuration, 28
 - different results than client, 17
 - domain name (Windows), 12
 - file creation, 17
 - host name, 12, 14
 - information for end users, 16
 - log files, 23
 - monitoring usage, 29
 - password, 12
 - port number, 12, 14, 23
 - server processes, 29
 - temp directory, 24
 - unresponsive processes, 19
 - user accounts, 16
 - user authentication, 16
 - user ID, 12
- IBM SPSS Statistics data access technology, 9
- IBM SPSS Statistics data files
 - importing and exporting, 11
- IBM SPSS Statistics license location, 31
- IBM SPSS Text Analytics, 2
- in-database caching, 30
- in-database mining, 42
- io_buffer_size
 - options.cfg* file, 25
- kernel limits on UNIX, 19
- LDAP, 36
 - securing, 36
- legal notices, 74
- log files
 - displaying generated SQL, 41, 48
 - for IBM SPSS Modeler Server, 23, 37
- log_database_access
 - options.cfg* file, 23
- log_database_errors
 - options.cfg* file, 23
- log_directory
 - options.cfg* file, 23
- log_file_access
 - options.cfg* file, 23
- log_file_errors
 - options.cfg* file, 23
- logging in to IBM SPSS Modeler Server, 12
- max_file_size
 - options.cfg* file, 24
- max_login_attempts
 - options.cfg* file, 23
- max_parallelism
 - options.cfg* file, 25
- max_sessions
 - options.cfg* file, 23
- max_sql_string_length
 - options.cfg* file, 27
- memory, 9
- memory management
 - administration options, 25
- memory_usage
 - options.cfg* file, 25
- messages
 - displaying generated SQL, 41, 48
- messages.log* file, 37
- Microsoft SQL Server
 - SQL optimization, 44–45
- model nuggets
 - viewing SQL for, 49
- modeling
 - memory management, 25
- modelling_memory_limit_percentage
 - options.cfg* file, 25
- multiple stream execution, 26
- node caching
 - SQL generation, 51
 - writing to database, 51

- nodes
 - that support SQL generation, 51
- ODBC data sources
 - and UNIX, 62
 - Connect for ODBC and UNIX scripts, 62
- operating systems
 - 64-bit, 8
- operators
 - SQL generation, 54
- optimization, 39
 - SQL generation, 44–45, 47
- options.cfg* file, 30
- Oracle
 - SQL optimization, 44–45, 58
- parallel processing
 - controlling, 25
 - enabling, 39
- password
 - IBM SPSS Modeler Server, 12
- paths, 11
- performance
 - of IBM SPSS Modeler Server, 39
- port number
 - IBM SPSS Modeler Server, 12, 14, 23
- port settings
 - options.cfg* file, 25
- port_number
 - options.cfg* file, 23
- preview
 - SQL generation, 48
- processes, unresponsive, 19
- processors, 7
 - multiple, 25
- program_file_path
 - options.cfg* file, 24
- program_files_restricted
 - options.cfg* file, 24
- purple nodes, 39
 - SQL optimization, 46
- pushbacks, 39, 44–45, 47, 51
 - CLEM expressions, 54
- RAM, 9
- rc.modeler* (UNIX)
 - location of, 60
- results
 - differences between Client and Server, 17
 - record order, 17
 - rounding of, 17
- searching COP for connections, 15
- Secure Sockets Layer, 32
- securing
 - LDAP, 36
- security
 - configuring file creation on UNIX, 62
 - file creation, 17
 - SSL, 32
- server
 - adding connections, 14
 - logging in, 12
 - searching COP for servers, 15
- server port settings
 - options.cfg* file, 25
- single sign-on, 13
- SPSS Modeler Server, 1
- SQL
 - duplicate column names, 57
 - optimizing Oracle, 58
 - previewing generated, 48
 - queries, 57
 - viewing for model nuggets, 49
- SQL generation, 39, 44–45, 51
 - CLEM expressions, 50, 54
 - enabling, 47
 - enabling for IBM SPSS Modeler Server, 27
 - logging, 41, 48
 - previewing, 41, 48
 - stream rewriting, 50
 - tips, 50
 - viewing for model nuggets, 49
- SQL pushback. *See also* SQL generation, 44
- SQL Server
 - SQL optimization, 44–45
- sql_generation_enabled
 - options.cfg* file, 27
- @SQLFN function, 56
- SSL, 32
 - overview, 32
 - securing communications, 32
- SSL data encryption
 - enabling for IBM SPSS Modeler Server, 27
- ssl_certificate_file
 - options.cfg* file, 27
- ssl_enabled
 - options.cfg* file, 27
- ssl_private_key_file
 - options.cfg* file, 27
- ssl_private_key_password
 - options.cfg* file, 27
- starting IBM SPSS Modeler Server
 - on UNIX, 18
 - on Windows, 18
- status
 - of IBM SPSS Modeler Server on UNIX, 18
 - of IBM SPSS Modeler Server on Windows, 18
- stopping IBM SPSS Modeler Server
 - on UNIX, 18
 - on Windows, 18
- stream rewriting, 50
 - enabling, 39

stream_rewriting_enabled
 options.cfg file, 25

temp directory
 for IBM SPSS Modeler Server, 24
temp_directory
 options.cfg file, 24
temporary files, 8
 permissions for (IBM SPSS Modeler Server), 17
tiers, database support, 10
trademarks, 75

UNC filenames, 11

UNIX

 configuring file permissions, 62
 library paths, 68
 user authentication, 16

UNIX kernel limits, 19

UNIX scripts

auto.sh, 60
 editing, 61
 modelersrv.sh, 60
 rc.modeler, 60

URL prefix, 36

user accounts

 IBM SPSS Modeler Server, 16

user authentication, 16

user ID

 IBM SPSS Modeler Server, 12

zombie processes, IBM SPSS Modeler Server, 19