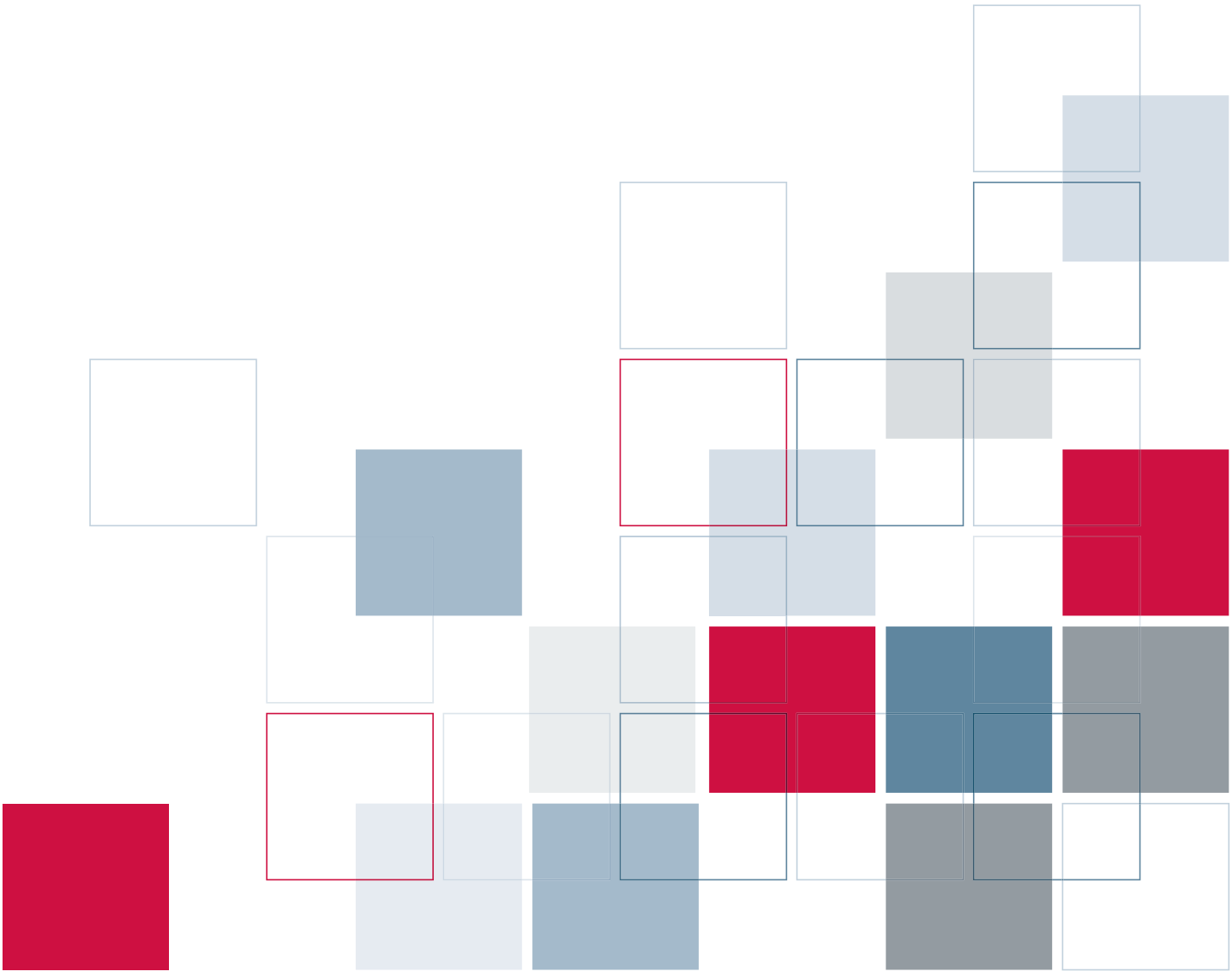
PASW® Collaboration and Deployment Services 4 Installation and Configuration Guide (Windows)



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

Printed in the United States of America.

Preface

PASW Collaboration and Deployment Services enable widespread use and deployment of predictive analytics with features like centralized, secure, and auditable storage of analytical assets, advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms of delivering the results of analytical processing to the end users.

This manual, , documents the software and hardware requirements for PASW Collaboration and Deployment Services and the system installation and configuration. Tasks such as setting up content repository server, managing users, auditing the repository, etc. are documented in the *PASW Collaboration and Deployment Services4 Administrator's Guide*. The tasks associated with everyday use of the analytical facilities of PASW Collaboration and Deployment Services are documented in *Deployment Manager 4 User's Guide*.

Technical Support

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

Tell Us Your Thoughts

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

Contents

Chapter
1

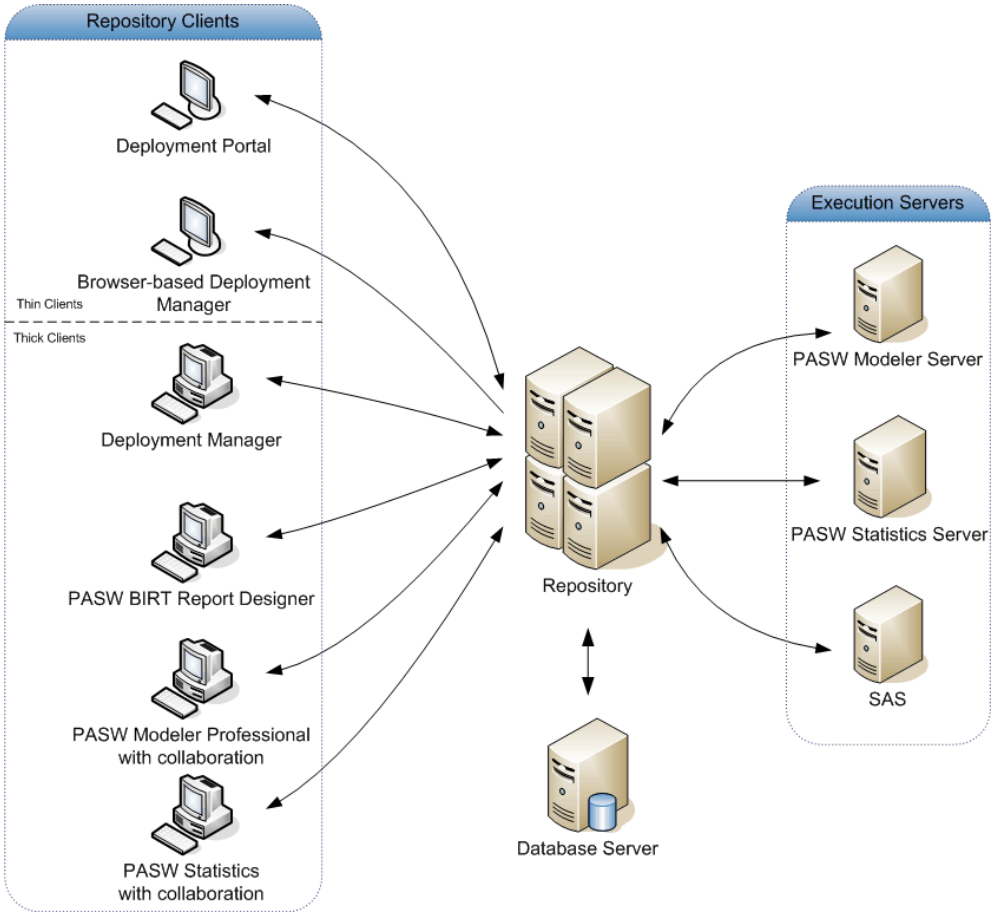
System Overview

PASW Collaboration and Deployment Services is an enterprise-level application that enables widespread use and deployment of predictive analytics. PASW Collaboration and Deployment Services provides centralized, secure, and auditable storage of analytical assets and advanced capabilities for management and control of predictive analytic processes, as well as sophisticated mechanisms for delivering the results of analytical processing to the end users. The benefits of PASW Collaboration and Deployment Services include safeguarding the value of analytical assets, ensuring compliance with regulatory requirements, improving the productivity of analysts, and minimizing the IT costs of managing analytics.

PASW Collaboration and Deployment Services consists of the following components:

- Repository for analytical artifacts
- Deployment Manager
- Deployment Portal
- Browser-based Deployment Manager
- Enterprise View
- PASW BIRT Report Designer

Figure 1-1
PASW Collaboration and Deployment Services Architecture



PASW Collaboration and Deployment Services requires access to one or more execution servers to perform model building, scoring, and other analyses. Analytical processing can also be performed by PASW Statistics and PASW Modeler desktop applications used as repository clients with collaboration.

Repository

The repository provides a centralized location for storing analytical assets, such as models and data.Repository features include:

- Security
- Version control
- Searching
- Auditing

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

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

Deployment Manager

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

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

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

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

Deployment Portal

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

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

Browser-based Deployment Manager

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

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

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

Enterprise View

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

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

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

Execution Servers

Execution servers provide the ability to execute objects stored within the repository. Execution servers currently supported by PASW Collaboration and Deployment Services include:

- **PASW Modeler**. The PASW Modeler execution server is PASW Modeler Server, which permits distributed analysis for data mining and model building. Use this execution server to process PASW Modeler streams.
- **PASW Statistics**. The PASW Statistics execution server corresponds to the batch facility included with the PASW Statistics Server product. However, unlike the PASW Statistics server product, the batch facility can be run from the command line and requires no user credentials for execution. Use this execution server to process PASW Statistics syntax files.

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

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

PASW BIRT Report Designer

The reporting functionality of PASW Collaboration and Deployment Services is enabled by BIRT (Business Intelligence and Reporting Tools), an open-source package distributed by Eclipse Foundation under the Eclipse Public License. BIRT provides core reporting features, such as report layout, data access, and scripting. For more information about BIRT, see the BIRT project page (*http://www.eclipse.org/birt*). The PASW Collaboration and Deployment Services installation includes the BIRT reporting engine server components, which enable the execution of BIRT report syntax files as part of the PASW Collaboration and Deployment Services reporting job steps. PASW BIRT Report Designer is a standalone application that can be used in conjunction with PASW Collaboration and Deployment Services. It provides a rich user interface with a number of advanced features for creating reports and must be installed separately.

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

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

Chapter 2

What's New in This Release?

PASW Collaboration and Deployment Services 4 introduces the following new features and enhancements that impact system configuration:

PASW Collaboration and Deployment Servicesrepository can be deployed into a clustered application server environment to ensure provide high availability, load balancing, and failover of the enterprise analytical assets.

The repository and PASW Collaboration and Deployment Services clients can be installed in virtualized server environments thus enabling more efficient utilization of enterprise computing resources. If the virtualized server environment configuration meets the minimum system requirements for PASW Collaboration and Deployment Services, performance equals that of a non-virtual installation.

The repository and its clients can be configured to use a FIPS 140-2 validated cryptography module for all necessary encryption and decryption, providing Security Level 1 as specified by FIPS 140-2.

Chapter
3

Installing PASW Collaboration and Deployment Services

This chapter provides the information about the installation and configuration of the following PASW Collaboration and Deployment Services components:

- The repository
- Web installation modules for client components
- Remote Process Server
- Python Scripting

Configuration of the repository environment may consist of:

Provisioning. Certain prerequisites must be in place before beginning the installation. This includes verifying certain hardware and software requirements are met, setting up database connections and tables, and determining the installation directory of the application server PASW Collaboration and Deployment Services will use for distributed access.

Installing. New users must perform a clean installation of the repository in Windows, UNIX, or IBM i environment.

Licensing. Once the installation is successful, a license code must be specified before you can begin using the repository.

Upgrading. Users with an existing version of the repository can conveniently upgrade their environment to take advantage of new features and functions.

Uninstalling. In the event that an installation becomes corrupt or the application needs to be reinstalled due to system errors, the repository can be removed and the system restored to its original state.

When finished, verify the installation is successful and install Deployment Manager on client workstations that will connect to the repository.

Provisioning the System

Prior to installing the repository, verify that the necessary application server, database configuration, hardware, software, and permissions requirements have been met.

Hardware Requirements

The following hardware requirements must be met prior to installing the repository. Note that this does not reflect the hardware requirements of software beyond the repository, such as operating systems and databases.

Table 3-1
Hardware requirements

Component	Requirement
Processor	At least Pentium 1.8 GHz
Hard Drive	At least 5 GB of free space
Memory	At least 4 GB RAM
Optical drive	DVD-ROM

Software Requirements

The repository can be installed into application servers running on the following operating system(s):

- Windows 2003 Standard and Enterprise (32- and 64-bit)

Other requirements include:

- J2SE 5.0 appropriate to the application server selected for the installation. For more information, see application server vendor documentation.
- For repository access through a Web browser, Internet Explorer 6.0 or 7.0, Firefox 2.0 or 3.0, or Safari 3.1.2.

Note: Safari cannot be used if your PASW Collaboration and Deployment Services installation is configured to be compliant with FIPS 140-2 security standard or single sign-on.

File System Permissions

File System Permissions

The user installing PASW Collaboration and Deployment Services must have the following permissions on the host system:

- Write permissions to the PASW Collaboration and Deployment Services installation directory and subdirectories.
- Execute permissions for all library files under *<PASW Collaboration and Deployment Services Installation Directory>/components/setup/jni/[win64|windows]*.
- Write permissions to the deployment and configuration directories and read and execute permissions to other application server directories.

Application Servers

Before installing the repository, a supported application server or a server cluster must be installed and accessible. The repository installation requires a connection to the application server to deploy the necessary Web services and components. If the repository is reinstalled, it is strongly recommended to use a new instance of the application server. It is also essential to make sure the latest versions of vendor patches have been applied to application server installations.

Supported application servers include:

- JBoss 4.2.0
- BEA WebLogic 9.2 and 10.0
- IBM WebSphere 6.1
- Oracle Application Server 10g Release 3 (10.1.3.1.0)
- SAP NetWeaver 7.1

Note: Application server clustering is supported only for WebSphere and WebLogic application servers.

Whether or not the application server should be running during installation depends on the server.

- For deployment into JBoss, the application server should not be running.
- For deployment into Oracle Application Server, the application server should not be running.
- For deployment into WebLogic, the application server should not be running.
- For deployment into WebSphere, the application server should be running.
- Configuring single sign-on for PASW Collaboration and Deployment Services running on WebSphere 6.1 requires Patch 19.

Notes:

- For Oracle AS 10g R3 installations, it is strongly recommended that the application server and the repository be installed under the same user credentials.
- Setting up message-based job processing on Oracle 10g to enable durable subscriptions requires that the *clientID* property of the *ConnectionFactory* JMS configuration attribute be specified using the application server administration console.
- For JBoss application server, it is recommended that only one instance of the server be run. If multiple instances of JBoss application server to be used with the repository must be set up on a single machine, consult vendor documentation.
- For repository setup on WebSphere, IBM J2SE 5.0 is required.
- For details about NetWeaver configuration following PASW Collaboration and Deployment Services installation, see Appendix E.

For additional information on installing an application server, refer to the vendor documentation.

Databases

Before installing the repository, a supported database must be running and accessible. Repository installation requires a connection to the database to establish the necessary control tables and infrastructure. Supported databases include:

- Microsoft SQL Server 2005 (32-bit, Standard or Enterprise Editions) running Windows 2003 Server (32-bit)
- Microsoft SQL Server 2005 (64-bit, Standard or Enterprise Editions) running Windows 2003 Server (64-bit)
- Oracle 10g (32-bit, Standard or Enterprise Editions) running on Windows 2003 Server or Solaris 9
- Oracle 10g (64-bit, Enterprise Edition) running on Windows 2003 Server for 64-bit extended systems, Solaris 9, or Solaris 10
- Oracle 11g (32-bit, Standard or Enterprise Editions) running on Windows 2003 Server or Solaris 9
- Oracle 11g (64-bit, Enterprise Edition) running on Windows 2003 Server for 64-bit extended systems, Solaris 9, or Solaris 10
- DB2 Universal Database 9.1 or 9.5 running on AIX 5L, Linux (Red Hat Enterprise), or Windows 2003 Server
- DB2 for IBM i running on IBM i V5R4 or V6R1

The database and the repository do not need to be installed on the same server, but some configuration information is necessary to ensure connectivity. During the installation, you will be prompted for the database server name, port number, username and password, and the name of the database to use for information storage and retrieval.

Important! With databases other than DB2 on IBM i , you must manually create the database prior to installation. Any valid database name can be used, but if a previously created database does not exist, the installation will not continue.

Note: For DB2 IBM i , DB2 XML Extender package must be enabled.

Database Permissions

The user must also have the following general permissions to the database to perform the install and initial startup of PASW Collaboration and Deployment Services:

- Create session
- Create table
- Drop table
- Create view
- Drop view
- Create function
- Create procedure
- Select

- Insert
- Update
- Delete
- Execute procedure

The exact names of these permissions vary depending on the database type. Also depending on the database, some additional permissions may be needed. For example, Oracle also requires an explicit CONNECT and CREATE INDEX permissions; MS SQL Server requires a REFERENCES permission.

Oracle Database Configuration

When using an Oracle 10g or 11g database in conjunction with PASW Collaboration and Deployment Services, the following parameters and configurations must be followed. Changes are made to the *init.ora* and *spfile.ora* parameter files.

Table 3-2
Oracle Database Parameters

Parameter	Setting
OPEN_CURSORS	150
NLS_CHARACTERSET	AL32UTF8
NLS_NCHAR_CHARACTERSET	AL16UTF16

Note: Both NLS_CHARACTERSET and NLS_NCHAR_CHARACTERSET should be set when creating the Oracle instance.

DB2 Configuration

When using a non-IBM i DB2 UDB database in conjunction with PASW Collaboration and Deployment Services, the default database creation parameters are not sufficient. The following parameters and configurations must be followed:

- Use a UTF-8 codeset.
- Use an 8K buffer pool (*SPSSEIGHT*).
- Include the two tablespaces, *SPSSEIGHT* and *SPSSLARGE*.
- Create a temporary system tablespace.

An example script for creating a database named *SPSSPLAT* follows:

```
CREATE DATABASE SPSSPLAT ON C: USING CODESET UTF-8 TERRITORY US COLLATE USING SYSTEM;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING APP_CTL_HEAP_SZ 128;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING APPGROUP_MEM_SZ      10715;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING APPLHEAPSZ 32000;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING CATALOGCACHE_SZ 32000;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING CHNGPGS_THRESH  60;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING DBHEAP       600;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING LOCKLIST      50;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING LOGBUFSZ      131;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING LOGFILSIZ     1024;
```

```
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING LOGPRIMARY     3;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING USEREXIT YES;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING LOGSECOND      -1;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING MAXAPPLS       100;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING MAXLOCKS       60;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING MINCOMMIT      1;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING NUM_IOCLEANERS  1;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING NUM_IOSERVERS   2;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING PCKCACHESZ      859;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING SOFTMAX        120;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING SORTHEAP       352;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING STMTHEAP       5000;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING DFT_DEGREE     1;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING DFT_PREFETCH_SZ 32;
UPDATE DATABASE CONFIGURATION FOR SPSSPLAT USING UTIL_HEAP_SZ    37991;
UPDATE DATABASE MANAGER CONFIGURATION USING SHEAPTHRES        14113;
UPDATE DATABASE MANAGER CONFIGURATION USING INTRA_PARALLEL     OFF;
UPDATE DATABASE MANAGER CONFIGURATION USING MAX_QUERYDEGREE    1;
UPDATE DATABASE MANAGER CONFIGURATION USING MAXAGENTS         200;
UPDATE DATABASE MANAGER CONFIGURATION USING NUM_POOLAGENTS     200;
UPDATE DATABASE MANAGER CONFIGURATION USING NUM_INITAGENTS     0;
UPDATE DATABASE MANAGER CONFIGURATION USING FCM_NUM_BUFFERS    1024;
UPDATE DATABASE MANAGER CONFIGURATION USING PRIV_MEM_THRESH       32767;
CONNECT TO SPSSPLAT;
ALTER BUFFERPOOL IBMDEFAULTBP SIZE 113974;
SET CURRENT QUERY OPTIMIZATION = 5;
COMMIT;
CONNECT RESET;
BACKUP DATABASE SPSSPLAT TO "C:\Temp" WITH 2 BUFFERS BUFFER
1024 PARALLELISM 1 WITHOUT PROMPTING;
CONNECT TO SPSSPLAT;
CREATE Bufferpool SPSSEIGHT IMMEDIATE  SIZE 250 PAGESIZE 8 K ;
CREATE Bufferpool SPSSTEMP IMMEDIATE  SIZE 250 PAGESIZE 32 K ;
CREATE  REGULAR  TABLESPACE SPSSEIGHT PAGESIZE 8 K  MANAGED BY SYSTEM
  USING ('C:\DB2\NODE0000\SPSSEIGHT' ) EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16
  TRANSFERRATE 0.14 BUFFERPOOL "SPSSEIGHT" DROPPED TABLE RECOVERY OFF;
COMMENT ON TABLESPACE SPSSEIGHT IS '';
CREATE  LARGE  TABLESPACE SPSSLARGE PAGESIZE 8 K  MANAGED BY DATABASE
  USING ( FILE 'C:\DB2\NODE0000\SPSSLARGE' 2560 ) EXTENTSIZE 16 OVERHEAD 10.5
  PREFETCHSIZE 16 TRANSFERRATE 0.14 BUFFERPOOL "SPSSEIGHT";
COMMENT ON TABLESPACE SPSSLARGE IS '';
CREATE SYSTEM TEMPORARY TABLESPACE SPSSTEMP PAGESIZE 32 K MANAGED BY AUTOMATIC STORAGE
EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE 0.14 BUFFERPOOL "SPSSTEMP";
COMMENT ON TABLESPACE SPSSTEMP IS '';
CONNECT RESET;
```

Microsoft SQL Server Configuration

Appropriate options must be used when setting up the database for processing non-Latin character sets. For example, it is recommended to use the Kana-sensitive (_KS) option to distinguish between Hiragana and Katakana Japanese characters. For more information about database collation, refer to Microsoft SQL Server documentation.

SPSS Inc. Products Compatibility

The system is compatible with the following versions of SPSS Inc. applications.

Table 3-3
Supported versions of SPSS Inc. applications

Application	Supported versions
PASW Modeler server	Version 13.0 or higher
PASW Modeler client	Version 13.0 or higher
PASW Statistics server	Version 17.0 or higher
PASW Statistics client	Version 17.0 or higher
ShowCase Suite	Version 8.0 or higher

PASW Statistics client, PASW Modeler client, and ShowCase Suite client are not required for use of PASW Collaboration and Deployment Services. However, these applications offer interfaces for using the repository to store and retrieve objects. The server versions of these products are required if jobs containing PASW Statistics syntax, PASW Modeler streams, or ShowCase files/sets will be executed.

Notes:

- PASW Collaboration and Deployment Services is backward-compatible with SPSS 16.0 and Clementine 12.0
- By default, the repository is installed without content repository adapter and process manager packages for PASW Modeler and PASW Modeler users must install the content repository adapter packages corresponding to their version of PASW Modeler. PASW Modeler 13.0 packages can be found on the PASW Modeler distribution disk and installed with PASW Collaboration and Deployment Services Package Manager utility. For more information, see Updating the Repository in Chapter 8 on p. 67.

Virtualization

PASW Collaboration and Deployment Services server or client components can be deployed into virtualized environments provided by third-party software. For example, in order to simplify deployment of a PASW Collaboration and Deployment Services development and testing environment, a system administrator can configure a virtual server on which to install the repository. The virtual machines hosting PASW Collaboration and Deployment Services components must meet minimum system requirements. For more information, see Provisioning the System on p. 7.

Supported Platforms

- The repository can be deployed into VMWare ESX Server.
- PASW Collaboration and Deployment Services clients can be deployed through Windows Terminal Services and Citrix Presentation Server.

Assuming that the configured virtualized environment meets the minimum system requirements, no performance degradation PASW Collaboration and Deployment Services server or client installations is expected. It is important to note, however, that virtualized systems might share available physical resources, and resource contention on systems with a heavy processing load can cause performance degradation of the hosted PASW Collaboration and Deployment Services installations.

Installing the Repository

Installation involves:

1. Copying the necessary files from the DVD to the target computer.
2. Deploying the repository into an application server for general use.

   This can be accomplished by using either the graphical installation wizard or a command line equivalent. Environments without a graphical interface must use the command line approach. When executing the Windows batch file or executable shell scripts provided on the installation DVD, the user installing the application must have permissions to install software under the operating system.

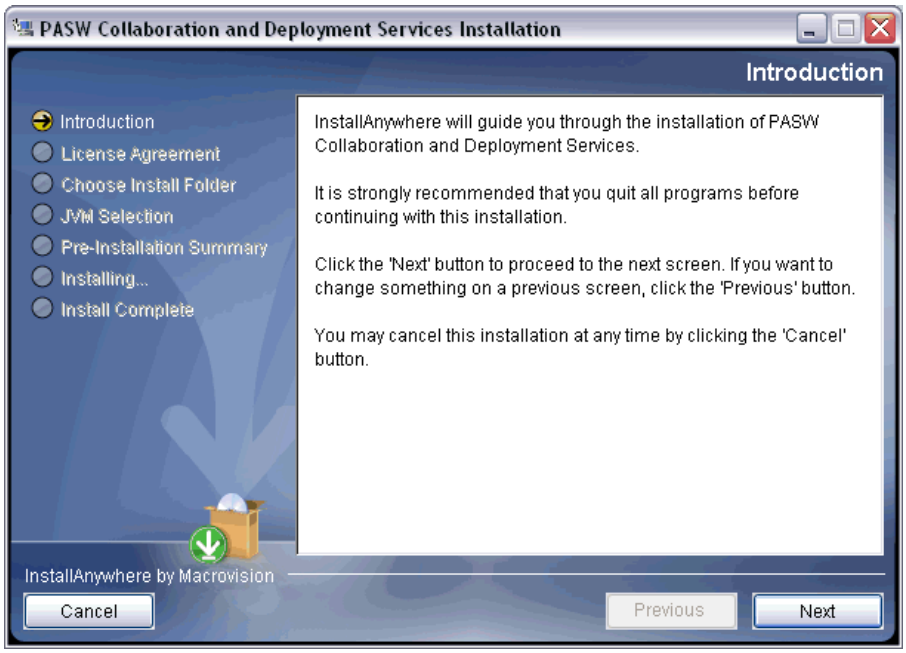
   Note: The installation progress is recorded in *<Installation directory>/setup/log/setup.log*. If you are installing the repository with BEA WebLogic application server, for security reasons this file must be deleted after you have verified that the installation completed successfully.

Graphical Installation Wizard

After verifying that a database server exists for the repository to connect to, execute the setup file associated with the operating system to start the installation wizard. The file is located in the */PASW/Disk1/InstData/<OS Name>* directory of the installation DVD.

install.exe

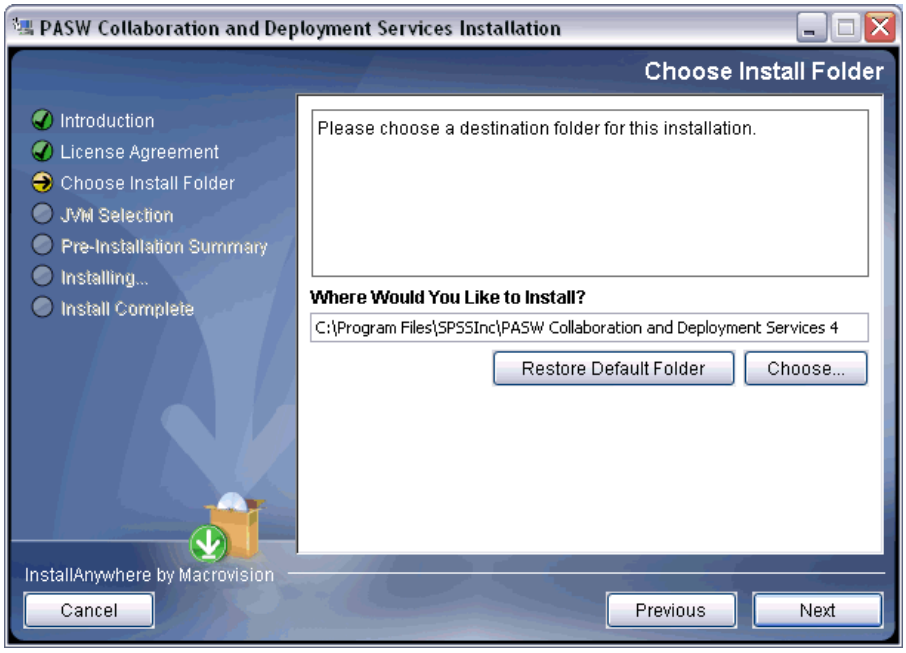
Figure 3-1
Installation Welcome



1. Click Next to begin the installation. The License Agreement screen appears.

2. Read the license agreement, select the Accept radio button, and click Next. The Choose Install Folder screen appears.

Figure 3-2
Select Directory

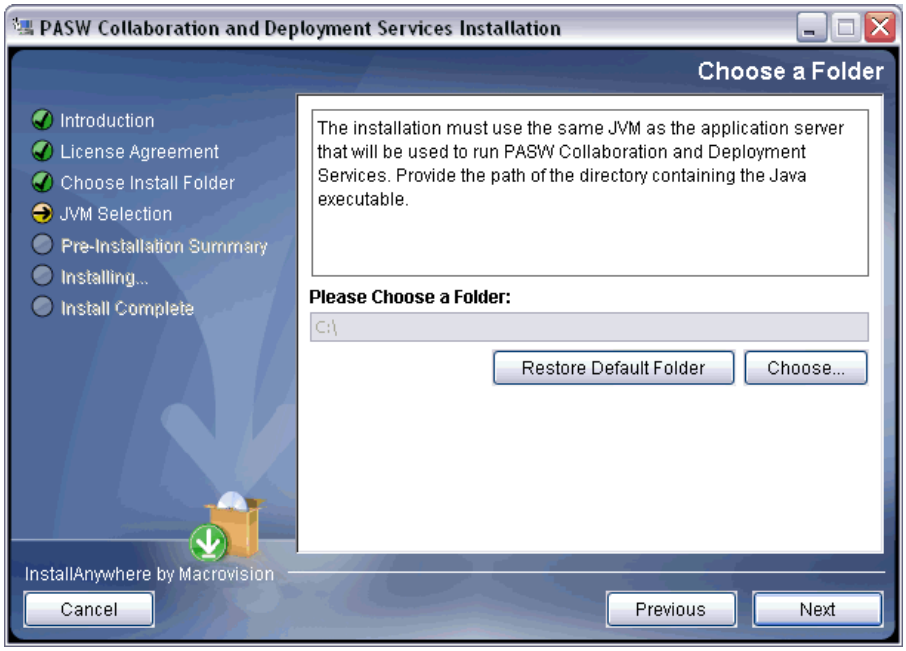


3. In the Directory Name field, type the path for the installation or click Browse and select the directory from the Open dialog. If installing the repository and the Deployment Manager on the same machine, use a different directory for each.

   Note: The path of the installation directory cannot contain extended ASCII characters.

4. Click Next to continue. The Choose a Folder screen appears.

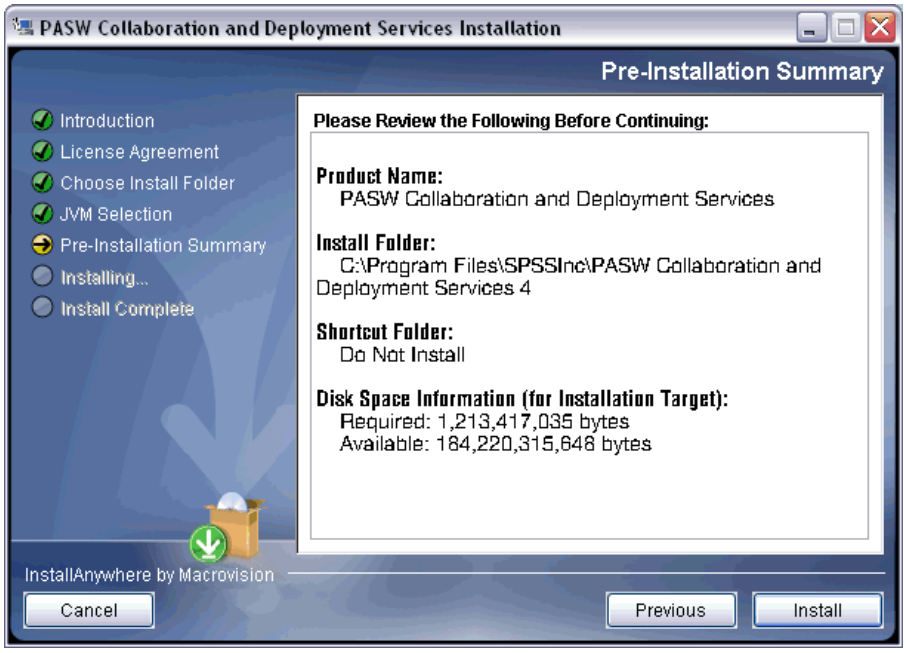
Figure 3-3
Select JVM location



5. Specify the path of the JVM to be used for running repository installation. The path must point to JVM used by the application server.

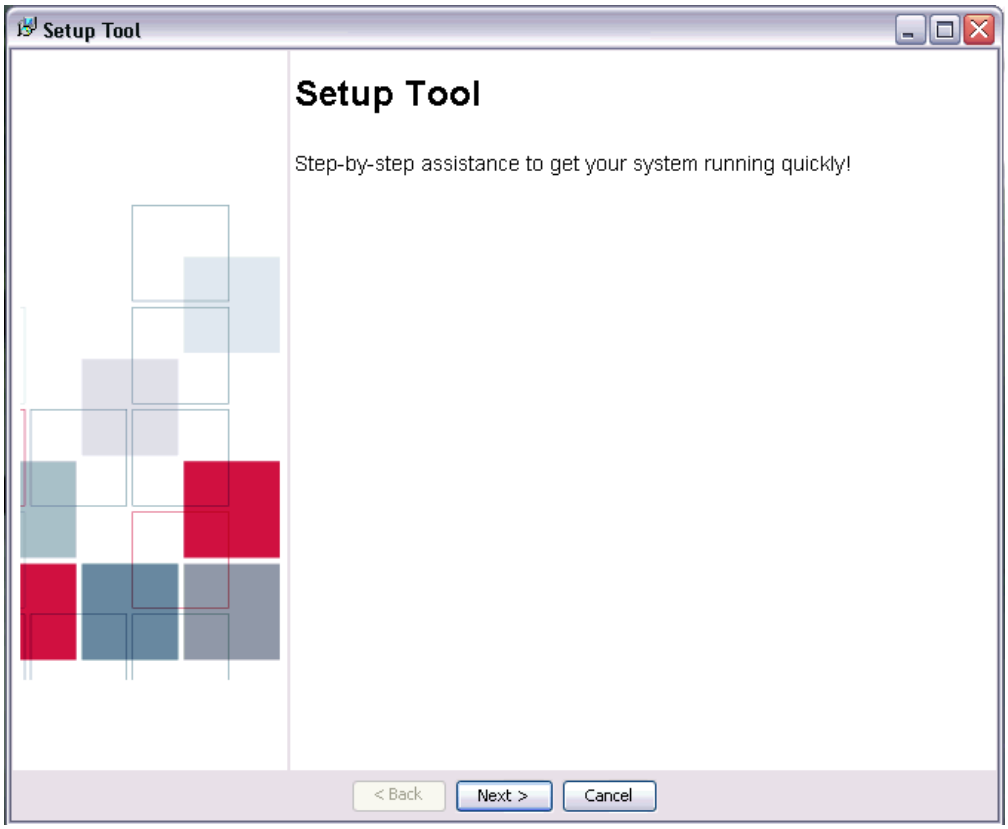
6. Click Next to continue. The Pre-Installation Summary screen appears.

Figure 3-4
Installation Summary



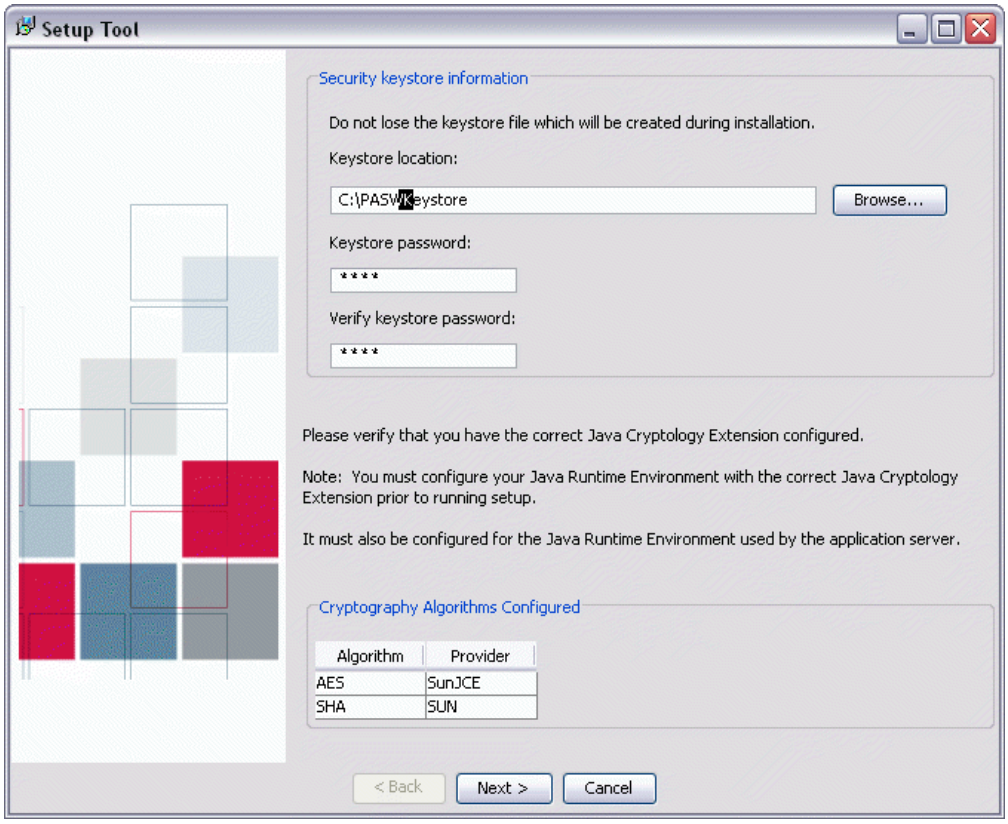
7. Verify the path is correct and adequate disk space exists. Click Back to return to the previous screens and modify information or click Next to launch PASW Collaboration and Deployment Services Setup Tool.

Figure 3-5
Setup Tool



8. To proceed with the setup, click Next. Security keystore information screen appears.

Figure 3-6
Specify keystore location and password and FIPS 140-2 compliance level



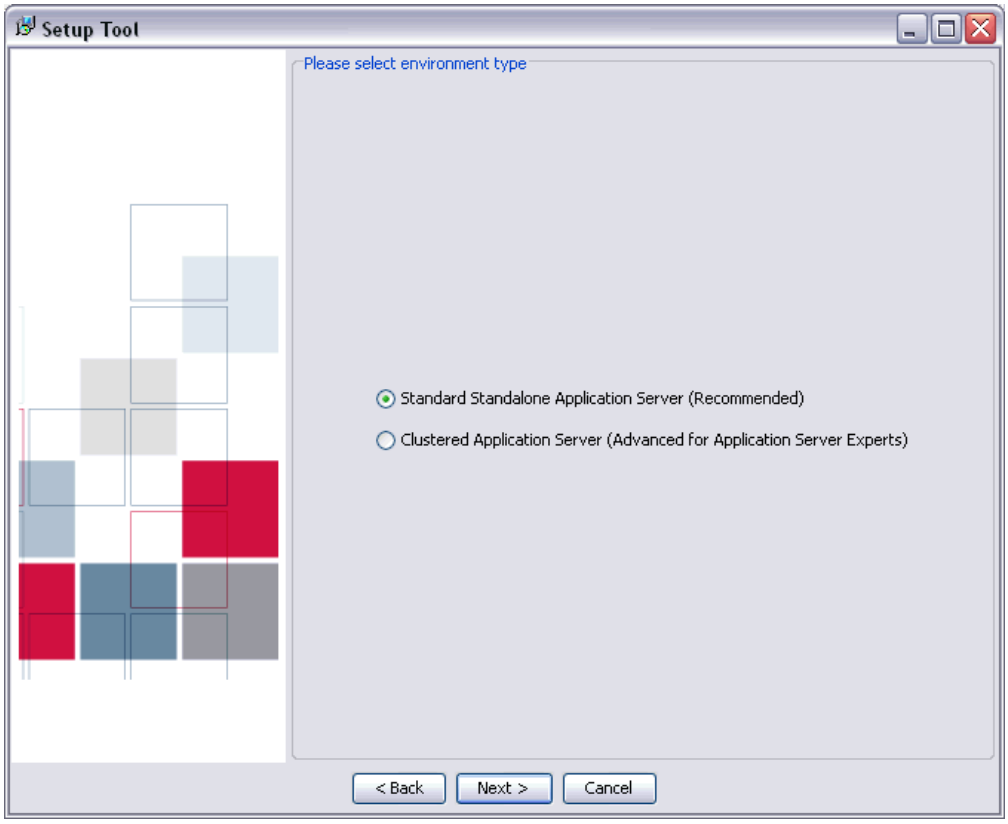
9. Specify the keystore location and then specify and confirm the password for accessing the keystore. The keystore is an encrypted file that contains the key for decrypting the passwords used by the repository, such as the repository administration password, the database access password, etc.

   Important! If the keystore file is lost, none of the passwords can be decrypted and the system becomes unusable and must be reinstalled. Therefore, it is recommended that backup copies of the keystore file be maintained.

   The available encryption algorithm will be listed in the table. If no algorithms are listed, you must exit the setup, configure the encryption modules for your Java runtime environment, and then restart the setup. For more information, see your JVM vendor documentation.

10. Click Next. Select Environment Type screen appears.

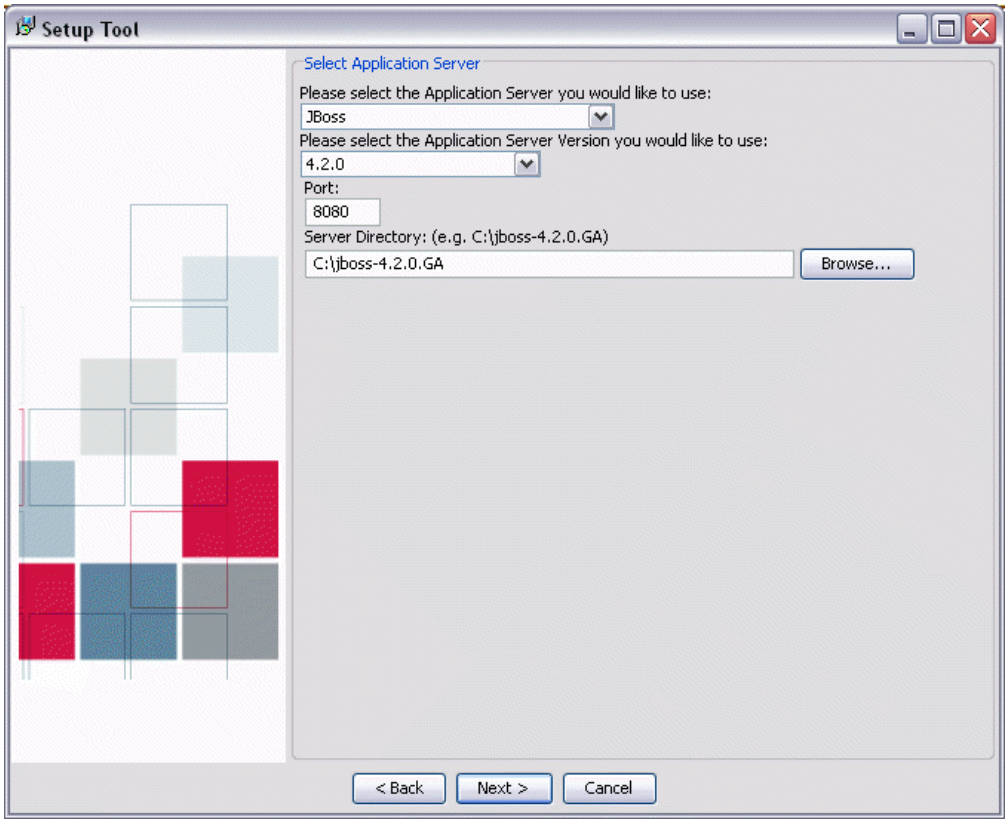
Figure 3-7
Select Environment Type



Select the environment type for PASW Collaboration and Deployment Services installation: standalone application server or application server cluster.

11. Click Next. Select Application Server screen appears.

Figure 3-8
Select Application Server



Select the application server into which to deploy the repository. Alternatively, select Manual to deploy the repository into the application server yourself. For this option, the installation creates an output directory in the specified location containing the files to be deployed and a *readme.txt* file containing instructions for manual deployment. Manual deployment should only be attempted by those with expertise in the application server being used.

If you are installing PASW Collaboration and Deployment Services into a an application server cluster, the following parameters must be specified:

- **Load Balancer URL.** The address of the load balancer. Users will be accessing PASW Collaboration and Deployment Services at this address.
- **Secure HTTP/SOAP Communication Between Components.** Specifies that communication between nodes in the cluster will be secure.

Important! Deploying PASW Collaboration and Deployment Services into an application server cluster includes a number of additional configuration steps. For more information, see Clustering in Chapter 4 on p. 53.

12. If you have chosen a standalone application server installation, specify configuration parameters for the application server. The parameters needed depend on the application server.

JBoss

- **Port.** The port number on which the application server runs.
- **Server Directory.** The installation location of the application server.

WebLogic

- **Port.** The port number on which the application server runs.
- **Server Directory.** The installation location of the application server.
- **Domain Location.** The directory location of the WebLogic domain.
- **Domain Name.** The name of the domain.
- **Server Name.** The name of WebLogic server.
- **Server Admin User ID.** Administrative login for the application server.
- **Server Admin Password.** Password associated with the specified application server administrative login.

Note: The repository can be installed into an existing domain, but both the domain and server must exist. If neither the domain nor the server exist, they will be created. However, if domain exists and the server does not, the installation cannot be completed.

WebSphere

- **Port.** The port number on which the application server runs.
- **Server Directory.** The installation location of the application server.
- **Server Admin User ID.** Administrative login for the application server.
- **Server Admin Password.** Password associated with the specified application server administrative login.
- **WebSphere SOAP Connector Address Port.** The port number used by WebSphere for incoming SOAP requests via HTTP.
- **Server Name** The name of the WebSphere server.
- **Node.** The name of the WebSphere node on which to install.
- **Cell.** The WebSphere cell containing the node.

Oracle 10G AS

- **Port.** The port number on which the application server runs.
- **Server Directory.** The installation location of the application server.
- **Server Admin User ID.** Administrative login for the application server.
- **Server Admin Password.** Password associated with the specified application server administrative login.
- **OC4J Instance.** The OC4J (Oracle Container for J2EE) instance to which the repository is deployed. Note that a new dedicated OC4J instance must be set up for each repository deployment.

- **OC4J Group.** The group association of the OC4J instance.
- **OPNM Request Port.** The port number for the OPMN (Oracle Process Management and Notifications) connection.

Note: The OPMN request port can be determined by running the $ORACLE_HOME/opmn/bin/opmnctl status -port command or looked up in the *$ORACLE_HOME/opmn/conf/opmn.xml* file.

NetWeaver

- **Port.** The port number on which the application server runs.
- **Server Directory.** The installation location of the application server.
- **Server Admin User ID.** Administrative login for the application server.
- **Server Admin Password.** Password associated with the specified application server administrative login.
- **SID.** System identifier of the SAP server.
- **Instance Number** System identifier of the SAP instance.
- **P4 Port.** The port number for P4 (RMI) access.

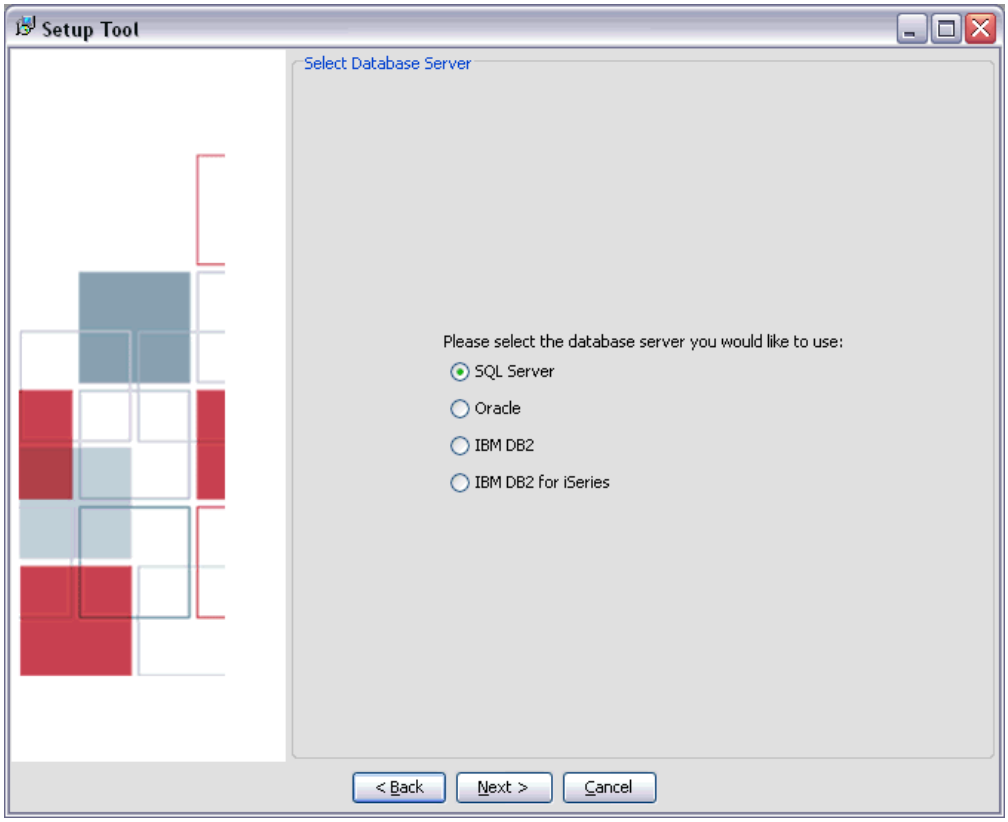
Manual (for expert J2EE server users)

- **Port.** The port number on which the application server runs.
- **Output Directory.** The directory where]PASW Collaboration and Deployment Services file will be installed. The location must be accessible to all servers in the cluster, for example as a mapped or mounted disk drive.

For more information about the parameters, consult application server vendor documentation.

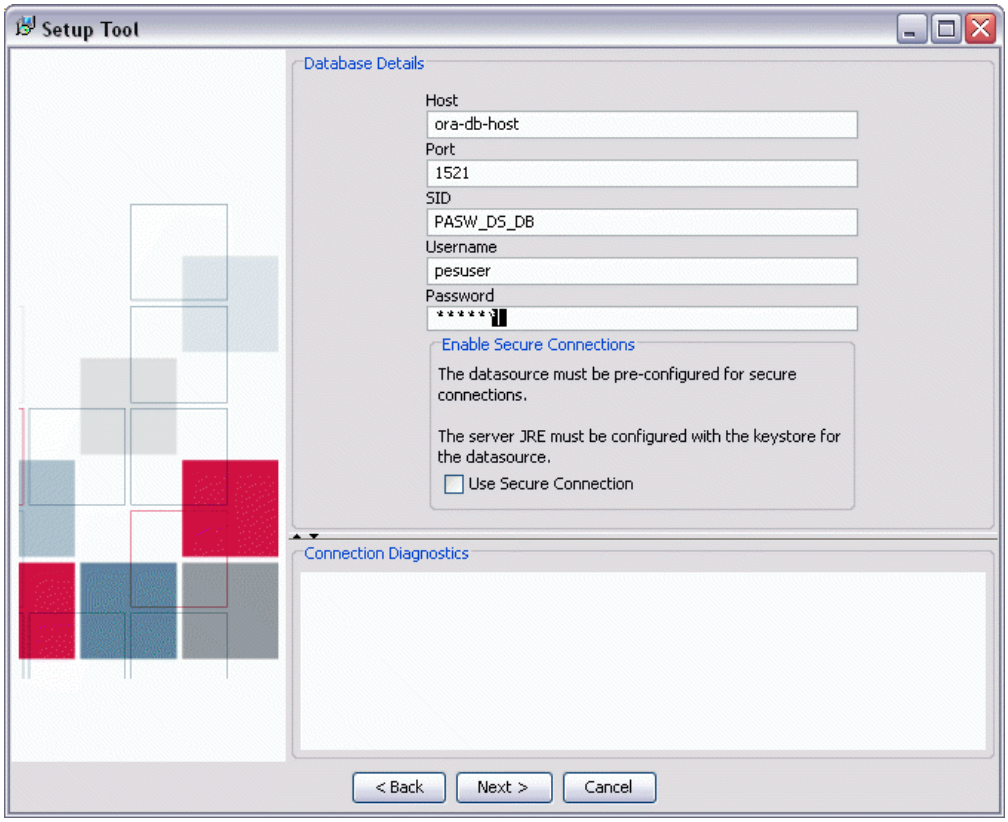
13. Click Next. The Select Database Server screen appears.

Figure 3-9
Select Database Server



14. Select the type of database used for the installation and click Next. The Database Details screen appears.

Figure 3-10
Database Details



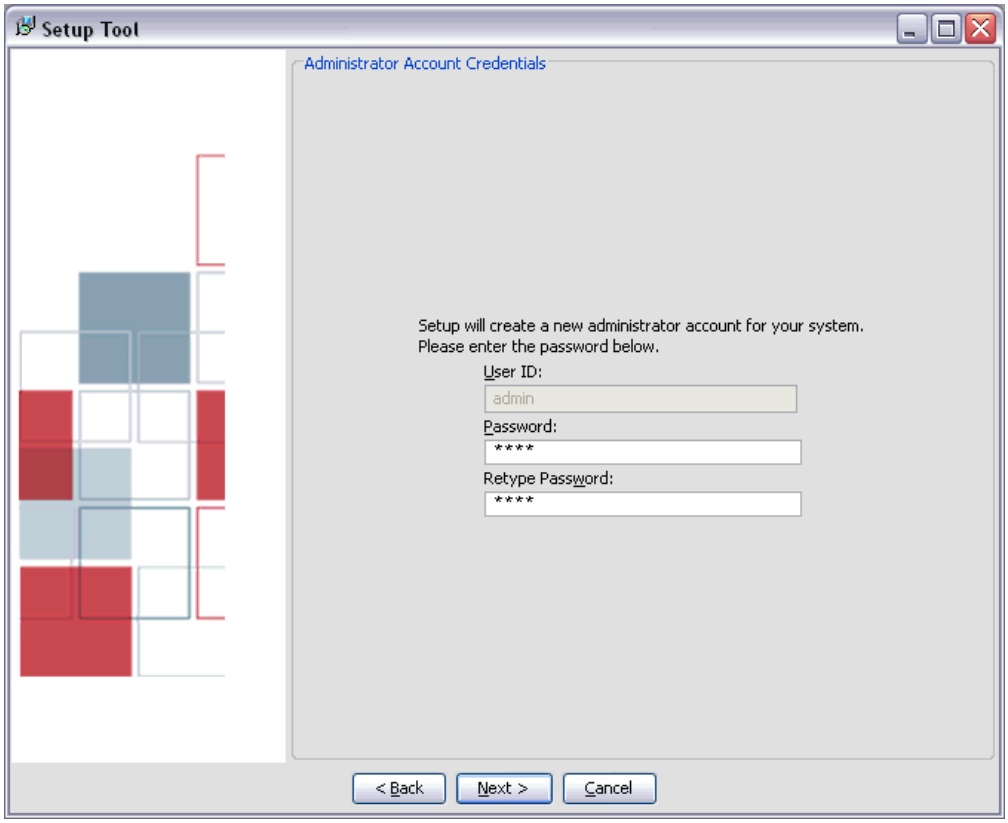
15. Supply the necessary parameters for connecting to the database. The parameters needed depend on the database and include:
    - **Host.** Host name or IP address of the database server.
    - **Port.** Port number on which the database server is running.
    - **Database/SID.** For databases other than DB2 on IBM i, name of an existing database to which to connect.
    - **Username.** Account used to connect to the database. This user must have rights to modify the selected database.
    - **Password.** Password associated with the username.
    - **Library.** For DB2 on IBM i, the name of the library collection to be used. If the library does not exist, it will be created.

16. Specify whether secure (SSL) database connections must be used.

    Note: To enable SSL connection to the database, the database must be pre-configured for SSL access. Consult vendor documentation for more information. Also, the application server JRE must have the certificates installed. For information on managing certificates, see *http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html*.

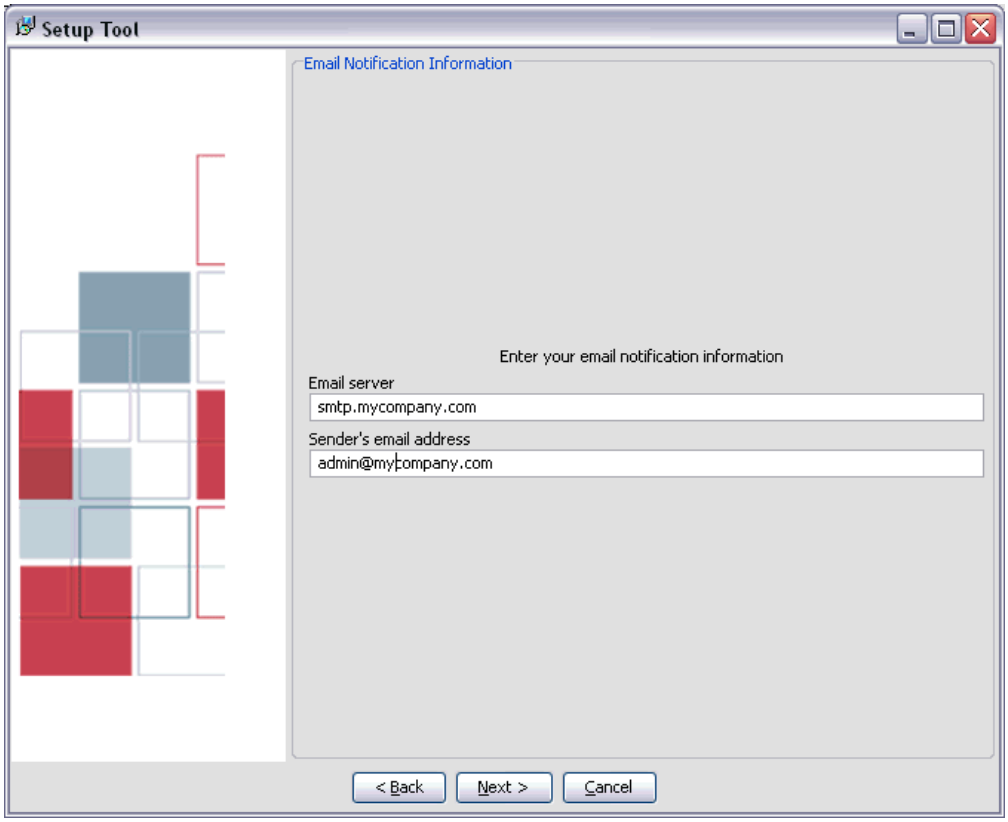
17. Verify the information entered is correct and click Next. The Administrator Account Credentials screen appears.

Figure 3-11
Administrator Account Credentials



18. Type a user ID and password for the administrator account and click Next. The administrator account is used when logging in for the first time; additional users are created after logging in to the system using this account. The E-mail Notification Information screen appears.

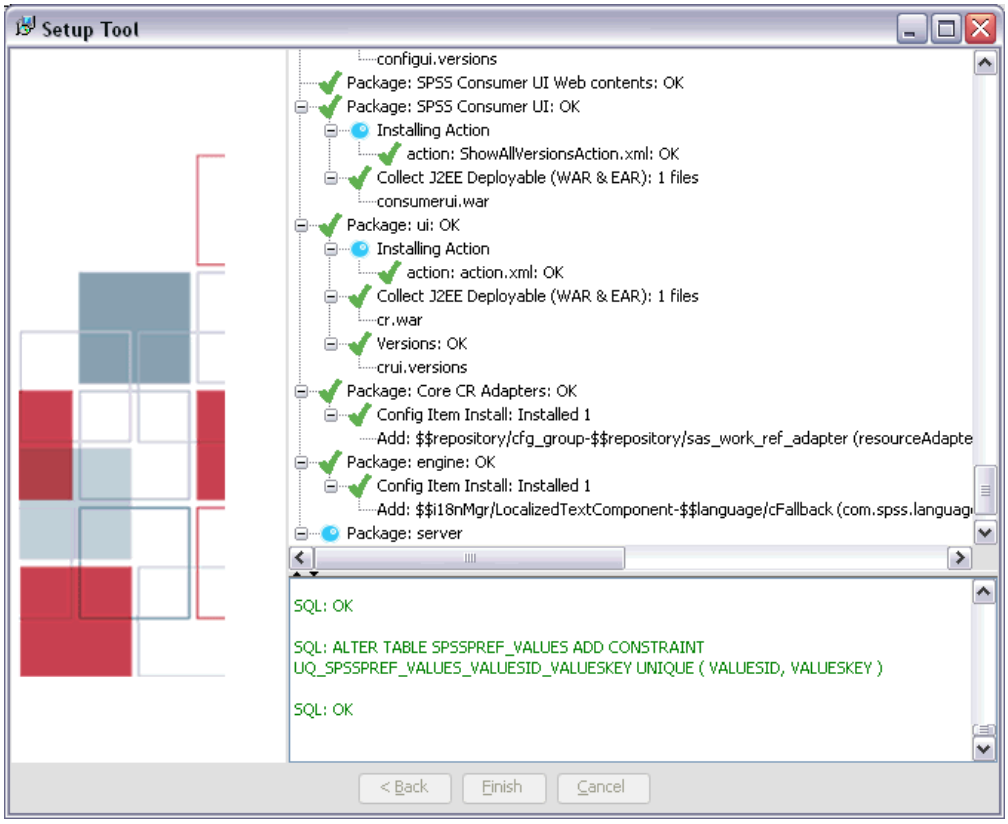
Figure 3-12
E-mail Notification Information



19. Type the name or IP address of the server used for outgoing e-mail and a valid address for the e-mail sender. Click Next. The Deployment screen appears.

20. Click Next to begin deploying the components. When complete, the Deployment Summary screen appears.

Figure 3-13
Deployment Summary



21. Review the deployment log. A green check mark icon indicates a component has been successfully deployed. In the event of a system or deployment error, a red X icon appears, along with a diagnostic report of the error.

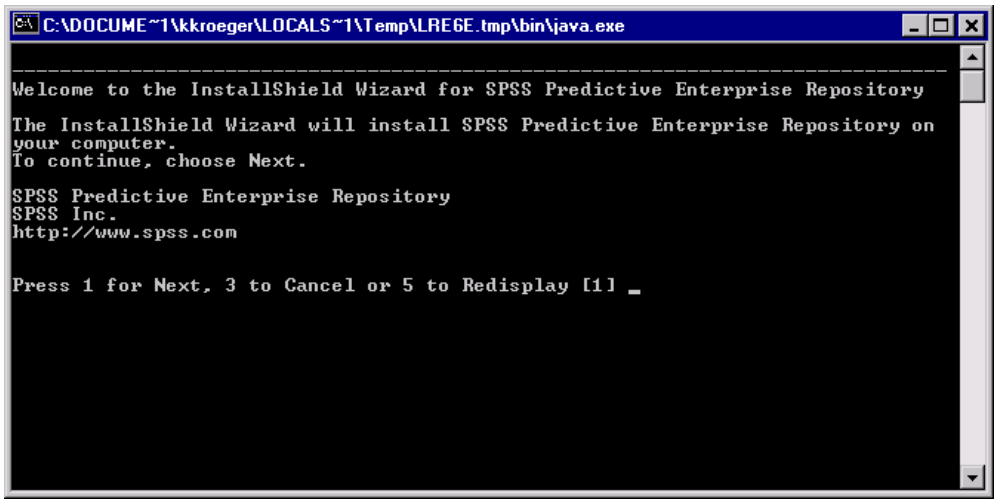
22. Click Finish to complete the installation.

Command Line Installation

After verifying that a database server exists for the repository to connect to, execute the operating system-specific program with the `console` command line switch to start the command line installation wizard. The program file is located in the */PASW/Disk1/InstData/<OS Name>/NoVM* directory of the installation DVD.

```
insall.exe -i console
```

Figure 3-14
Command Line Installation



Command line setup prompts for the same information as the graphical installation wizard. For more information, see Graphical Installation Wizard on p. 14. Many fields have default values shown in square brackets. Pressing Enter will accept the default value. Although passwords are echoed on-screen as typed, they are saved in encrypted form. At any time, typing \restart and pressing Enter (or Return) will return to the initial installation screen. Information reflecting installation processing is available in the logs written to *./setup/log* folder of the installation directory.

Following the installation, you must license the PASW Collaboration and Deployment Services by running the command line license tool. For more information, see Licensing Your Product on p. 29.

Licensing Your Product

Note: PASW Collaboration and Deployment Services is installed with an automatic 30-day temporary license. PASW Collaboration and Deployment Services will no longer run after the 30-day period unless a permanent license is installed.

After installing PASW Collaboration and Deployment Services, license the product from the command prompt.

Note: Licenses are tied to your computer's hardware with a **lock code**. If you replace your computer or its hardware, you will have a new lock code and will need to repeat the authorization process. If you find out that you exceeded the allowable number of authorizations specified in the license agreement, contact your sales representative.

Warning: The license is sensitive to time changes. If you must change the system time and then cannot run the product, contact SPSS Inc. Technical Support.

Using the License Authorization Wizard

- If you don't launch the License Authorization Wizard during installation or cancel the License Authorization Wizard before obtaining a license, you can launch it by choosing License Authorization Wizard in the Windows Start menu program group for PASW Collaboration and Deployment Services.
- When prompted, choose License my product now.
- When prompted, enter one or more authorization codes. Authorization codes are delivered (on a separate sheet of paper) along with your software.

  The License Authorization Wizard sends your authorization code over the Internet to SPSS Inc. and automatically retrieves your license. If your computer is behind a proxy, click Configure proxy settings and enter the appropriate settings.

  If the authorization process fails, you will be prompted to send an e-mail message. Choose whether you want to send the e-mail message through your desktop e-mail program or through a Web-based e-mail application.

  - If you choose the desktop option, a new message with the appropriate information will be created automatically.
  - If you choose the Web-based option, you must first create a new message in your Web-based e-mail program. Then copy the message text from the License Authorization Wizard and paste it into your e-mail application.

  Send the e-mail message and respond to the prompt in the License Authorization Wizard. The e-mail message will be processed almost instantaneously. You can click Enter License Code(s) to enter any license code(s) that you receive. If you already closed the License Authorization Wizard, restart it and select License my product now. On the Enter Codes panel, add the license code that you received and click Next to complete the process.

Installing a License from the Command Prompt

You have two options for installing from the command prompt. You can use *licenseactivator* to get a license from the Internet automatically, or you can use *echoid* to get a license manually.

Using licenseactivator to Install a License Automatically

The computer on which you are installing the license must be connected to the Internet. If it isn't, install the license manually. Instructions for installing the license manually follow these for using *licenseactivator*.

- Log in as the user who installed PASW Collaboration and Deployment Services.
- Open a command prompt and change directories to the PASW Collaboration and Deployment Services installation directory.
- You typically have an authorization code. In the most simple case, you enter the following at the command prompt. See below for more details about the command prompt usage.

licenseactivator <auth-code>

where <auth-code> is your authorization code.

You should see a message that the license was added successfully. If it wasn't, note the error code and try installing the license manually.

When you use *licenseactivator*, it licenses the product and writes a log file to its directory. The name of the log file is *licenseactivator_<month>_<day>_<year>.log*. If any errors occur, you can check the log file for more information. This information is also useful if you contact SPSS Inc. for support.

Using licenseactivator with authorization codes. *licenseactivator* is typically used with one or more authorization codes that you received when you purchased the product.

```
licenseactivator authcode1[:authcode2:...:authcodeN] [proxy-userid] [proxy-password]
```

- Multiple authorization codes are separated by colons (:).
- The proxy user ID and password are optional, but you may need them if you are using a proxy server. The proxy settings work only when the Local Area Network (LAN) settings in the Internet Settings control panel reference a specific proxy server address and port.

Using licenseactivator with license codes. In less common scenarios, SPSS Inc. may have sent you a *license*.

```
licenseactivator licensecode[:licensecode2:...:licensecodeN]
```

- Multiple license codes are separated by colons (:).
- When using license codes, *licenseactivator* does not connect to the Internet, so you do not need to specify proxy information.

Installing a License Manually

1. Log in as the user who installed PASW Collaboration and Deployment Services.
2. Open a command prompt and change directories to the PASW Collaboration and Deployment Services installation directory.
3. Get the lock code for the server machine. At the command prompt, type echoid (Windows) or ./echoid (UNIX/IBM i).
4. Send the lock code and your authorization code to SPSS Inc. by calling your local office or sending an e-mail message to *service@spss.com*. SPSS Inc. will then provide a license code or a file containing a license code.
5. Use *licenseactivator* to enter the license code or codes.

Viewing Your License

You can view the license by running *showlic.exe*, which is located in the PASW Collaboration and Deployment Services installation directory. When you run the file, you can see the license expiration date.

Changing the Database Password

For security reasons, it may be necessary to change the database password following repository installation. In such cases the password used by the repository for database access must also be changed. PASW Collaboration and Deployment Services provides a utility for changing the database password which can be used in GUI or command line mode.

Note: If WebLogic application server is used with the repository, the password must be changed in PASW Collaboration and Deployment Services before it is changed in the database.

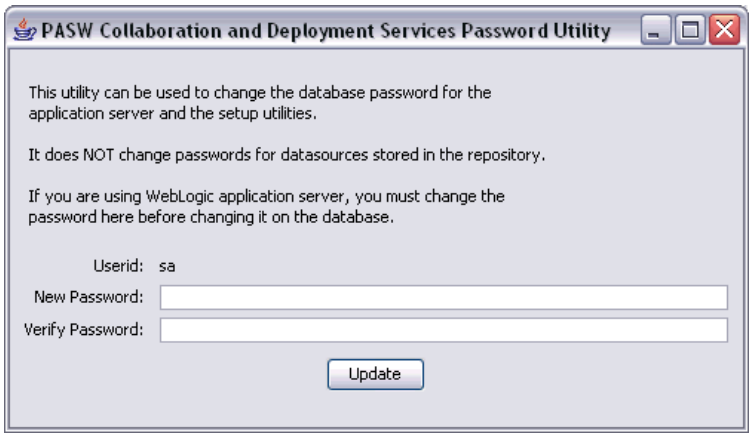
To run the password change utility in GUI mode:

1. Execute

   <PASW Collaboration and Deployment Services Installation Directory>/setup/dbpassword.bat

   Password Utility dialog opens.

Figure 3-15
Password Utility



2. Specify and confirm the new password.
3. Click Update. The password used by the repository for database access is changed.

To run the password change utility in command line mode:

1. Execute

   <PASW Collaboration and Deployment Services Installation Directory>/setup/clidbpassword.bat

2. Specify and confirm the new password using the command prompt.

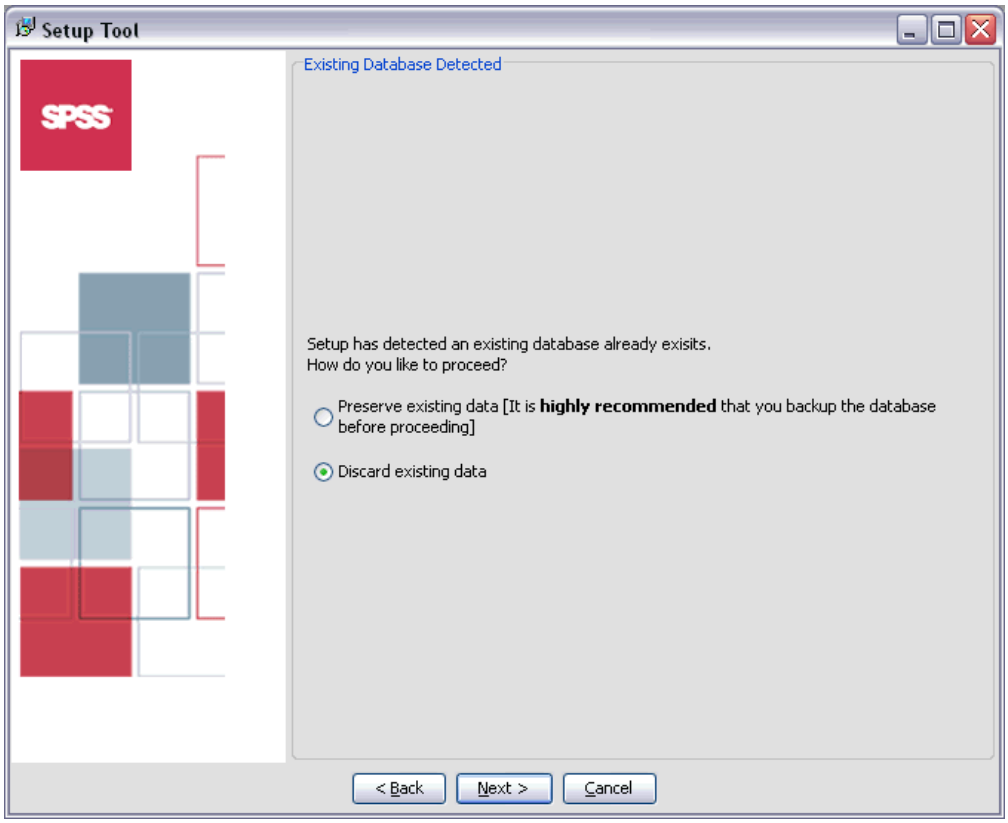
The password can also be changed by modifying the application server settings. Note that the password is stored in encrypted form, therefore the new password must be converted to an encrypted string by running *encrypt.bat* with the password as command line argument.

Upgrading Repository

Users with an existing version of the repository can conveniently upgrade their environment to take advantage of new features and functions. To upgrade to the current version:

1. Verify that hardware and software requirements are met and determine an installation directory for the application.
2. Reinstall the application server. The old instance of the application server cannot be used with the upgraded repository installation.
3. Install the latest version of the repository. It is recommended to use the already existing installation directory.
4. When prompted, specify the path of the application server.
5. When prompted, preserve the existing data in the existing database.

Figure 3-16
Existing Database



Uninstalling Repository

In the event that an installation becomes corrupt or the repository needs to be reinstalled due to system errors, the current version must be uninstalled.

Note: Back up the database before continuing. Uninstalling removes any tables it has created in the database. You will not be prompted to save this data.

To uninstall the repository:

1. Stop the repository.
2. Back up any data you wish to save in the repository. These tables are removed during the uninstallation process.
3. From the installation path, navigate to the *setup* directory.
4. On Windows, run *uninstall.bat*.
5. When prompted, confirm that the repository should be removed from the system. The uninstall script then undeploys services and deletes tables from the database.
6. When the script is complete, manually delete the root installation directory for the application.

JDBC Drivers

The reporting functionality of PASW Collaboration and Deployment Services is enabled by BIRT (Business Intelligence and Reporting Tools), an open-source package distributed by Eclipse Foundation under the Eclipse Public License. BIRT provides core reporting features, such as report layout, data access, and scripting. For more information about BIRT, see the BIRT project page (*http://www.eclipse.org/birt*). The PASW Collaboration and Deployment Services installation includes the BIRT reporting engine server components, which enable the execution of BIRT report syntax files as part of the PASW Collaboration and Deployment Services reporting job steps. PASW BIRT Report Designer is a standalone application that can be used in conjunction with PASW Collaboration and Deployment Services. It provides a rich user interface with a number of advanced features for creating reports and must be installed separately.

PASW BIRT Report Designer installation contains a set of SPSS Inc. JDBC drivers for all major database systems: Oracle, DB2, and SQL Server. These JDBC drivers are also installed by default with the repository. If a BIRT report uses a JDBC driver other than the ones installed by default, the driver must be installed in the repository. Depending on the application server, the directory location of the JDBC drivers is as follows:

JBoss. *<JBoss Installation Directory>/server/default/lib*

Oracle Application Server 10g R3.*<Oracle AS Installation Directory>/j2ee/<OC4J Instance>/shared-lib/com.spss.global/1.0*

WebLogic. *<Repository Installation Directory>/SPSSDomain/lib*

WebSphere. *<WebSphere Installation Directory>/lib/ext*

Enabling Web Installations from the Repository

In order to enable Web installations of PASW BIRT Report Designer and Enterprise View Driver, the following optional packages must be deployed into the repository:

- PASW BIRT Report Designer—*birtdesignerinstall.package*
- Enterprise View Driver—*pevdriverinstall.package*

Note: You must stop the repository before deploying client Web installation packages.

Installation involves:

1. Copying the necessary files from the distribution DVD to the target computer.
2. Deploying the repository into an application server for general use.

   This can be accomplished by using either the graphical installation wizard or the command line equivalent. Environments without a graphical interface must use the command line approach. When executing the Windows batch file or executable shell scripts provided on the installation DVD, the user installing the application must have permissions to install software under the operating system. The installation can also be launched in wizard or command line mode by executing the *SETUP.JAR* file in the *WEB* directory of the installation DVD, for example, `java -jar SETUP.JAR -console`. If for some reason the installation cannot be completed after the package files have been copied to the specified location, the packages can be deployed into the repository using Package Manager. For more information, see *PASW Collaboration and Deployment Services Administrator's Guide*. After the installation completes, the repository must be restarted.

   Note: If you are enabling Web installations in a repository running on WebSphere 6.1 application server on Windows 2003, it may be necessary to increase the value of `com.ibm.SOAP.requestTimeout` attribute in the *IBM\WebSphere\AppServer\profiles\<profile>\properties\soap.client.props* configuration file.

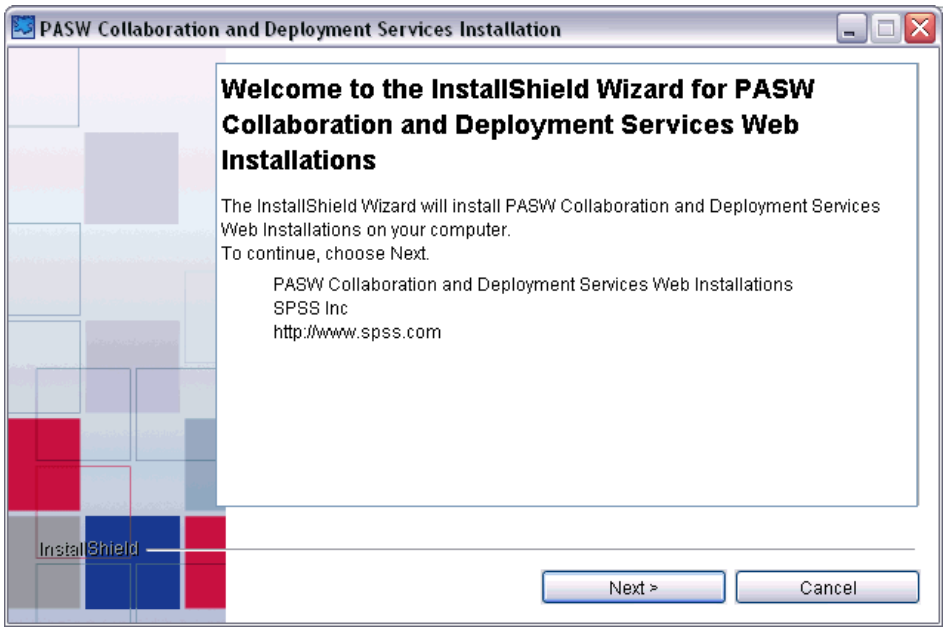
Graphical Installation Wizard

1. Verify that the repository has been stopped.
2. When the disk menu opens, click Install Enterprise Repository Web Apps or execute the setup program associated with the operating system to start the installation wizard in the */PASW/Web* directory of the DVD.

   ```
   setupwin32.exe
   ```

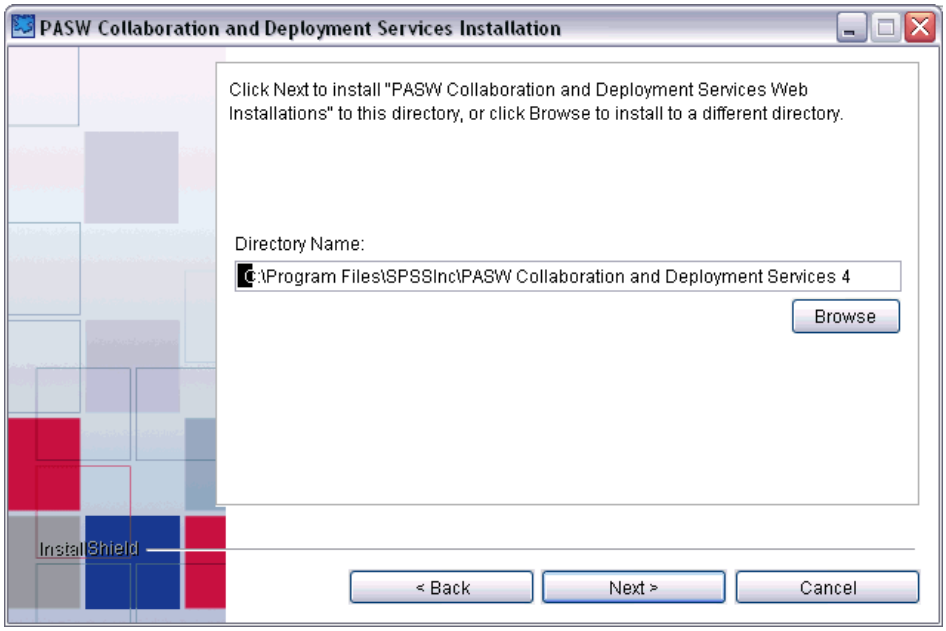
   The Welcome screen of the wizard appears.

Figure 3-17
Installation Welcome



3. Click Next to begin the installation. The Select Directory screen appears.

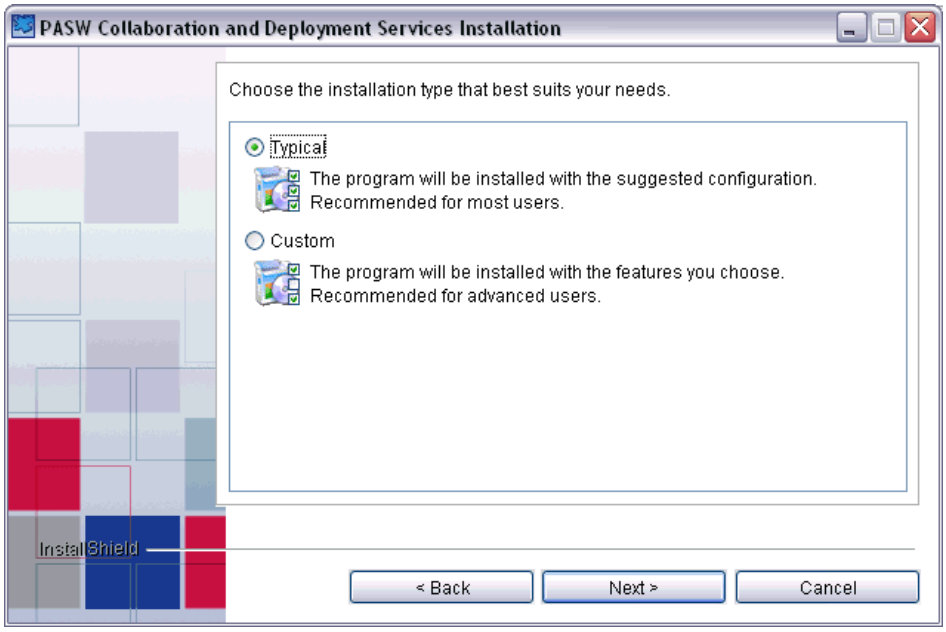
Figure 3-18
Select Directory



4. In the Directory Name field, type the path for the installation or click Browse and select the directory from the Open dialog.

5. Click Next to continue. The Installation Type Selection screen appears.

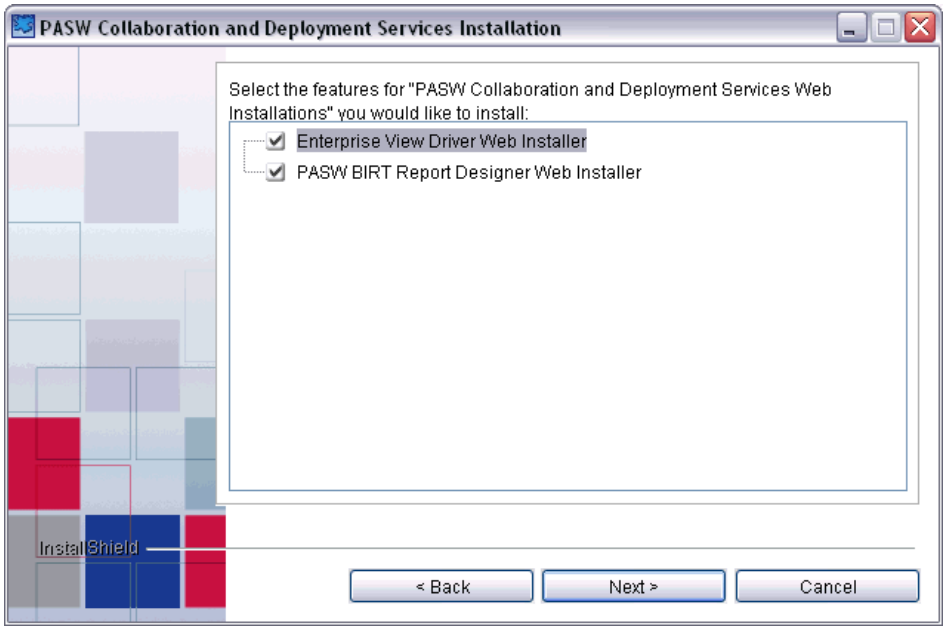
Figure 3-19
Installation Type Selection



6. Select Typical to install all components and click Next.

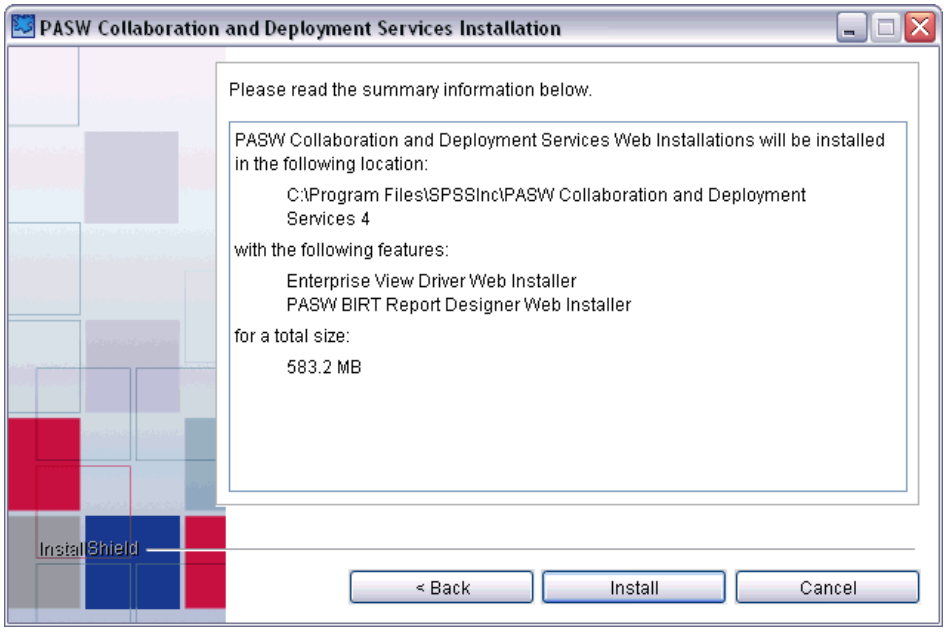
   Alternatively, select Custom to select individual components to be installed. The Component Selection screen appears.

Figure 3-20
Component Selection



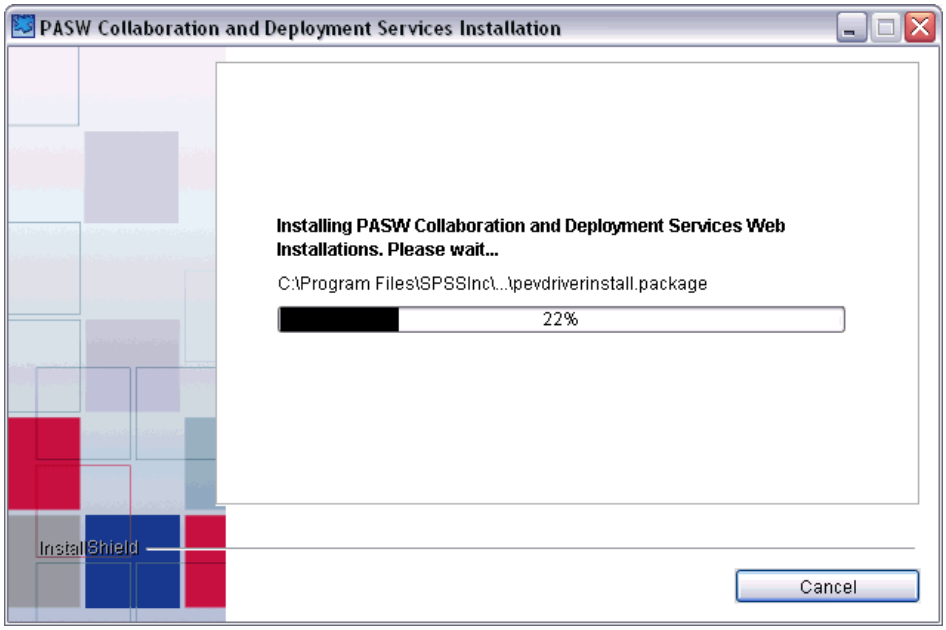
7. Click Next. The Installation Summary screen appears.

Figure 3-21
Installation Summary



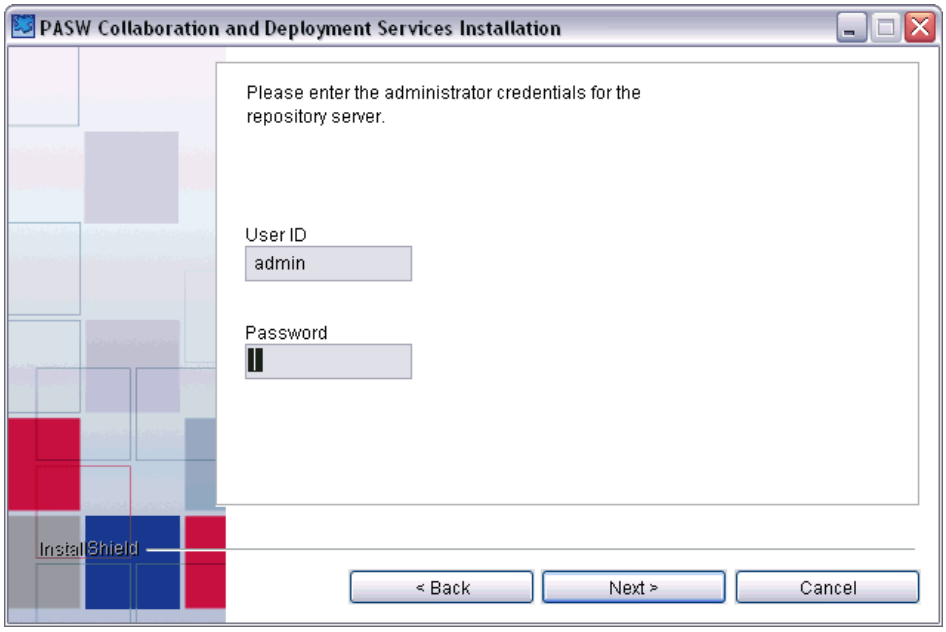
8. Verify the information entered is correct and click Install to continue. The Installation Progress screen appears.

Figure 3-22
Installation Progress



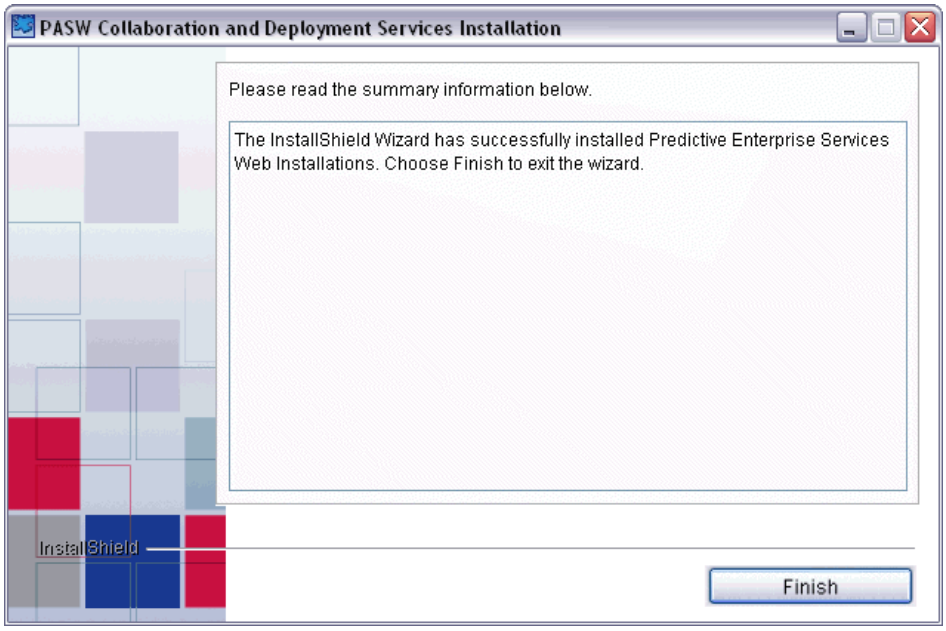
After the package files have been copied to the specified directory, they must be deployed into the repository. The PASW Collaboration and Deployment Services Credentials Entry screen is displayed.

Figure 3-23
Credentials Entry



9. Type the login name and password for an administrator account and click Next to begin deploying the components. When complete, the summary screen appears.

Figure 3-24
Deployment Completion



10. Click Finish to complete the installation.

Command Line Installation

Command line installation must be used on systems without a graphical interface. After verifying that a database server exists for the repository to connect to, execute the setup program associated with the operating system with the console command line switch:

```
setupwin32.exe -console
```

Command line installation prompts for the same information as the graphical installation wizard. Many items have default values, which are always shown in square brackets. Pressing Enter will accept the default value. Although passwords are echoed on-screen as typed, they are saved in encrypted form. At any time, typing \restart and pressing Enter (or Return) will return to the initial installation screen.

Installing Remote Process Server

In order to enable remote process execution in PASW Collaboration and Deployment Services, the remote process component must be deployed on the machine that is to be configured as a remote server. The installation involves:

1. Copying the necessary files from the distribution media to the target computer.
2. Configuring the remote process server.
3. Starting the remote process server.

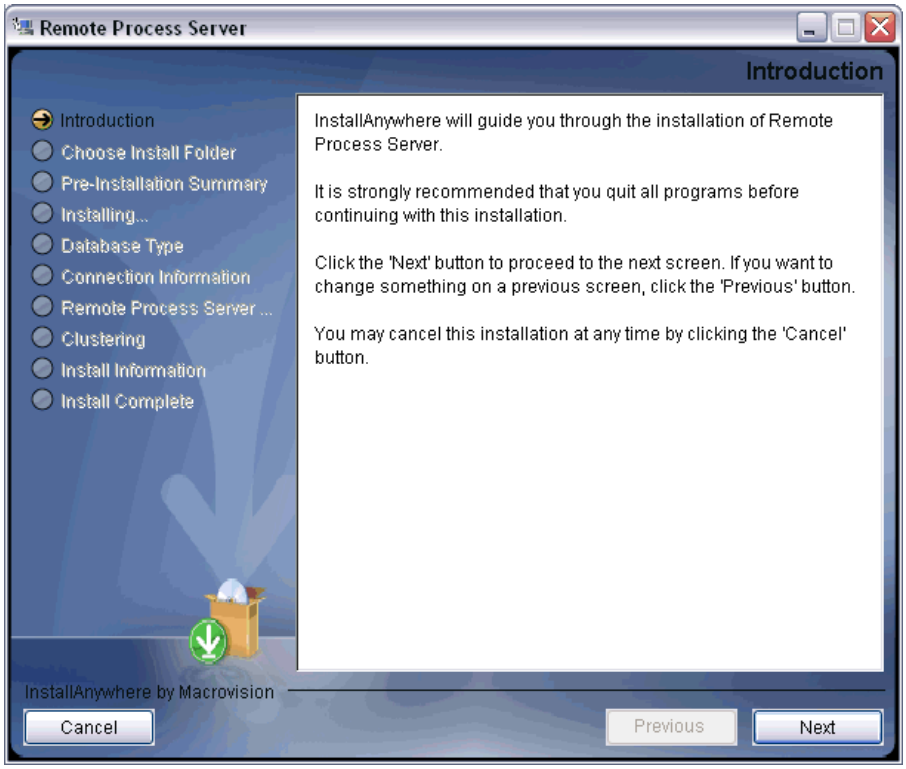
   This can be accomplished by using either the graphical installation wizard or the command line equivalent. Environments without a graphical interface must use the command line approach. When executing the Windows batch file or executable shell scripts provided on the installation media, the user installing the application must have permissions to install software under the operating system.

Graphical Installation Wizard

1. When the disk menu opens, click Install Remote Process Server, or execute the program to start the installation wizard in the */RPS/Disk1/InstData/<OS Name>* directory of the DVD. For Windows, this is *install.exe*. For Unix-based systems, the setup file is named *install.bin*.

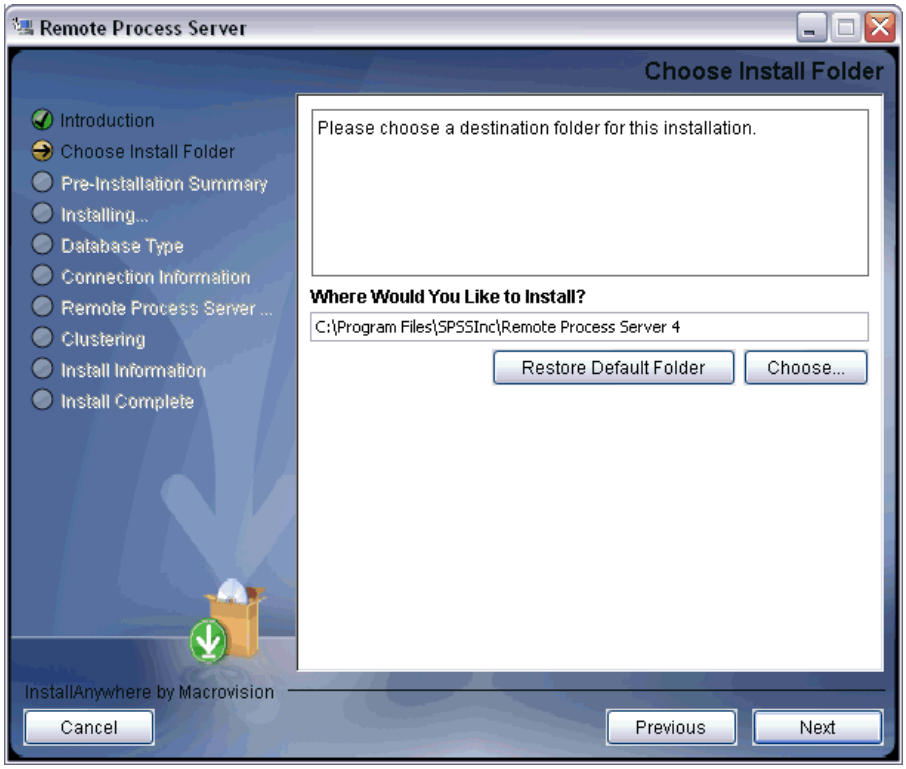
   The Introduction screen of the wizard appears.

Figure 3-25
Installation Welcome



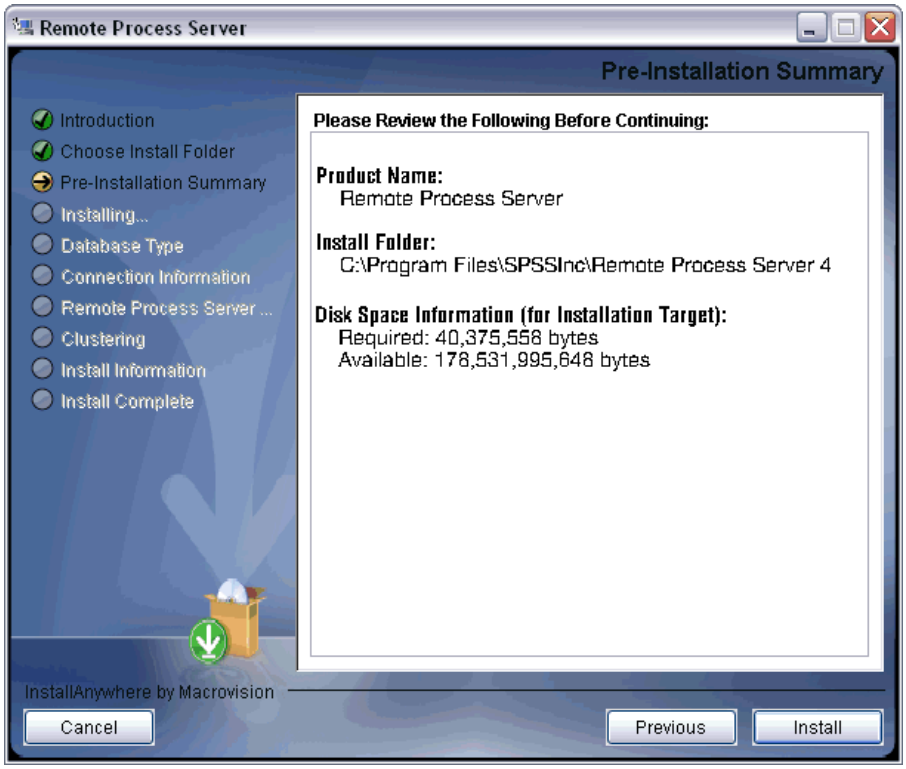
2. Click Next to begin the installation. The Choose Install Folder screen appears.

Figure 3-26
Choose Install Folder



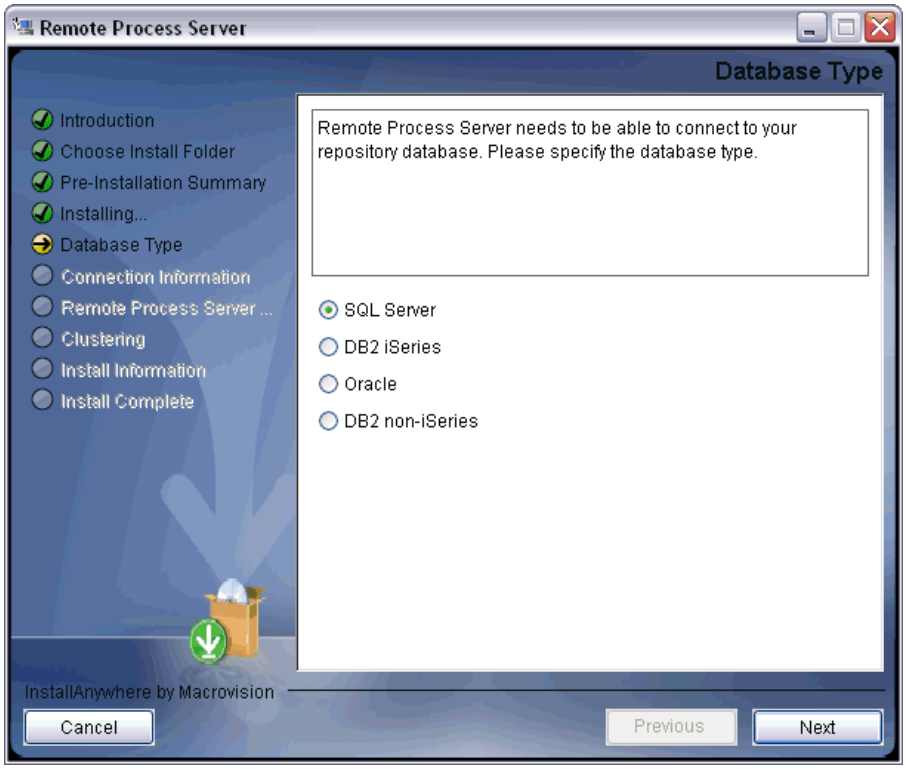
3. Type the path for the installation or click Browse to navigate to the desired folder. Click Next to continue. The *Pre-Installation Summary* screen appears.

Figure 3-27
Installation Summary



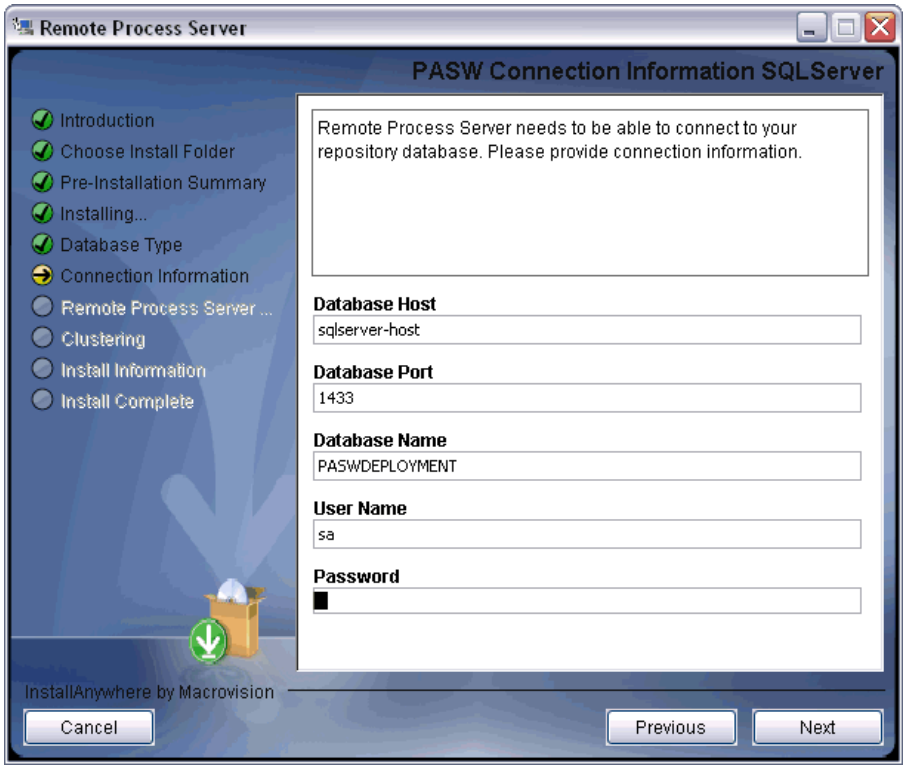
4. Click Install to continue. The remote process server component is copied to the specified directory on the system. After the component has been copied, the repository database connection information must be specified. The *Database Type* screen appears.

Figure 3-28
Database type selection



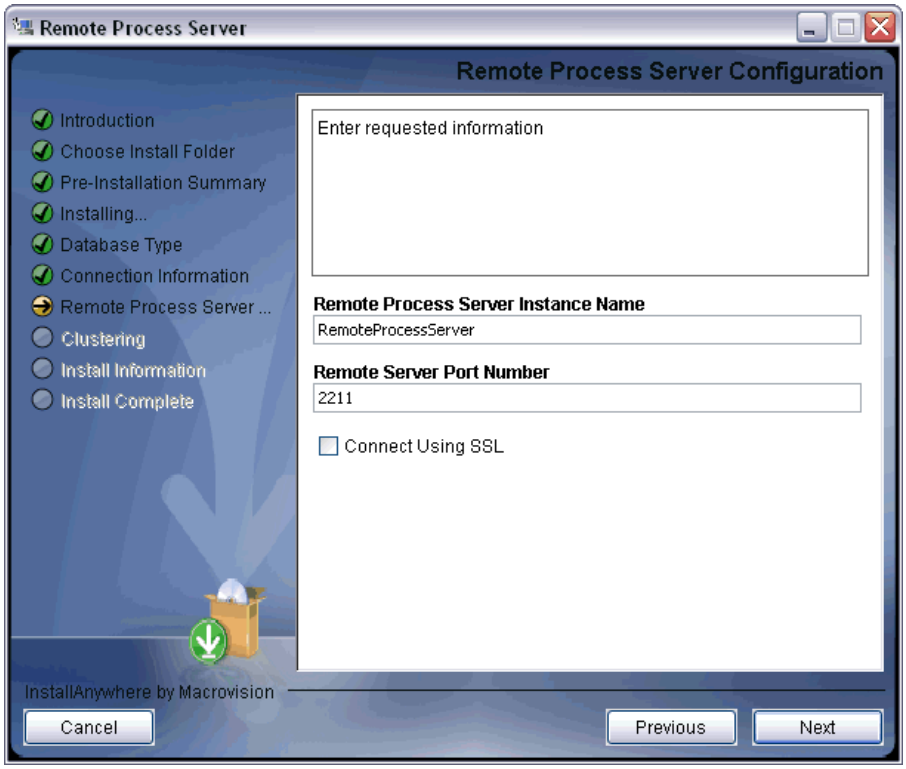
5. Select the database type and click Next to continue. The *Connection Information* screen appears.

Figure 3-29
Database connection information



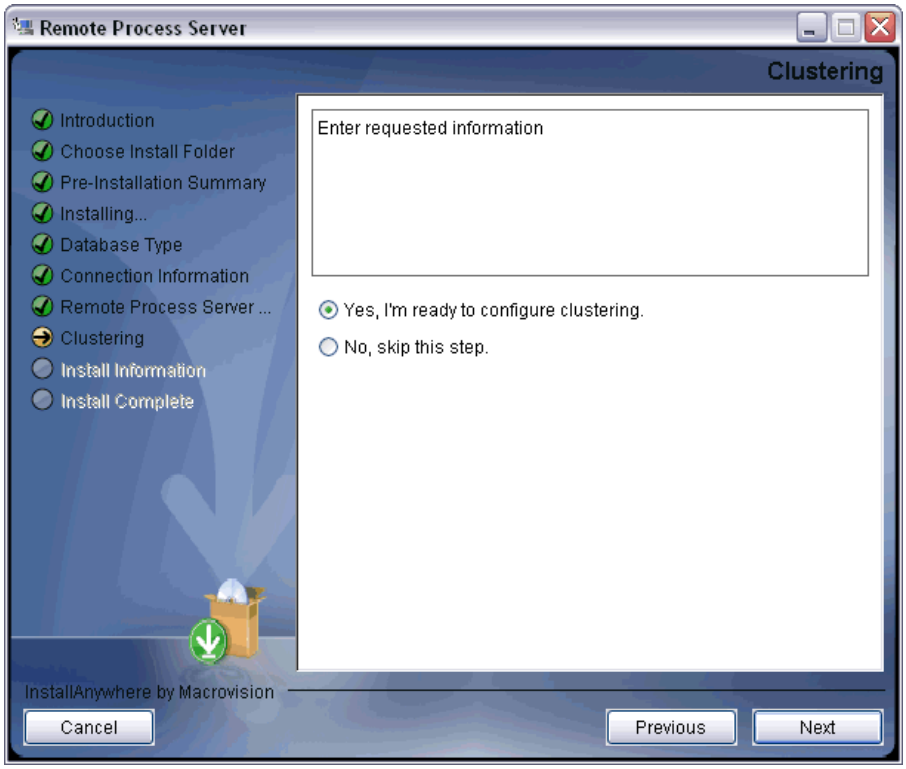
6. Specify the database host, database name, user name, and password. Click Next. The *Remote Process Server Configuration* screen appears.

Figure 3-30
Remote server configuration



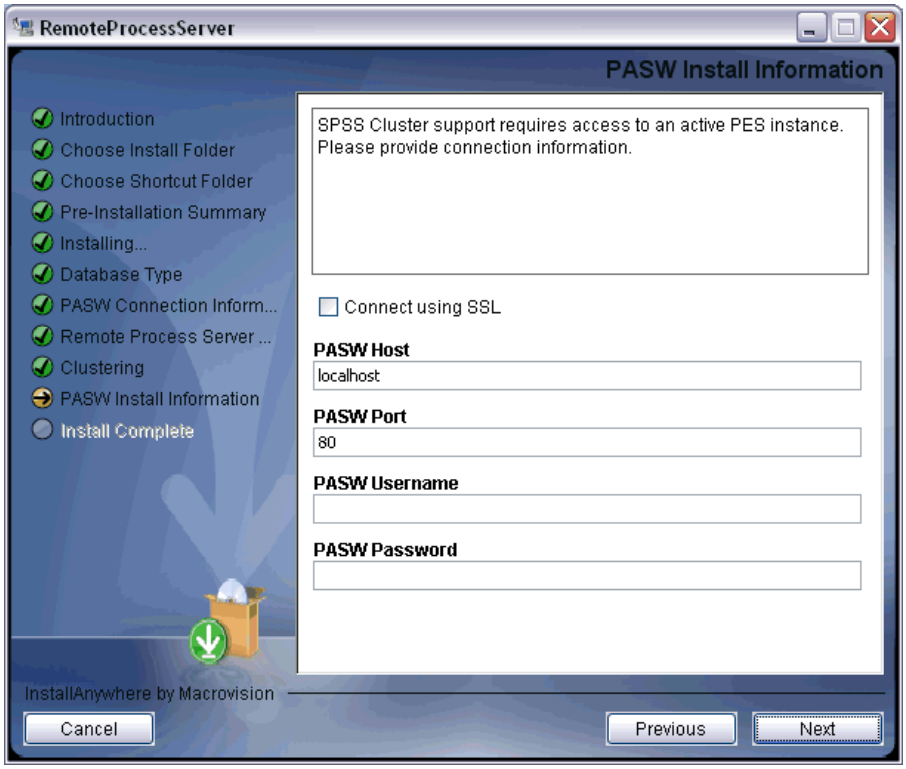
7. Specify the remote processing server name, access port, and whether a secure connection is to be used. Click Next. The *Clustering* screen appears.

Figure 3-31
Remote server clustering



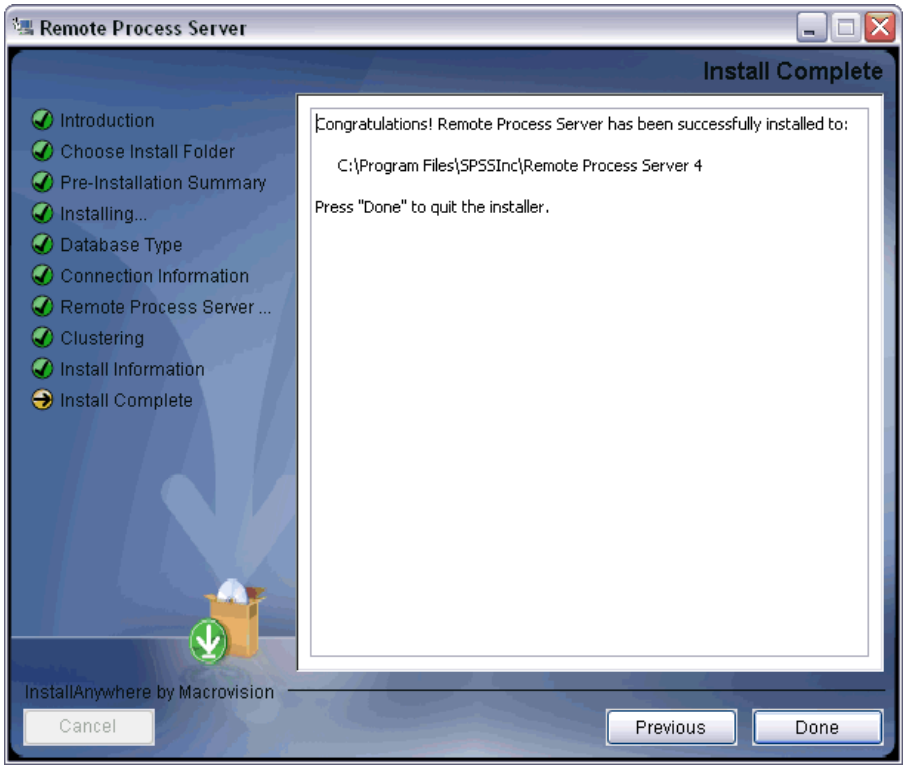
8. Specify if you want to enable clustering for your remote server. If clustering of the server is enabled for a specific instance of a repository, it will be possible to include the remote server in a cluster defined in that repository. If you choose not to enable clustering, the installation will proceed to completion after you click Next. Otherwise, the repository installation information screen is displayed.

Figure 3-32
Repository information



9. Specify the host, port, and login credentials of the repository for which clustering is to be enabled. Click Next. The installation completion screen appears.

Figure 3-33
Installation completion



10. Click Done to complete the installation.

Command Line Installation

Command line installation must be used on systems without a graphical interface. After verifying that a database server exists for the repository to connect to, execute the program in the */RPS/Disk1/InstData/<OS Name>* directory of the DVD with the console command line switch.

- On Windows:

```
install.exe -console
```

- On UNIX:

```
./install.bin -console
```

- On IBM i, in QShell environment copy *setupi5.sh* script and the installation JAR files to a temporary directory and then run setup using commands similar to the following:

```
cp /qopt//OPT_DVD/RPS/setupi5.sh /temp
cp /qopt//OPT_CD/RPS/*.jar /temp
cp /qopt//OPT_CD/RPS/SETUP.JAR /temp
/temp/setupi5.sh
```

Note: Remote process server installation on IBM i requires classic JVM 1.5 to be enabled.

Command line installation prompts for the same information as the graphical installation wizard. Many items have default values, which are always shown in square brackets. Pressing Enter will accept the default value. Although passwords are echoed on-screen as typed, they are saved in encrypted form.

Starting and Stopping Remote Process Server

After the remote process server has been installed on the target host system, it must be started.

- To start the server, execute the following command:

  (Windows)
  <Remote Process Server Installation directory>/startserver

  (UNIX and IBM i)
  <Remote Process Server Installation directory>/startserver.sh

- To enable remote process server over a secure connection additional parameters must be specified:

  (Windows)
  <Remote Process Server Installation directory>/startserver "-Djavax.net.ssl.keyStore=./keystore" "-Djavax.net.ssl.keyStorePassword=remote"

  (UNIX and IBM i)
  <Remote Process Server Installation directory>/startserver.sh "-Djavax.net.ssl.keyStore=./keystore" "-Djavax.net.ssl.keyStorePassword=remote"

- To stop remote process server, execute the following command:

  (Windows)
  <Remote Process Server Installation directory>/shutdown

  (UNIX and IBM i)
  <Remote Process Server Installation directory>/shutdown.sh

Installing PASW Collaboration and Deployment Services Scripting

PASW Collaboration and Deployment Services provides a scripting framework with a set of Content Repository and Process Management APIs that advanced users and administrators can use to write independent routines or batch jobs that combine a set of routines. This can greatly simplify bulk tasks such as changing security permissions for a large group of users, labeling or removing a label from a large number of folders/files, or uploading/downloading a large number of folders/files. The framework includes the ability to perform tasks from the command line, as well as a rich API for interacting with PASW Collaboration and Deployment Services within your own Python code.

For general information about Python, a dynamic object-oriented programming language, see the Python site (*http://www.python.org*).

To install PASW Collaboration and Deployment Services Scripting on a Windows system:

1. If Python is already installed on your system, uninstall it.
2. Insert the installation media.
3. Open the *PYTHON\Disk1\InstData\NoVM* directory and double-click *install.exe*. Follow the screen instructions to complete the installation. Install to the default location. This installs the required Python, ZSI, and PyXML technologies.
4. Open the *PYTHON* directory on the installation media and extract the contents of *cads-scripting-1.0.zip* to a temporary directory.
5. Add the Python scripting location to your PC's **Path** system environment variable.
6. At a command prompt, change the current directory to the folder where you extracted *cads-scripting-1.0.zip*. Type the following command and press Enter.

```
python setup.py install
```

To install PASW Collaboration and Deployment Services Scripting on a UNIX system:

1. If Python 2.4.3, ZSI 2.0 rc3, and PyXML 0.8.4 are not already installed on your system, install after downloading from their respective web sites before proceeding to step 2.
   - Python 2.4.3: *http://www.python.org/download/releases/2.4.3/*
   - ZSI 2.0 rc3: *http://sourceforge.net/projects/pywebsvcs*
   - PyXML 0.8.4: *http://sourceforge.net/project/showfiles.php?group_id=6473*
2. Insert the installation media.
3. Open the *PYTHON* directory and extract the contents of *cads-scripting-1.0.tar.gz* to a temporary directory.
4. In the temporary directory, edit *setup.cfg*. Replace `<PythonInstallDir>` with PASW Collaboration and Deployment Services scripting installation path. If no value is specified, the path will default to Python library, for example */usr/lib/python2.4*.

```
[install]
install-base = <PythonInstallDir>
install-data = <PythonInstallDir>
install-purelib = <PythonInstallDir>
install-scripts = <PythonInstallDir>
install_headers = <PythonInstallDir>
```

5. At a command prompt, change the current directory to the folder where you extracted *cads-scripting-1.0.tar.gz*. Execute the following command:

```
python setup.py install
```

To install PASW Collaboration and Deployment Services Scripting on an IBM i system:

1. Log into your IBM i system using a Telnet terminal.

   Insert the installation media.
2. Start QShell with the following command

```
QSH
```

3. Change the directory to */qopt/PASW/IBMi/Python*.
4. Copy the content of the directory to a temporary location.
5. Run the installation script by executing the following command:

   ```
   ./PyInst.scr
   ```

   Python is installed as */QOpenSys/usr/local/bin/python2.4* and PASW Collaboration and Deployment Services Scripting is installed in */QOpenSys/usr/local/lib/python2.4/site-packages*.

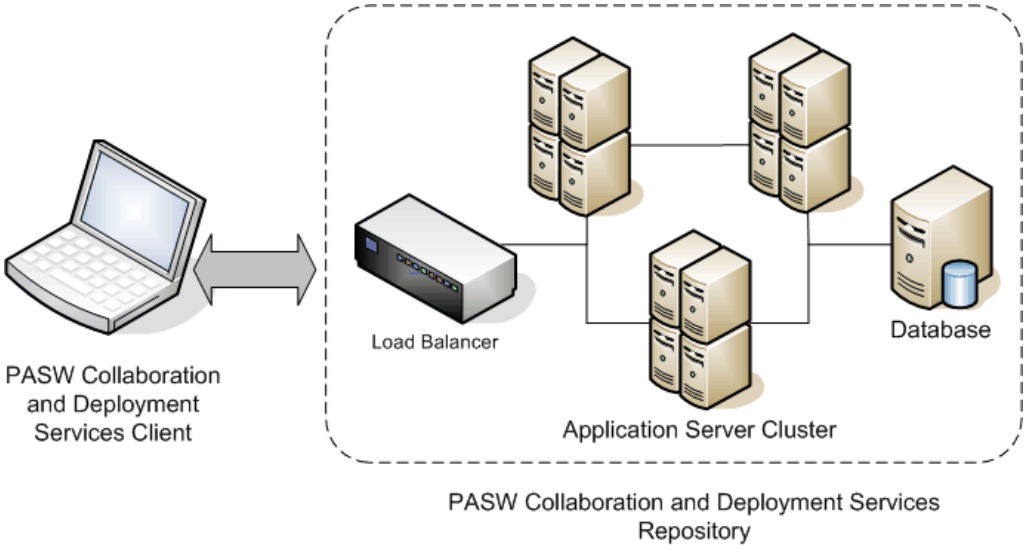
   For information about using scripting, see the customization documentation.

Chapter
4

Clustering

The repository can be deployed into an environment of clustered J2EE application servers. Each application server in the cluster should have the identical configuration for the hosted application components and the repository is accessed through a hardware- or software-based load balancer. This architecture allow processing to be distributed among multiple applications servers and it also provides redundancy in case of a single server failure.

Figure 4-1
PASW Collaboration and Deployment Services clustered deployment architecture



PASW Collaboration and Deployment Services currently supports clustering WebSphere and WebLogic application servers.

Installation

The process of installing the repository into the cluster includes the following steps:

- Initial installation and configuration of application components on an arbitrarily selected node in the cluster, which is performed by the repository installation wizard.
- Subsequent deployment of the application components into all of the nodes of the cluster performed through Jython-based script utilities or manually.

Initial installation of the repository components must follow these guidelines:

- The repository should be installed on a single node in the cluster.

- The cluster install location should be a shared directory available to all nodes in the cluster as a sheared directory or a mounted drive.
- In the setup wizard, clustered install option must be selected.
- Regardless of the application server type, the following application server information must be provided:

Property	Description
Cluster Install Location	Location of the files to be deployed into the cluster. This directory will contain the repository applications and configuration files and a set of scripts that can assist in cluster configuration.
Cluster Name	The name of the WebSphere/WebLogic cluster that you will deploy into. If a cluster has been pre-configured, the cluster name must be provided. Otherwise, you must remember the name you specify because it will be required to create the cluster at a later time.
Load Balancer URL	The URL the clients will use to connect to the cluster. Typically this will be the URL of the load balancer.
Secure Communication	The option specifies whether secure communication will be used for HTTP/SOAP messages within the cluster. If it is selected, SSL must be configured in the cluster.

- The setup must be completed. For more information about the wizard, see Installing the Repository on p. 14

After initial installation and configuration has been completed, the following directory structure is created in cluster install directory:

Subdirectory	Description
bin	OS-specific scripts for automating cluster deployment and configuration.
doc	Text files containing application server-specific instructions for deploying the repository into a cluster.
lib	Global libraries required for running the repository.
licensing	Repository licensing files.
logging	Logging configuration files.
scripts	Jython scripts for automating cluster deployment and configuration.
toDeploy	Deployable application files are located

Follow the instructions in the documentation files listed below to complete a script-assisted or manual deployment into all nodes.

Table 4-1
WebLogic Deployment Instructions

File Name	Description
weblogic-cluster-readme.txt	General information about manual and script-based deployment to a WebLogic cluster.
weblogic-config.txt	The properties in the *<cluster install location>/scripts/config.ini* file that is used to control script-based cluster configuration and deployment.

File Name	Description
weblogic-cluster-script-readme.txt	The information for deploying the repository into a WebLogic cluster using script-based utilities.
weblogic-cluster-manual-readme.txt	The information necessary to manually deploy the repository into a WebLogic cluster. This installation method is intended for advanced users only. It is expected that the cluster has already been configured and is ready for deployment.
load-balancer-readme.txt	Recommended load balancer configuration.

Table 4-2
WebSphere Deployment Instructions

File Name	Description
websphere-cluster-readme.txt	General information about manual and script-based deployment to a WebSphere cluster.
websphere-config.txt	The properties in the *<cluster install location>/scripts/config.ini* file that can be used to control cluster configuration and deployment.
websphere-config.txt-cluster-script-readme.txt	The information for deploying the repository into a WebSphere cluster using script-based utilities.
websphere-config.txt-cluster-manual-readme.txt	The information necessary to manually deploy the repository into a WebSphere cluster. This installation method is intended for advanced users only. It is expected that the cluster has already been configured and is ready for deployment.

Load Balancer Configuration

A software- or hardware-based load balancer must be configured for accessing the repository in a clustered environment. Both WebLogic and WebSphere application servers provide built-in software-based load-balancer utilities.

WebLogic Apache Plugin

WebLogic ships with a plugin that can be used with the Apache Web Server to act as a load balancer.

The plugin setup includes the following steps:

1. Install Apache Web Server. For more information, see Apache documentation at *http://httpd.apache.org/docs/2.0/install.html*
2. Configure the WebLogic plugin. For more information, see WebLogic documentation. It can be accessed online at *http://e-docs.bea.com/wls/docs92/plugins/apache.html*.

Plugin configuration requires editing the corresponding section of the configuration *httpd.conf* file to specify the nodes in the cluster, for example:

```
# Sample from httpd.conf
```

```
LoadModule weblogic_module modules/mod_wl_20.so

<IfModule mod_weblogic.c>
  Debug                ON
  DebugConfigInfo      ON
  KeepAliveEnabled     ON
  KeepAliveSecs        30
  MatchExpression      WebLogicCluster=WLC1:8080,WLC2:8080,WLC3:8080|Debug=ON
</IfModule>
```

IBM HTTP Server for WebSphere Application Server

IBM HTTP server can be configured to act as a load balancer.

The configuration includes the following steps:

1. Install IBM HTTP Server. For more information, see WebSphere documentation. It can be accessed online at *http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp*
2. Use the administration console to create a Web server object.
3. Use the administration console to generate a plugin descriptor and propagate it to IBM HTTP Server.
4. Start IBM HTTP Server.

Updating PASW Collaboration and Deployment Services in Clustered Environment

Occasionally it may be necessary to install updates for the repository as such updates become available. Updates are deployed as compressed files with **.package* extension with the Package Manager utility. For more information, see Updating the Repository in Chapter 8 on p. 67.

To perform an update in a clustered environment:

1. Run Package Manager on the main node in the cluster. This will create *<cluster install location>/updates/<timestamp>/toDeploy* directory containing updated components that subsequently must be deployed to the other nodes in the cluster.
2. Deploy the updates to other nodes by one of the following methods:
   - Use administration console of the application server.
   - Specify the path to the application updates by editing the deploy.directory property of *config.ini* and run *platformDeploy.py* script with the applications command-line option.

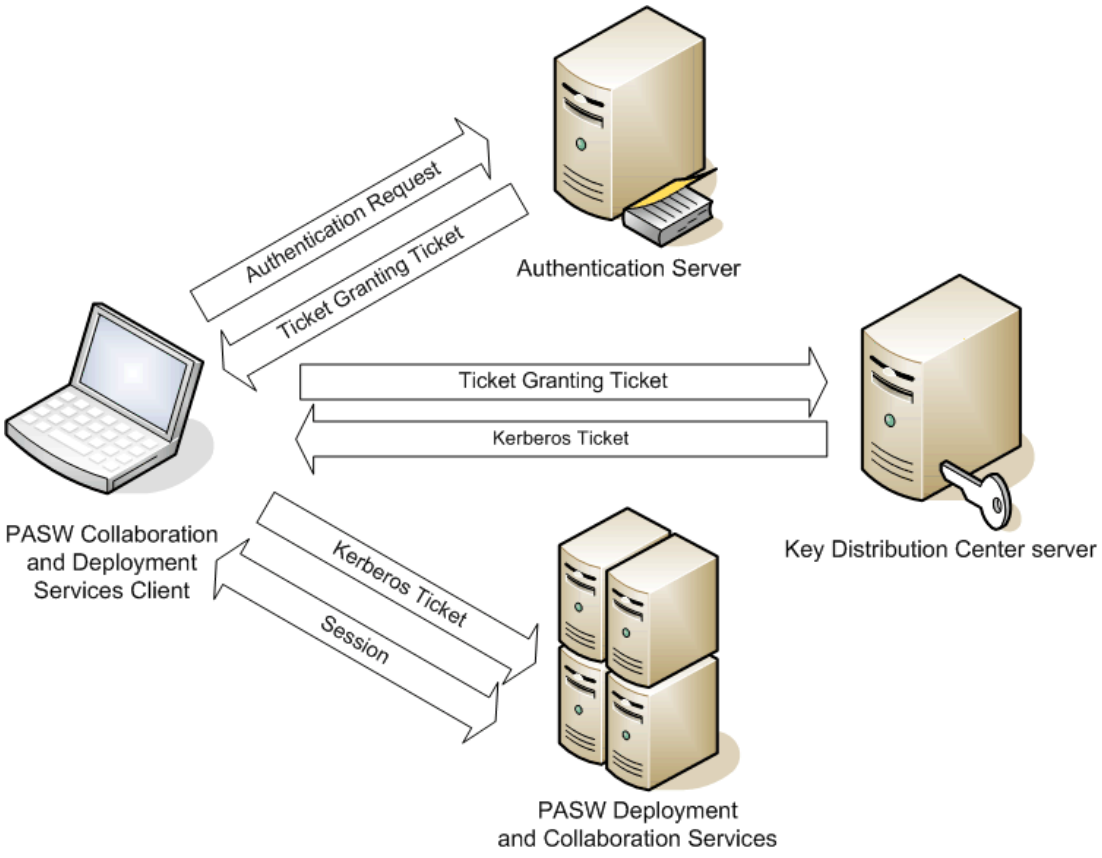
Chapter
5

Single Sign-On

Single sign-on (SSO) is a method of access control that enables a user to log in once and gain access to the resources of multiple software systems without being prompted to log in again. PASW Collaboration and Deployment Services provides the single sign-on capability by initially authenticating users through an external directory service based on **Kerberos** security protocol, and subsequently using the credentials in all PASW Collaboration and Deployment Services applications, for example, Deployment Manager, Deployment Portal, or a portal server without additional authentication.

Note: Single sign-on is not allowed for browser-based Deployment Manager.

Figure 5-1
PASW Collaboration and Deployment Services SSO architecture



For example, if PASW Collaboration and Deployment Services is used in conjunction with Windows Active directory, to enable single sign-on it is necessary to configure **Kerberos Key Distribution Center (KDC)** service. The service will supply session tickets and temporary session keys to users and computers within an Active Directory domain. The KDC must run on each domain controller as part of Active Directory Domain Services (AD DS). When single sign-on is enabled, PASW Collaboration and Deployment Services applications log into to a Kerberos domain and use Kerberos tokens for Web services authentication. It is strongly recommended that SSL communication be configured for the repository if single sign-on is enabled.

PASW Collaboration and Deployment Services single sign-on configuration is performed on the Server Administration tab of Deployment Manager. For more information, see PASW Collaboration and Deployment Services administrator documentation.

The following prerequisites must be in place before single sign-on is set up for PASW Collaboration and Deployment Services:

- Directory authentication server must be configured. Authentication can be based on Microsoft Active Directory, OpenLDAP directory, or IBM i profile directory.
- Kerberos Key Distribution Center server must be configured.
- Credential delegation must be enabled for the Kerberos Service Principal on the Kerberos Key Distribution Center server. The procedure for enabling credential delegation will be different depending on your directory server and Kerberos environment.
- Kerberos credential delegation must also be enabled on all client systems.
- For Windows client systems, the registry must be updated for Kerberos LSA access.
- Depending on the database used with the repository, the database may need to be configured for single sign-on.
- Depending on the application server used with repository, it may be necessary to update the application server configuration.
- Windows client systems must have HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\ registry value updated. For more information, Updating Windows Systems Registry for Single Sign-On
- For thin-client access to PASW Collaboration and Deployment Services (for example, with Deployment Portal), the Web browser must have Simple and Protected GSS-API Negotiation (SPNEGO) enabled.

Updating Windows Systems Registry for Single Sign-On

PASW Collaboration and Deployment Services installation disk includes registry update files for configuring Windows XP SP2, Windows Vista, and Windows 2003 systems for Kerberos-based single sign-on. The files are as follows:

- */PASW/Kerberos/Win2003_Kerberos.reg*
- */PASW/Kerberos/WinXPSP2_Kerberos.reg*

For Windows Vista systems, use the *Win2003_Kerberos.reg* file.

The registry files allow the system administrator to push registry changes to all systems on the network that must have single sign-on access to PASW Collaboration and Deployment Services.

Chapter 6

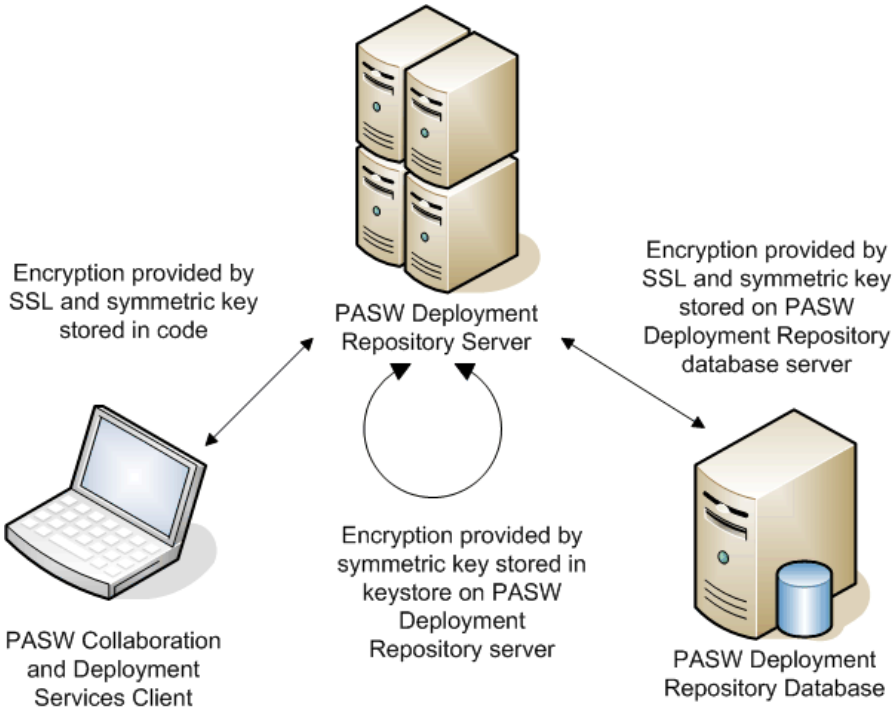
FIPS 140–2 Compliance

The Federal Information Processing Standard (FIPS) Publication 140-2, FIPS PUB 140-2, is a U.S. government computer security standard used to accredit cryptographic modules. The document specifies the requirements for cryptography modules which include both hardware and software components, corresponding to four different levels of security that are mandated for organization that do business with the U.S. government. PASW Collaboration and Deployment Services can be configured to provide Security Level 1 as specified by FIPS 140-2.

Security configuration for FIPS 140-2-compliance must follow these guidelines:

- Communications between the repository and client applications must use SSL for transport layer security of general data transfers. Additional AES encryption is provided for credential passwords using a shared key stored in the application code. For more information, see Using SSL to Secure Data Transfer in Chapter 7 on p. 63.
- The repository server uses AES algorithm with the key stored in a keystore on the server file system to encrypt passwords in PASW Collaboration and Deployment Services configuration files, application server configuration files, security provider configuration files, etc.
- Communications between the repository server and the database server can optionally use SSL for transport layer security for general data transfer. AES encryption is provided for credential passwords, configuration passwords, user preference passwords, etc. using a shared key stored in a keystore on the database server file system.

Figure 6-1
PASW Collaboration and Deployment Services FIPS 140-2-compliant security setup



Repository Configuration

PASW Collaboration and Deployment Services repository configuration for FIPS 140-2-compliance must follow these guidelines:

- The database must be set up to accept SSL communications; the JCE encryption module must also be configured.
- If the repository is installed on UNIX, the default JRE must be set up with a JCE module.
- The application server JRE must also be set up with a JCE module.
- The application server must be configured to accept SSL communications; a JCE module must also be configured.
- If the repository is installed on Windows, you must exit the installation at setup screen, configure a JCE module, then restart the setup and select to run in FIPS 140-2-compliant mode on the appropriate screen. For more information about the installation wizard, see Installing the Repository on p. 14
- If PASW Collaboration and Deployment Services is deployed into a clustered environment, keystore must be replicated to all nodes in the cluster.
- The JREs that are being used by SPSS Inc. server applications interacting with PASW Collaboration and Deployment Services, such as PASW Statistics Server and PASW Modeler Server, must have SSL certificates installed.

Desktop Client Configuration

For PASW Collaboration and Deployment Services desktop client applications, such as Deployment Manager, JCE encryption module must be enabled for the JRE used to run the applications. The JRE must have SSL certificates installed.

Browser Configuration

- Mozilla Firefox can be configured to run in FIPS 140-2 compliant mode by modifying the application options. For more information, see *http://support.mozilla.com/en-US/kb/Configuring+Firefox+for+FIPS+140-2*.
- Internet Explorer configuration requires enabling Windows cryptography and modifying the browser settings. For more information, see*http://support.microsoft.com/kb/811833*.
- Apple Safari cannot be used in FIPS 140-2 compliant mode.

Chapter
7

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:

- Obtain and install the SSL certificate and keys.
- If desired, install unlimited strength encryption on the client computers.
- If using a self-signed certificate, copy the certificate on the client computer.
- Instruct end users to enable SSL when connecting to the server.

Note: Occasionally a server product acts as a client. An example is PASW Statistics Server connecting to the repository. In this case, PASW Statistics Server is the *client*.

Obtain and Install 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.
- 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.

Install 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.

Copy the Certificate File to Client Computers

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

If you are using a self-signed certificate, you need to copy the certificate to the *client* computers. Be aware that a server computer may also act as a client. An example is PASW Statistics Server connecting to the repository. In this case, PASW Statistics Server is the *client*, and therefore you need to copy the certificate for the repository server to the PASW Statistics Server.

- Copy *root.pem* to the following location on the client computers. By default, all SPSS Inc. client products look in this location for trusted self-signed certificate files. If you would like to use another location, create an `SSL_CERT_DIR` environment variable and set the value of the variable to the location.

  Windows. *C:\Documents and Settings\All Users\Application Data\SPSSInc\certificates*

  If you already copied a *root.pem* file to the client for another SPSS Inc. product, append the certificate information from the new server to the existing *root.pem* file. This file is a text file so you can copy and paste the certificate.

Add the Certificate to Client Keystore (For Connections to PASW Collaboration and Deployment Services)

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

If you using SSL to connect to PASW Collaboration and Deployment Services 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.

- 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`.

Instruct 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.

URL Prefix Configuration

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

1. Log into PASW Collaboration and Deployment Services using browser-based Deployment Manager.
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 Deployment Manager installation. This allows the LDAPS connection to be established between the 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 Deployment Manager.
4. From the Tools menu, choose Server Administration.
5. Log in to a previously defined administered repository 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.

Chapter
8

Updating the Repository

Occasionally it may be necessary to install updates for the repository as such updates are made available. It may also be necessary to install optional components that extend repository functionality to support additional content types, security providers, etc., or install Deployment Manager updates which will be pushed to clients when they access the server.

Updates are deployed on the repository server as compressed files with **.package* extension in the *<PASW Collaboration and Deployment Services Installation Directory>/staging/* directory with the Package Manager utility. Optional packages included with the installation are located in *<PASW Collaboration and Deployment Services Installation Directory>/pasw4.0/optional/* directory.

Installing Packages

Package Manager utility can be used as a GUI application or as a command line application. It can also be called in batch mode by other applications to install their package files into PASW Collaboration and Deployment Services. The repository must be stopped prior to installing packages.

Note: If WebSphere application server is used with PASW Collaboration and Deployment Services, it must be running while packages are being installed. WebSphere must be stopped and restarted prior to running Package Manager.

The user must have administrator-level privileges to be able to install packages. The deployed packages are located in the *<PASW Collaboration and Deployment Services Installation Directory>/staging/*.

To prevent the newer version of a package from being overwritten by an older version, Package Manager performs a version check. Package manager also checks for prerequisite components to ensure that they are installed and their versions are equal to or newer than the required version. If any of these checks fail, the package is marked as missing prerequisites in the dialog pane nut can still be installed. However, it is not recommended to install packages that failed dependencies checks.

Note: Dependency checks failures cannot be overridden if Package Manager is called in batch mode.

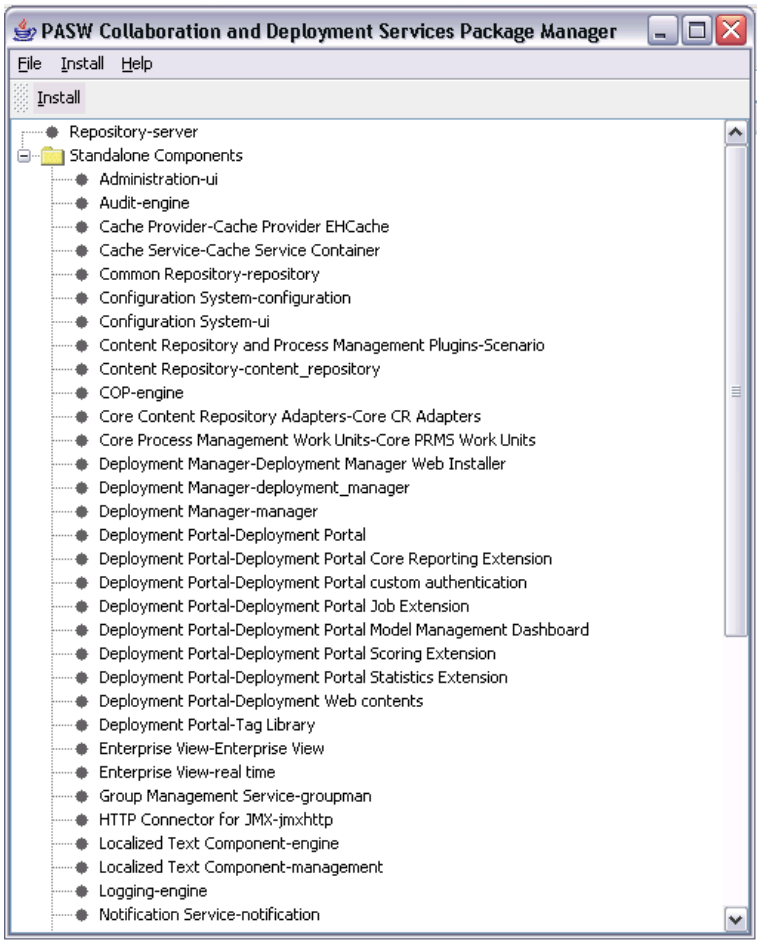
To install a package using the GUI interface:

1. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.
2. Depending on the operating system, execute *packagemanager.bat* on Windows or *packagemanager.sh* on UNIX.
3. When prompted, enter the user name and password.

Figure 8-1
Admin Login

4. Click OK to login. The Package Manager dialog box appears.

Figure 8-2
Package Manager



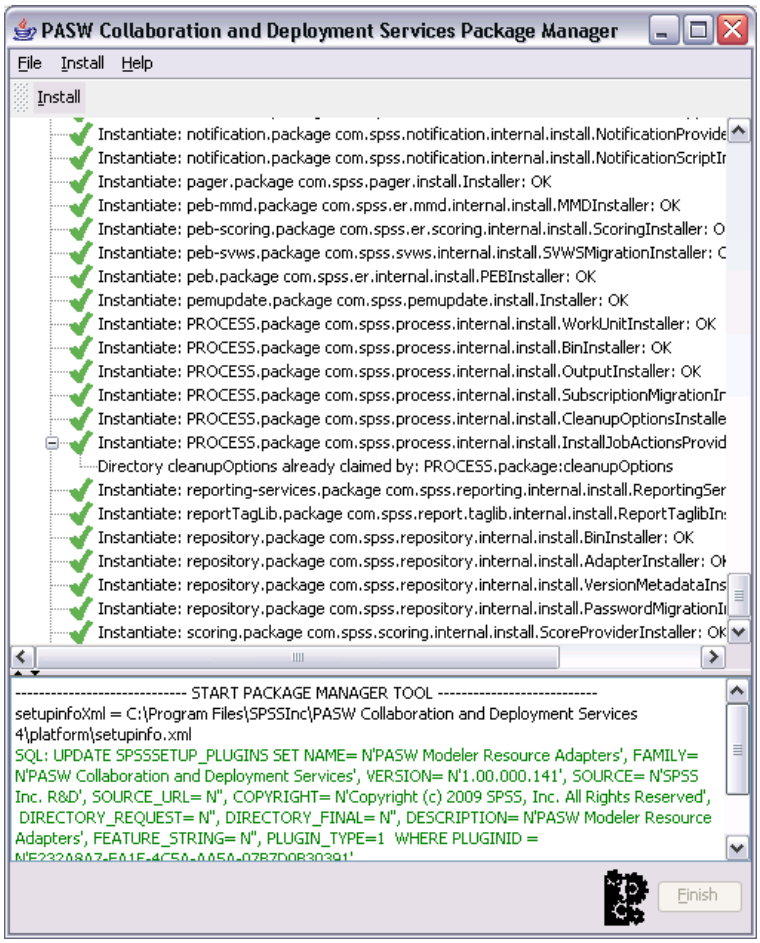
5. From the Install menu, select Install.

6. From the installation path, navigate to the location of the package file.

7. Select the package and click OK. The installation status panel appears.

If failed dependencies are detected, the panel displays the Install packages with failed dependencies check box. Select the check box and click OK to continue the installation, or click Cancel to abort.

Figure 8-3
Package Manager installation



Installation log can be found in *<PASW Collaboration and Deployment Services Installation Directory>/setup/logs/setup.log*.

8. Click Finish when the installation is complete. If errors occur during installation, they are displayed in red in the bottom pane. To close Package Manager dialogue box, click Abort.

To install a package from the command line:

1. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.
2. Depending on the operating system, execute *clipackagemanager.bat* on Windows or *clipackagemanager.sh* on UNIX.
3. When prompted, enter the user name and password.

Note: The password is not masked when it is entered in the command prompt.

4. Type the install command and press Enter. The command must include the `install` option and the path of the package in quotes, as in the following example:

   install 'C:\dir one\package1.package'

   If failed dependencies are detected, you will be presented with a choice to ignore the failures and continue the installation or abort.

5. When the installation is completed, use exit command to exit Package Manager.

   Note: To display more command line install options, type `help` and press Enter key. The option include:

   - info "<package path>" Display information for a specified package file
   - install "<package path>" Install the specified package files into PASW Collaboration and Deployment Services
   - tree Display installed package tree information

Chapter 9

Saving and Restoring the Repository

Occasionally it may be necessary to save and restore the repository—for example, when it is migrated to a different server. The configuration and the contents of the repository are saved and restored using the Save Tool and the Restore Tool. Included are the following:

- Content repository files and folder structure
- Scheduling and notification components
- Local users
- Locally defined overrides of remote directory user lists and groups
- Role definitions and membership
- User preferences
- Notification templates
- Icons

The Save Tool allows the user to encrypt the data.

Important! The save and restore mechanism in PASW Collaboration and Deployment Services is intended primarily for migration purposes and is not a substitute for database backup. A regular backup of the repository database outside of PASW Collaboration and Deployment Services is strongly recommended.

Saving the Repository

The Save Tool can be used as a GUI application or as a command line utility. On systems without a GUI interface, it must be used as a command-line application. It can also be called in batch mode by other applications. The user must be assigned the Administrator role in PASW Collaboration and Deployment Services to perform the save operation. It is strongly recommended to stop PASW Collaboration and Deployment Services before saving.

Saving Using the GUI Application

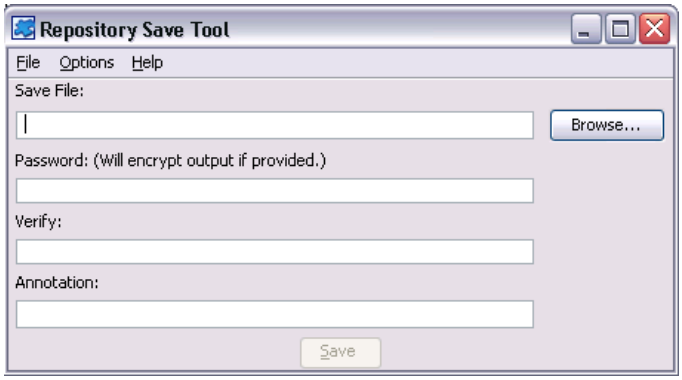
To save the repository using the GUI application:

1. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.
2. Depending on the operating system, execute *save.bat* on Windows or *save.sh* on UNIX.
3. When prompted, enter the username and password.

Figure 9-1
Local Administrator Logon dialog box for the Save Tool

4. Click OK to log in. The Save Tool dialog box opens.

Figure 9-2
Save Tool dialog box



5. Select the save format.
   - To save the data as a compressed archive, from the menus choose:

Options
  Single .PESsave

   - To save the data as a collection of files, from the menus choose:

Options
  Directory with Files

6. Enter the file/directory path or click theBrowse button to navigate to the location where the data will be saved.

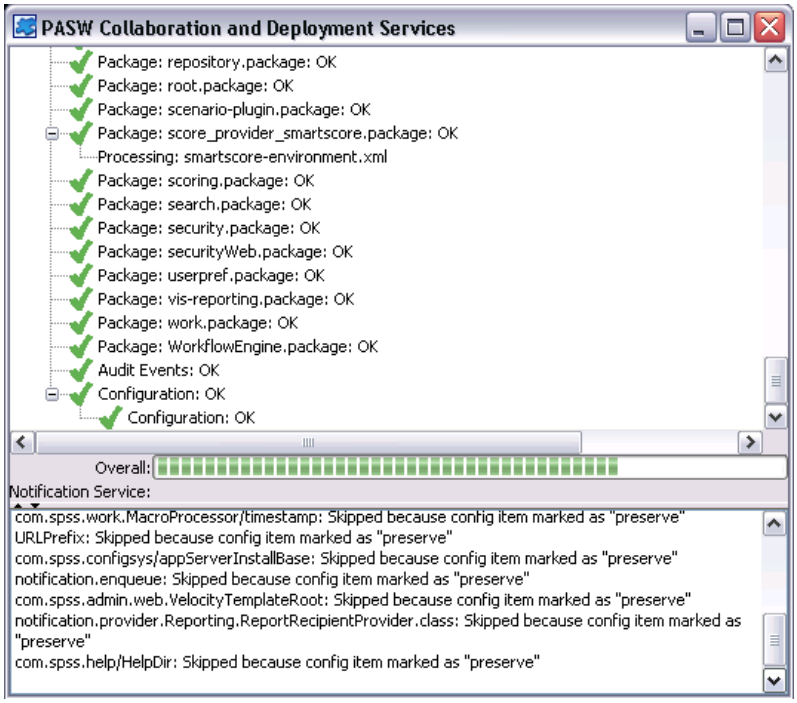
   Note: If the archive file has been selected as the save option, the *.PESsave* extension will be automatically appended to the specified filename. If the directory has been selected as the save option, the target directory cannot already contain PASW Collaboration and Deployment Services save data.

7. To encrypt the data, enter and verify the password. Any alphanumeric string can be used as the password.

8. Add the annotation to the saved data if necessary. An annotation is a descriptive string that will be displayed when the data source (archive file or directory) is selected for system restore.

9. Click Save. The status panel appears.

Figure 9-3
Save operation progress



If errors occur during the save operation, they are displayed in red in the bottom pane. The installation log can be found in *<PASW Collaboration and Deployment Services Installation Directory>/setup/logs/saverestore.log*. At the end of the operation, a message specifying duration is also displayed.

10. Close the status panel. This will also close the Save Tool.

Saving Using the Command Line

To save the repository using the command line utility:

1. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.

2. Execute the saverestore -headless command with the following required arguments:
   - -userid <user ID>. The user under whose credential the save operation is being performed.
   - -userpassword <password>. The password of the user.
   - -save <data location path>. The path of the saved data.

   Optional arguments include:
   - -explode. The option to save the data as a directory.
   - -filepassword <file password>. Encryption password.

- -annotation <annotation>. The annotation string. If the annotation contains spaces, it must be enclosed in quotation marks.
- -lang <language code>. The language code for localized instances of PASW Collaboration and Deployment Services.

The following example illustrates saving the contents of the repository in a password-protected file with an annotation.

```
saverestore -headless -userid admin -userpassword pass1234 -save c:/temp/saveFile -filepassword secret
-annotation "Preparing data for migration 1/09/2009"
```

Restoring the Repository

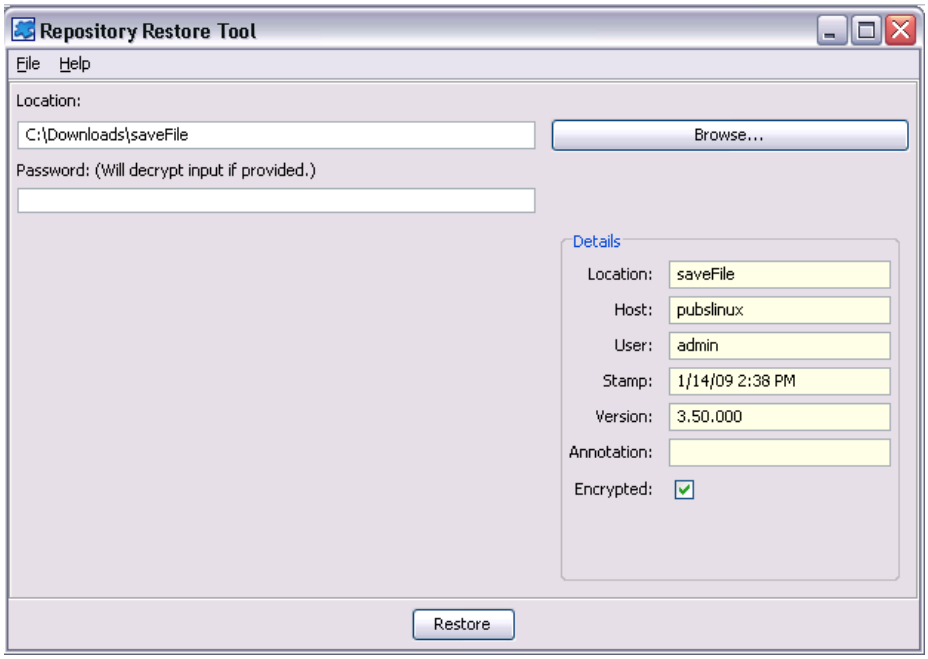
The Restore Tool can be used as a GUI application or as a command line application. On systems without a GUI interface, it must be used as a command-line application. It can also be called in batch mode by other applications. The user must be assigned the Administrator role in PASW Collaboration and Deployment Services to perform the save operation. If PASW Collaboration and Deployment Services is being restored over an existing instance, the existing content will be overwritten. In such cases, it is strongly recommended to stop PASW Collaboration and Deployment Services before restoring. If PASW Collaboration and Deployment Services is being migrated to another server, application components must already be in place on the new host. Therefore, the installation must be run prior to restoring. After the repository has been restored, it must be reindexed. For information about reindexing, see administrator documentation.

Restoring Using the GUI Application

To restore the repository the GUI application:

1. Stop the repository server.
2. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.
3. Depending on the operating system, execute *restore.bat* on Windows or *restore.sh* on UNIX.
4. When prompted, enter the username and password.
5. Click OK to log in. The Restore Tool dialog box opens.

Figure 9-4
Restore Tool dialog box: Restore tab



6. Enter the file/directory path or click the Browse button to navigate to the location where the data were previously saved. After the data source has been selected, the corresponding information is displayed in the Details group box.

   Note: If the restore utility is run, the specified data source path is retained and will be displayed by default the next time the restore utility is opened.

7. If the data have been encrypted, enter the password. The field is unavailable for unencrypted files.

8. Click Restore. The status panel appears. If errors occur during the restore operation, they are displayed in red in the bottom pane. The installation log can be found in *<PASW Collaboration and Deployment Services Installation Directory>/Enterprise Repository/setup/logs/saverestore.log*. At the end of the operation, a message specifying duration is also displayed.

9. Close the status panel. This will also close the Restore Tool.

Restoring Using the Command Line

To restore the repository using the command line utility:

1. Stop the repository server.

2. Navigate to *<PASW Collaboration and Deployment Services Installation Directory>/setup/*.

3. Execute the saverestore -headless command with the following required arguments:
   - -userid <user ID>. The user under whose credential the restore operation is being performed.

- -userpassword <password>. The password of the user.
- -restore <data location path>. The path of the restored data.

Optional arguments include:

- -filepassword <file password>. For encrypted files, the password.
- -setupdir <path>. Option to indicate that the setup directory is different from the current directory.

The following example illustrates restoring the contents of the repository from password-protected file.

```
saverestore -headless -userid admin -userpassword pass1234 -restore c:/paswuser/saveFile -filepassword secret
```

Restoring Files from Previous Versions

When restoring saved information, the repository receiving the information must be the same version as the repository from which the data was saved. In addition, both repository servers must have the same packages installed for the restored information to function as it did in the original repository.

If you have a repository from a previous version, you can migrate the information in it to a new version in one of two ways.

- Copy the database used in the older version. When installing the newer version of the repository, use this copy as the database with the option to retain existing data.
- Restore the older saved information into the new repository. Afterwards, rerun the setup tool for the new version, using the default values from the older installation, with the option to retain existing data.

Chapter
10

Logging Services

Logging tools are essential when troubleshooting existing problems as well as when planning preventive maintenance activities. As system and application events are generated, administrative personnel can be alerted when warning thresholds are reached or critical system events occur. Additionally, verbose information output can be stored in a text file or Syslog record for analysis at a later time.

The repository uses the **log4j** package for handling log information. Log4j is Apache Software Foundation's logging solution for **J2EE** applications. The log4j approach permits logging control using an XML-based configuration file; the application binary does not have to be modified. For a comprehensive discussion of log4j, see the log4j Web site (*http://logging.apache.org/log4j/docs/*).

The location of the *log4j.xml* configuration file varies, depending on the host application server:

- JBoss—<*JBoss installation directory*>*server**default**conf*.
- WebLogic—<*Repository installation directory*>*SPSSDomain**lib*\. Note that log4j components used by PASW Collaboration and Deployment Services for logging on to WebLogic are part of the repository installation.
- WebSphere—<*Repository installation directory*>*setup**resources**websphere*.

This file controls both the destination and the amount of log output. Configuration of log4j is handled by modifying this file to define **appenders** for log destinations and to route **logger** output to those appenders.

Appenders

Log output can be sent to a variety of destinations. In log4j, the destination is referred to as an **appender**. Table 10-1 describes the appenders available in log4j.

Table 10-1
Log4j appenders

Appender class	Description
org.apache.log4j.ConsoleAppender	*System.out* or *System.err* streams
org.apache.log4j.FileAppender	Log file
org.apache.log4j.DailyRollingFileAppender	Log file that is automatically backed up at a specified frequency
org.apache.log4j.RollingFileAppender	Log file that is automatically backed up at a specified size
org.apache.log4j.net.SMTPAppender	E-mail notification of log events
org.apache.log4j.jdbc.JDBCAppender	Database for log events
org.apache.log4j.net.JMSAppender	Notification of log events using Java Messaging Service

Appender class	Description
org.apache.log4j.lf5.LF5Appender	Swing-based logging console
org.apache.log4j.nt.NTEventLogAppender	Appends log events to NT event logs
org.apache.log4j.net.SocketAppender	Remote log server
org.apache.log4j.net.SocketHubAppender	Set of remote log servers
org.apache.log4j.net.SyslogAppender	Syslog daemon
org.apache.log4j.net.TelnetAppender	Read-only socket that can be monitored using TCP/IP
org.apache.log4j.ext.SNMPTrapAppender	Log4j extension that sends SNMP traps

The configuration file defines appenders using the appender element. This definition includes a name and class specification, plus any appender-specific parameters. The following example illustrates a *ConsoleAppender*. For more information about the child elements of appender, see the log4j documentation.

```
<appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="Target" value="System.out"/>
   <param name="Threshold" value="INFO"/>
   <layout class="org.apache.log4j.PatternLayout">
      <!–The default pattern: Date Priority [Category] Message\n –>
      <param name="ConversionPattern" value="%d{ABSOLUTE} %-5p [%c{1}] %m%n"/>
   </layout>
</appender>
```

By default, repository uses two appenders:

- *FILE*, a *DailyRollingFileAppender* that sends the log to a file named *server.log* in the JBoss log folder. At midnight, the year, month, and day are appended as a suffix to the filename, and a new *server.log* file begins recording log events for the next day.
- *CONSOLE*, a *ConsoleAppender* that sends the log to the *System.out* stream for display in a console window.

In addition, the configuration file includes a definition for a *DailyRollingFileAppender* named *FILE-MM*. This appender corresponds to a file named *mm.log* in the JBoss log folder that is similar to the *server.log* file. However, *FILE-MM* can be used for repository loggers to separate log information for the application from log information for the application server. The *FILE-MM* appender appears below:

```
<appender name="FILE-MM" class="org.jboss.logging.appender.DailyRollingFileAppender">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="File" value="${jboss.server.home.dir}/log/mm.log"/>
   <param name="Append" value="false"/>
   <!–Rollover at midnight each day –>
   <param name="DatePattern" value="'.'yyyy-MM-dd"/>
   <layout class="org.apache.log4j.PatternLayout">
      <!–The default pattern: Date Priority [Category] Message\n –>
      <param name="ConversionPattern" value="%-5p [%c] %m%n"/>
   </layout>
</appender>
```

Defining Appenders

To define an appender:

1. Open the *log4j.xml* configuration file in a text editor.
2. Locate the appender element that corresponds to the logging destination you want to employ. If the appender element is commented out of the file, remove the comment symbols (<!−and −>) that enclose the appender.
3. If the configuration file does not contain the desired appender, create a new appender element. Assign a name and specify the class for the desired log destination. See Table 10-1 on p. 77.
4. Modify the content of the appender element as needed to reflect your system and network settings.
5. Save the file.

   The repository automatically updates to reflect the changes. A restart of the server is not needed.

Loggers

Loggers represent application systems that generate log output. For each logger, the *log4j* configuration file specifies both the amount of information logged and the destination for that information.

Logger names typically consist of a series of text strings separated by periods corresponding to the names of software components, such as *com.spss.process*. This naming convention defines a hierarchy of parent/child relationships for loggers. For example, the *com.spss.cmor* logger is a child of the *com.spss* logger, which itself is a child of the *com* logger. The exception to this rule is the *root* logger, which is an ancestor of all loggers in the system. The table below lists the loggers available in the repository.

Table 10-2
Loggers

Logger	Description
root	Root logger
com.spss.cmor	Repository events
com.spss.security	Security events
com.spss.process	Job scheduling events

In the configuration file, the root and category elements define logger properties. The root element defines log destinations for all loggers in the system. The category element allows specification of behavior for particular loggers. The category specification for the repository follows:

```
<category name="com.spss.cmor">
   <priority value="WARN"/>
</category>
<category name="com.spss.security">
   <priority value="WARN"/>
</category>
<category name="com.spss.process">
   <priority value="WARN"/>
```

```
</category>
```

The priority element defines a logging level for the corresponding logger. The level controls the amount of information logged.

Logging Levels

The amount of information contained in the log output is controlled by the logging level. Valid levels include:

- **FATAL**. Severe errors that cause the application to fail.
- **ERROR**. *FATAL*-level errors plus errors resulting from specific requests that allow the application to continue functioning.
- **WARN**. *ERROR*-level errors plus suboptimal or unexpected events.
- **INFO**. *WARN*-level errors plus status messages reflecting general application processes.
- **DEBUG**. *INFO*-level errors plus detailed status messages used for application debugging purposes.

Levels are hierarchical; each level includes all of the output for levels above it. For example, setting the logging level to *WARN* results in all *WARN*, *ERROR*, and *FATAL* output being logged.

Configure the logging level for a particular logger using the priority element in the configuration file. This element uses the value attribute to specify the logging level. The following example sets the level for the *com.spss.cmor* logger to *WARN*:

```
<category name="com.spss.cmor">
   <priority value="WARN"/>
</category>
```

By default, the repository logs all information at the *WARN* level.

In the absence of a priority element for a logger, that logger inherits the level of the nearest ancestor. As a result, the logging level for all repository loggers could be set to the same level using the *com.spss* parent logger:

```
<category name="com.spss">
   <priority value="WARN"/>
</category>
```

Modifying Logging Levels

To modify a logging level:

1. Open the *log4j.xml* configuration file in a text editor.
2. Locate the category element for the logger to be modified.
3. Change the value for the child priority element to the desired logging level. For more information, see Logging Levels on p. 80.
4. Save the file.

The repository automatically updates to reflect the changes. A restart of the server is not needed.

Routing Logs

Routing log information involves associating appenders with loggers. **Loggers** define the amount of information being logged; **appenders** define the destination for the information. In the *log4j* configuration file, use the appender-ref element to assign appenders to loggers.

In log4j, all log output is sent to any appenders associated with the *root* logger. The repository uses the *CONSOLE* and *FILE* appenders for the *root* logger, defined by using two appender-ref elements as children of the root element.

```
<root>
   <appender-ref ref="CONSOLE"/>
   <appender-ref ref="FILE"/>
</root>
```

To send the output for a specific logger to an alternative destination, add an appender-ref element as a child of the category element for the logger. For example, suppose we wanted to isolate all job scheduling log output in a single file. Using the appender-ref element, we add a reference to the *FILE-MM* appender for the *com.spss.process* logger.

```
<category name="com.spss.process">
   <priority value="WARN"/>
   <appender-ref ref="FILE-MM"/>
</category>
```

In this case, the job scheduling log is sent to the *FILE-MM* appender plus any appenders defined for the *root* category. To prevent the scheduling log from going to the *root* appenders, set the additivity attribute for the appender-ref element to *false*.

```
<category name="com.spss.process">
   <priority value="WARN"/>
   <appender-ref ref="FILE-MM" additivity="false"/>
</category>
```

Assigning Appenders

To assign an appender to a logger:

1. Open the *log4j.xml* configuration file in a text editor.
2. Locate the category element for the logger to be modified.
3. Add a child appender-ref element. Supply an appender name as the value for the ref attribute. Use the additivity attribute to control whether the logger should continue to send information to the root appenders.
4. Save the file.

   The repository automatically updates to reflect the changes. A restart of the server is not needed.

Chapter
11

Import Tool

The Import Tool allows you to populate the repository with any file type, such as PASW Modeler streams. The PASW Modeler Stream Library is a set of streams that can help you learn to browse, view, and retrieve stored items. It also provides a methodology for organizing your own data mining work product. The streams provide a set of reusable data mining techniques that can help you to formulate solutions to business problems quickly.

The PASW Modeler Stream Library includes a set of sample streams organized into the following categories:

- Data Preparation—After cataloging data resources, data preparation includes any cleaning, selecting, constructing, integrating, and formatting of data.
- Data Understanding—An exploratory stage in which data are examined using plots, histograms, and basic summary statistics.
- Modeling—Information is extracted from the data using sophisticated analytical methods to select modeling techniques, generate test designs, and build and assess models.

Once the repository is installed and functioning, the streams included in the PASW Modeler Stream Library are imported into the database using the Import Tool Windows batch file or UNIX shell script. These import tools process streams, models, and standard output files included in the PASW Modeler Stream Library, but they can also be used to handle any data object stored in a file system.

Directory Structure

When the repository is installed, the Import Tool is included with the application. The tools are located in the */applications/ImportTool* directory within the repository installation directory and are described in the table below.

Table 11-1
File location and directory structure

Name	Description
ModelerStreamLibrary	Directory that contains subdirectories for: *Data Preparation* *Data Understanding* *Modeling* Each of the subdirectories contains the streams (*.str* files) that are imported into the database.
lib	Directory containing *.jar* files used by the application. These should not be changed or deleted.

Name	Description
importTool.bat	Windows batch file for importing data objects. When using the Import Tool on a supported Windows system, execute this file to populate the database.
importTool.sh	UNIX shell script for importing data objects. When using the Import Tool on a supported UNIX platform, execute this file to populate the database.
repository.properties	Configuration file containing system-specific attributes. Some attributes are required and must be changed before using the Import Tool.

Before You Begin

Before working with the Import Tool, the repository must be installed. The batch file and shell script both attempt to use the repository-installed JRE if *JAVA_HOME* is not set. You will need to change the value of the *MM_INSTALL_HOME* variable in the Windows batch file (*importTool.bat*) or the UNIX shell script (*importTool.sh*).

Before running the batch file or shell script, set the installation path of the repository. To set the installation path:

1. Open *importTool.bat* or *importTool.sh* in a text editor.
2. Change the value of *MM_INSTALL_HOME* to match the installation path of Deployment Manager.
3. Save and close the file.

Customizing Properties

Edit the *repository.properties* file using a text editor to customize the application properties. This file must specify the repository server name and login information. You may specify all of the properties for the connection, but the defaults are adequate in most cases.

Table 11-2
Description of repository.properties file

Name	Description
repository.host	The name of the server. *Required*.
repository.username	The name of the user being authenticated. *Required*.
repository.password	The associated password for the user being authenticated. *Required*.
streams.directory	The directory location of the files to load.
author.names	Assigned list of comma-separated names to apply to the imported files. Note that these are randomly assigned.
version.labels	Assigned version names for imported files. These are assigned in the order in which they are listed. The first time a file is imported, the first label is applied. The second time a file is imported, the second label is applied, and so on.
repository.port	The port number the server is using. By default, this value is 80. This must be changed if other applications are using the default or if the application server is assigned to another port.
repository.protocol	The protocol used. By default, this value is http.
repository.context	The URL context string.

Populating the Repository

To populate the repository, start the PASW Collaboration and Deployment Services server and execute the Windows batch file or run the shell script under a supported UNIX platform.

Note: Solaris users need to enter chmod +x importTool.sh before executing the shell script.

Verbose (and lengthy) INFO messages appear as the utility populates the PASW Collaboration and Deployment Services repository. Specific output varies for each installation but is similar to the following output:

```
Using JAVA_HOME installation at C:\SPSS\ModelManager\jre\
INFO  [main] - Creating URL with http://localhost:8080/cr-ws/services/ContentRepository
INFO  [main] - Starting directory: ClementineStreamLibrary
INFO  [main] - Validating repository connection
INFO  [main] - Connecting as admin
INFO  [main] - Service connection established.
INFO  [main] - Looking for topic: '/'
INFO  [main] - Found topic: /
INFO  [main] - Looking for topic: '//CRISP-DM'
INFO  [main] - Didn't find it.
INFO  [main] - Creating new topic: CRISP-DM in /
INFO  [main] - Created new topic with ID: 0a0b989f00b1b4c3000001028d5651008007
```

Note: The output should contain only INFO messages; output prefaced with ERROR indicates a configuration or system failure. Verify the settings in *repository.properties* and run the batch file or shell script again.

Assigning Topics

During stream import, the name of the file is used to assign a CRISP-DM topic to the stream. Topics provide searchable metadata to facilitate finding streams in the repository.

The first letter of the filename determines the topic assigned to the file. The table below describes the relationship between the first letter in the name and the assigned topics.

Table 11-3
Naming convention for topics

First letter	Assigned topics
p	CRISP-DM > Data Preparation
e	CRISP-DM > Data Understanding
m	CRISP-DM > Modeling
	CRISP-DM > Evaluation
d	CRISP-DM > Deployment

Files with names beginning with any other character are not automatically assigned a topic.

Verifying File Import

After the batch file or shell script has finished processing, verify that the files have been successfully imported using PASW Modeler or Deployment Manager.

PASW Modeler User Interface

To verify that files were imported correctly:

1. From the PASW Modeler user interface, establish a connection to the repository. For specific instructions, see the PASW Modeler documentation.
2. After a connection has been established, verify that the correct directory structure appears.

Deployment Manager User Interface

To verify that files were imported correctly:

1. From the Deployment Manager user interface, establish a connection to the repository.
2. In the Content Explorer, expand *Content Repository* by clicking the + icon.
3. Verify that the correct directory structure appears.

Chapter
12

SWDF Content Migration

PASW Collaboration and Deployment Services provides a utility for migrating legacy SPSS Web Deployment Framework (SWDF) content, including SmartViewer Web Server and Cleo objects and metadata.

Process Overview

Export File

The input to the migration utility is the export file (*.*syspak*) generated by the SPSS Web Deployment Framework 2.7 Offline Administrator. It is a compressed archive that contains the XML data files, the physical object files, the notification listener files, and the SWDF Administrator configuration files. For information about creating the file, see SWDF Administrator documentation.

The migration process requires a temporary database for storing the content. The database must be set up before the migration utility is run.

Content Migration

The general guidelines for the migration process are as follows:

- Since SWDF 2.7 did not have a root folder as such, all content is migrated with respect to the root folder: *Home* category in SWDF 2.7 would now be mapped to the root folder (*/*) in the repository. Similarly, *Uncategorized Documents* and *Expired Documents* categories are migrated as */Uncategorized Documents* and */Expired Documents* folders in the repository.
- Because SVWS 5.0 is case sensitive and PASW Collaboration and Deployment Services 4 is not, there is a possibility that there might be naming conflicts during migration. In such cases, the files are renamed by appending a number to the end of the filename preceded by a period. For example, if there are three files named *Test.doc*, *TEST.doc*, and *test.doc* in a category named *ABC*, these documents are migrated as *Test.doc*, *TEST.doc.1*, and *test.doc.2* in the folder *ABC*. There will be warnings in the migration report for the renamed files.
- In SVWS 5.0, every user has an implicit *Your Private Category*. These private categories are migrated under a predefined folder named *private_category*. Each user has a folder named after the username under this *private_category* folder. All of the files in the user's private category in SVWS 5.0 are migrated to this folder. The security to this folder is such that only that user can access it.
- SVWS 5.0 object security setting are migrated as object permissions.

- All object metadata, such as owner, author, description, keywords, expiration date, created date, last modified date, and last modified by user, are migrated to PASW Collaboration and Deployment Services 4. In case the owner was not migrated for some reason, the default owner is set to the <migration user>.
- There are two options for migrating SVWS 5.0 files that exist in multiple categories. The default option migrates the file to one repository folder and shows error messages for all others in the migration report. For example, if an SVWS 5.0 file *test.doc* is present in *CategoryA* and *CategoryB*, it will be migrated to the folder *CategoryA*. The alternative option migrates the file to one folder and adds a topic for each of the other categories associated with the file.
- Custom interfaces are not migrated.

Stop Words Migration

All available stop words from SVWS 5.0 are migrated to PASW Collaboration and Deployment Services 4. If the stop word already exists in the new database, it is flagged as a failure in the migration report.

Users and Groups Migration

The process will attempt to migrate all native and non-native users and groups present in SVWS 5.0. If a user is not recognized by a non-native security provider and no password exists in the SVWS input file, then the user is not migrated. Groups are migrated similarly to users with the exception that groups do not have a password associated with them. The migrated *Admin* group is assigned the role of *administrator*. The users and groups that have not published any documents but have been given permissions to execute reports are also migrated.

Roles Migration

SVWS 5.0 does not have the concept roles, but the Tools Privileges page of the Administrator application can be used to map tools to existing PASW Collaboration and Deployment Services roles. The Publisher tool defines which users/groups have write access to the repository. This tool can be mapped to the *Publisher* role in PASW Collaboration and Deployment Services with the appropriate actions associated with it: *Access Contents and Folders* and *Define and Manage Notifications*. Similarly, the Subscription tool can be mapped to the PASW Collaboration and Deployment Services *Subscription* role with the appropriate actions associated with it. Text Manager and the UI Text Editor tools must not be mapped to any roles in PASW Collaboration and Deployment Services 4.

Notification Migration

All notifications are migrated from SVWS 5.0 to PASW Collaboration and Deployment Services 4. In case the user is not migrated, her notifications are not migrated, and a failure is logged in the migration report. Similarly, if the folder is not migrated, then none of the associated notifications are migrated and it is logged as a failure.

User Preferences Migration

Only the e-mail address of the users will be migrated as a part of user preferences for SVWS migration.

Migration Utility

The migration utility can be used as a GUI application or as a command line application. The input arguments include the path of the SWDF export file, destination PASW Collaboration and Deployment Services host information and login, repository database connection information, temporary database connection information, and optional parameters. Before processing is started, the input argument summary is saved as *Installation Directory>/components/svws-migration/input-args-summary.txt*. While migration is processed, the details are displayed in the console window. After processing is completed, the log files are saved in *Installation Directory>/components/*.

Migration can be run in local or remote mode, depending on how Web services calls are made. If migration is run in local mode, then it is not necessary for PASW Collaboration and Deployment Services to be running. Remote mode requires PASW Collaboration and Deployment Services to be running. It is recommended to always run migration in remote mode.

To run migration in GUI mode:

1. Execute

   (Windows)
   <PASW Collaboration and Deployment Services Installation Directory>/components/svws-migration/runMigrationGui.bat

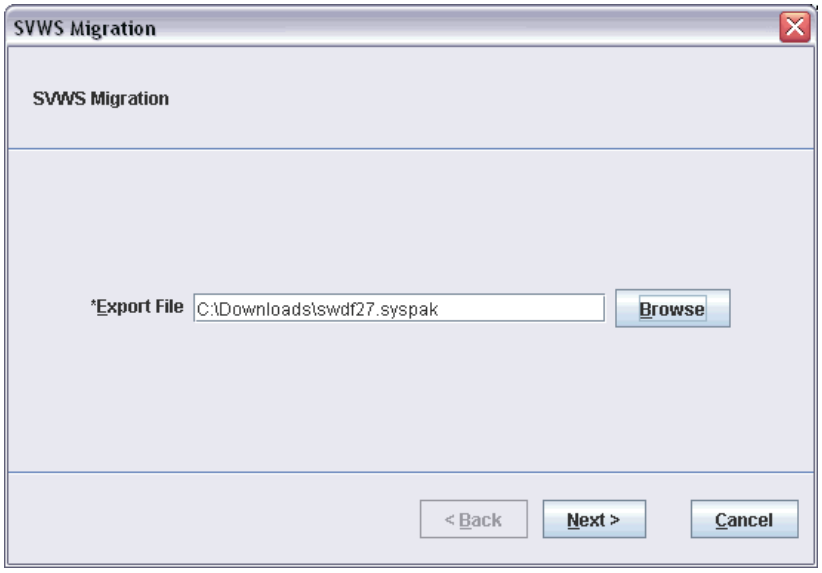
   or

   (UNIX)
   <PASW Collaboration and Deployment Services Installation Directory>/components/svws-migration/runMigrationGui.sh

   The SVWS Migration wizard opens.

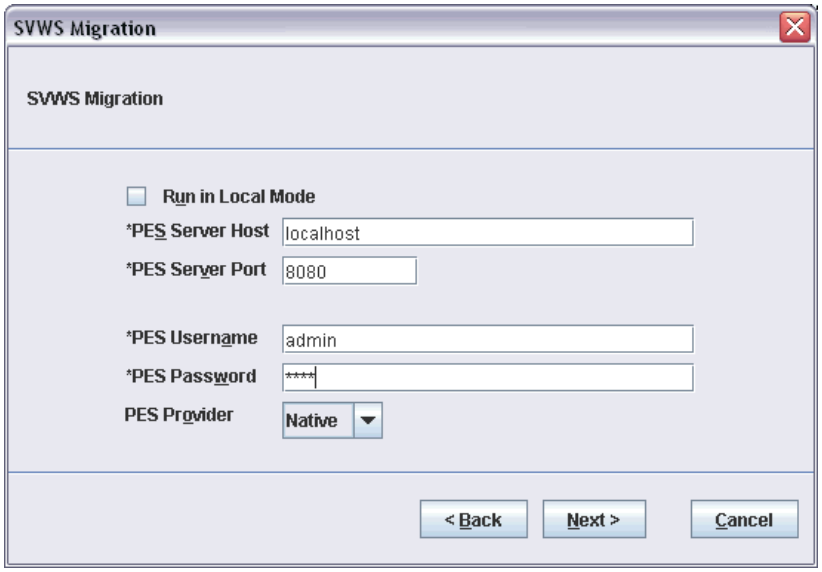
2. Select the export file, and click Next.

Figure 12-1
Export file selection dialog box of the SVWS Migration wizard



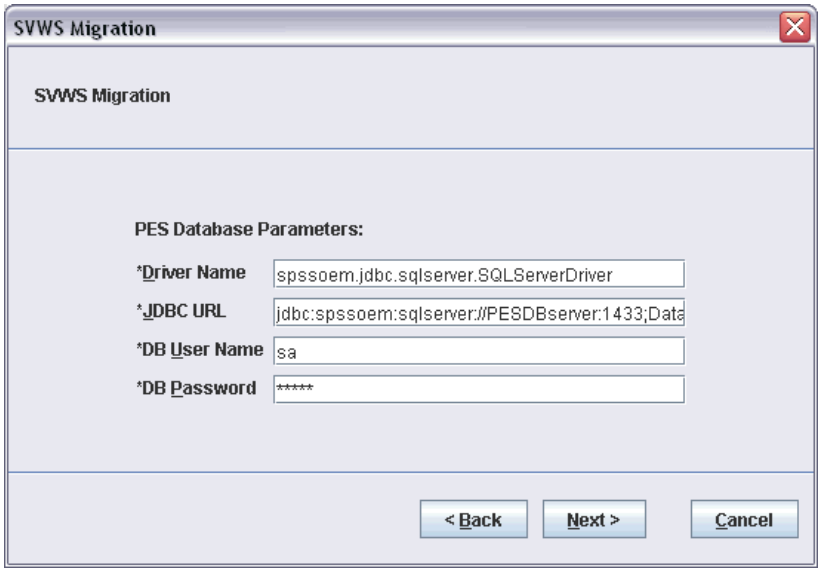
3. Specify the address and port of the PASW Collaboration and Deployment Services host, user credentials, and security (authentication) provider, and click Next.

Figure 12-2
PASW Collaboration and Deployment Services information dialog box of the SVWS Migration wizard



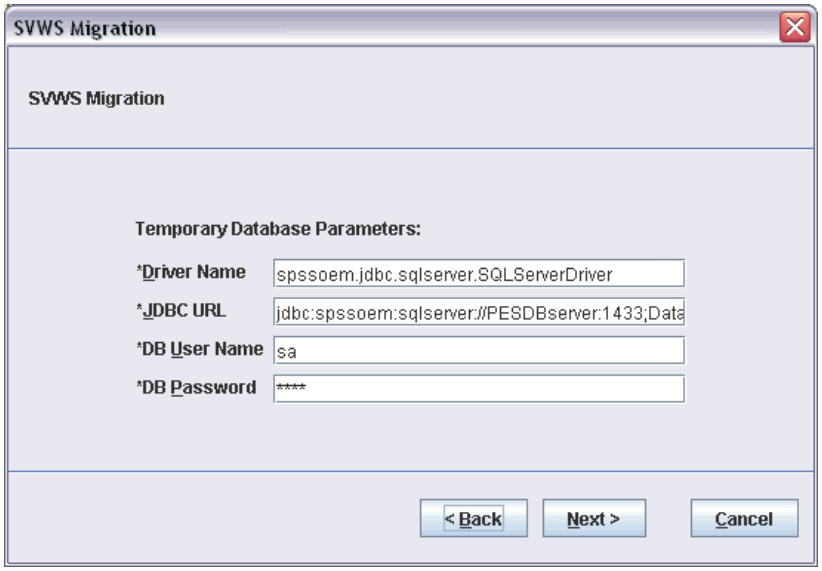
4. Specify repository database connection information, and click Next.

Figure 12-3
repository database connection dialog box of the SVWS Migration wizard



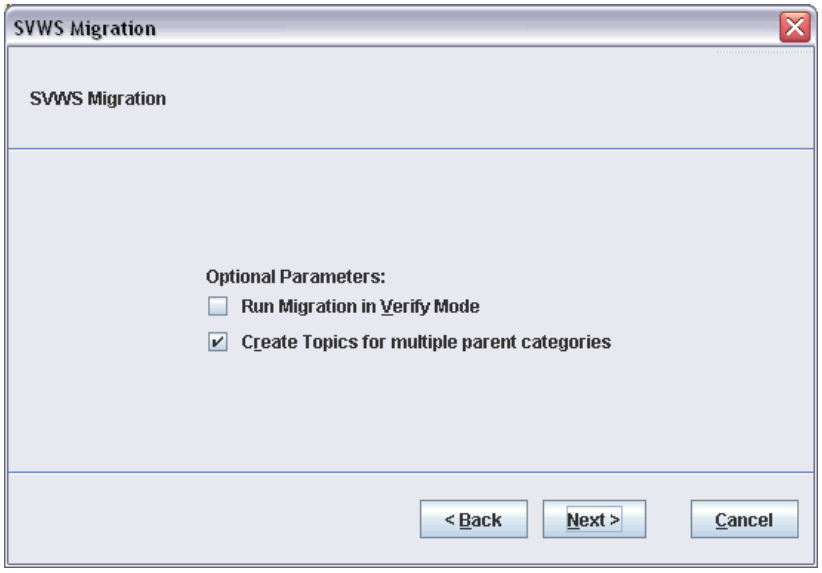
5. Specify the temporary database connection information, and click Next.

Figure 12-4
Temporary database connection dialog box of the SVWS Migration wizard



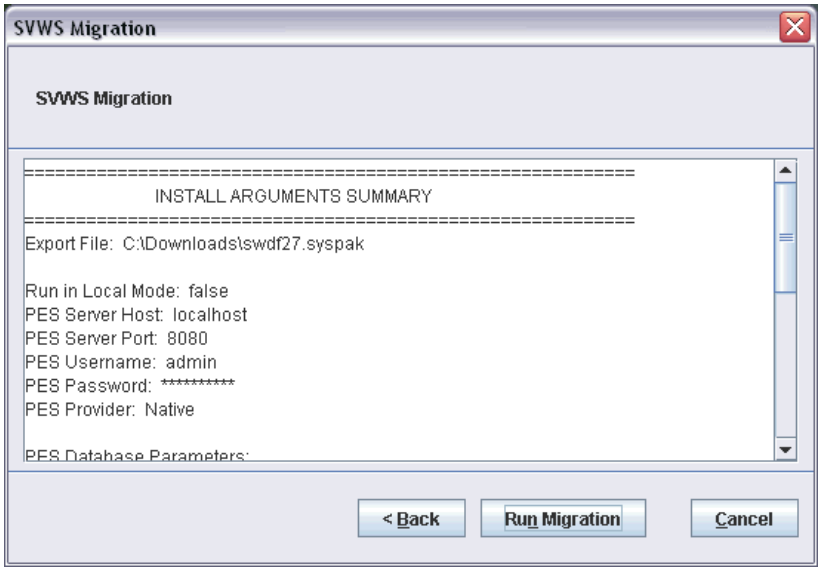
6. Specify whether migration is to be run in the verification mode and if topics should be created for parent SVWS categories, and click Next.

Figure 12-5
Optional parameters dialog box of the SVWS Migration wizard



7. Review the specified parameter summary, and click Run Migration.

Figure 12-6
Argument summary dialog box of the SVWS Migration wizard



To run migration in command line mode:

- Execute

(Windows)
<PASW Collaboration and Deployment Services Installation Directory>/components/svws-migration/runMigration.bat

or

(UNIX, IBM i)
<PASW Collaboration and Deployment Services Installation Directory>/components/svws-migration/runMigration.sh

with the following command line parameters:

Table 12-1
General parameters

-pesDir	The path of the PASW Collaboration and Deployment Services installation directory.
-file	The path of the SWDF input file.
-installDir	The path of the directory where the contents of the export file would be stored.
-verify	The option to run the process in verification mode. When true is specified, actual contents are not migrated to the repository database.
-createTopics	The option to create topics for multiple parent categories. The topics will not be created unless the value of true is specified.

Table 12-2
PASW Collaboration and Deployment Services parameters

-host	The name or the IP address of the PASW Collaboration and Deployment Services host.
-port	The port number for connecting to PASW Collaboration and Deployment Services.
-pesUser	PASW Collaboration and Deployment Services username.
-pesPassword	PASW Collaboration and Deployment Services password.
-pesProvider	PASW Collaboration and Deployment Services security provider information.
-local	The option specifies whether Web service calls made to PASW Collaboration and Deployment Serviceswill be local.

Table 12-3
repository database parameters

-pesDbDriver	JDBC driver to connect to the repository database.
-pesDbUrl	JDBC URL of the repository database.
-pesDbUser	repository database username.
-pesDbPassword	repository database password.

Table 12-4
Temporary database parameters

-driverName	JDBC driver to connect to the temporary database.
-url	JDBC URL of the temporary database.
-user	Temporary database username.
-password	Temporary database password.

Important! Do not discard the contents of the temporary database until you have verified that SWDF data has been successfully migrated to the repository.

Appendix

A

Troubleshooting

Certain error messages and symptoms are common when installing and working with PASW Collaboration and Deployment Services. Methods for clearing these errors and establishing a functional system exist for:

- **PASW Collaboration and Deployment Services**. Common problems when installing and starting the application on supported server platforms.
- **Solaris 9**. Known issues related to PASW Collaboration and Deployment Services on Sun's UNIX operating system.
- **DB2 for IBM I**. Symptoms and error messages that surface while transacting with a DB2 database running on IBM i.
- **Oracle 9i and 10g**. Symptoms and error messages that surface while transacting with an Oracle 9i and 10g databases.
- **JBoss**. JBoss application server running PASW Collaboration and Deployment Services.
- **Oracle 10g AS**. Oracle 10g application server running PASW Collaboration and Deployment Services.
- **WebLogic**. WebLogic application server running PASW Collaboration and Deployment Services.
- **WebSphere**. WebSphere application server running PASW Collaboration and Deployment Services.

It is always a good practice to refer to PASW Collaboration and Deployment Services log files to establish the cause of the problem. For more information, see Logging Services in Chapter 10 on p. 77.

PASW Collaboration and Deployment Services

The installation completed without reporting errors, but when I enter the IP address and port number into a browser, the page does not load.

Once the PASW Collaboration and Deployment Services installation is complete, you must run the startup script or batch file to launch the server before it can be accessed using a browser. After running the startup script, change the current working directory to:

<installation_path>/JBoss/bin

Run the PASW Collaboration and Deployment Services status tool. The *wrapper* process indicates if the application is currently running.

Note: The status tool is available only for Solaris in this release.

How do I prevent performance bottlenecks and CPU usage issues when starting and deploying PASW Collaboration and Deployment Services?

Depending on the specific system configuration, previously installed antivirus or spyware software may be configured for "deep scanning" of application components. These third party applications can be reconfigured to scan during certain times, or they can be turned off during installation and manually restarted.

Additionally, some of the more strict server-side firewall settings may negatively impact startup performance and not allow access.

If you are experiencing significant system degradation when starting the service, disable any nonessential processes and restart the PASW Collaboration and Deployment Services.

Once I log in to the administrative interface, how do I determine which database I am accessing?

Database connection information can be downloaded and accessed from the Web interface.

1. After authenticating, click About from the navigation list options. The About page appears.
2. Click the Download version and system details link at the bottom of the page. When prompted, save the file to disk.
3. Open the file in a text editor and search for *Database Details*. This section contains detailed information on the database being used, including name, version, and a table listing.

The application throws java.lang.OutOfMemoryError: PermGen space exception.

This error occurs when the JVM runs out of space in the permanent generation heap due to a large number of used classes. The solution is to increase the value specified with PermSize JVM parameter. For example, for JBoss installations, the size of permanent generation heap available to the wrapper service can be increased by modifying the following line in *<JBoss Installation Directory>/wrapper/conf/wrapper.conf*:

```
wrapper.java.additional.1=-Dprogram.name=run.bat -XX:PermSize=128m.
```

For information about increasing the permanent generation heap size for other application servers, see the application server vendor documentation.

When a BIRT report is run in Deployment Portal, the application is not able to authenticate my credential for accessing the data source of the report and is repeatedly displaying the login screen.

- Verify that the data source for the report and the credentials are defined correctly. For more information, see the corresponding section of the *Deployment Manager User's Guide*.
- If the data source for the report is JDBC-based, verify that the proper driver is installed with repository. For driver path information specific to the operating platform, see the installation instructions.

SAS syntax job processed in PASW Collaboration and Deployment Services running on a UNIX system fails with to a database connection error due to invalid library name ("ERROR: Error in the LIBNAME statement").

- Verify that the shared libraries path environment variable (LD_LIBRARY_PATH on Solaris, SHLIB_PATH on HP-UX, or LIBPATH on AIX) is set to an appropriate value.

How do I restore PASW Collaboration and Deployment Services if my keystore file has been lost?

The keystore file contains the keys used to encrypt passwords used by PASW Collaboration and Deployment Services, such as the master password for database access. If the keystore file is lost, the system becomes unusable. If backup of the keystore is available, it can be restored to the original location. If you are unsure what the original path of the keystore was, you can look up the *keystorePath* property of *keystoreSecurity* element in *<PASW Collaboration and Deployment Services Installation Directory>/platform/setupinfo.xml*.

If the keystore file is lost and backup is not available, the system must be reinstalled by re-running the setup utility in *<PASW Collaboration and Deployment Services Installation Directory>/setup* and pointing it to the existing repository database. All passwords that existed in the system, such as the passwords for external directory services, defined credentials, etc. must be manually reentered.

A BIRT report against DB2 IBM i V6R1 database using prompted credentials fails when run in PASW Collaboration and Deployment Services .

Add prompt=true parameter to the JDBC connection URL.

Driver Name: com.ibm.as400.access.AS400JDBCDriver
Driver URL: jdbc:as400://mySerever/B101E31E;prompt=false

"Build New Scoring Configuration Details Failed" error when configuring scoring on non-Windows PASW Collaboration and Deployment Services installations

"Build New Scoring Configuration Details Failed" error message is displayed when scoring configuration dialogue is opened in Deployment Manager. The problem is corrected by changing the permissions on *<PASW Collaboration and Deployment Services installation directory>/components/modeler/modelerserver* file to execute, for example:

```
cd /usr/PASWCDS4/components/modeler/modelerserver
sudo chmod +x modelerserver
```

Solaris

How do I avoid getting an access error message when trying to run the installation script?

PASW Collaboration and Deployment Services must be installed by a user with adequate privileges. Change the active user to *root* (or to another user with adequate access rights) and run the installation script.

To which other directories does the installing user need access?

The user running the installation must also have write access to */etc/.java* for the system to function properly.

If the installation is executed by a user without write access to */etc/.java*, switch to a user with write access and run the setup shell script again. Once the installation is complete, verify that the following file exists:

/etc/.java/.systemPrefs/com/spss/setup/component/services/prefs.xml

Unable to start PASW Collaboration and Deployment Services on JBoss and Solaris 9.

When attempting to start PASW Collaboration and Deployment Services on JBoss and Solaris 9, *"ld.so.1: wrapper: fatal: libm.so.2: open failed: No such file.."* error occurs.

To resolve the problem, create symbolic link */usr/lib/64/libm.so.2* to */usr/lib/64/libm.so.1*:

```
ln -s /usr/lib/64/libm.so.1 /usr/lib/64/libm.so.2
```

Oracle 9i

How do I create a user and tablespace?

To clear and reestablish the *spssplat* user and tablespace from an Oracle database, issue the following set of commands:

```
drop user spssplat cascade; CREATE USER spssplat IDENTIFIED BY spssplat
DEFAULT TABLESPACE SPSSPLAT TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON SPSSPLAT;
@$ORACLE_HOME/sqlplus/admin/pupbld;
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO spssplat;
```

JBoss

How is the session timeout value configured to adjust the amount of time a user can remain idle?

Once a user is logged in to PASW Collaboration and Deployment Services, a period of inactivity is allowed before the session is terminated and the user must reauthenticate. To increase or decrease this value:

1. From the installation directory, navigate to *\JBoss\server\default\deploy\jbossweb-tomcat50.sar*.
2. Open *web.xml* in a text editor.
3. Locate the section for *Default Session Configuration*, and edit the value for <session-timeout>.
4. Stop and restart the application.

   Note: This file is processed when the application is deployed; configuration changes do not take effect until the server is restarted.

How do I determine the port on which my version of JBoss is running?

The JBoss application server's HTTP port is defined in the file:

jboss-3.2.7\server\default\deploy\jbossweb-tomcat50.sar\server.xml

with the attribute:

/Server/Service/Connector@port

Note: Depending on the release of JBoss, the version numbers in the path may vary.

What additional settings are required for PASW Collaboration and Deployment Services FIPS 140-2 compliance on JBoss?

For PASW Collaboration and Deployment Services to function properly when running on JBoss in FIPS 140-2-compliant mode, {URIEncoding="UTF-8"} attribute must be specified for the HTTPS connector.

Alternatively, from the command line, the netstat command can be used to view applications and the ports that are in use.

Oracle 10g AS

Message-based processing is not being triggered in PASW Collaboration and Deployment Services running on Oracle 10g application server.

Setting up message-based job processing on Oracle 10g to enable durable subscriptions requires *clientID* property of the *ConnectionFactory* JMS configuration attribute to be specified using the application server administration console.

WebLogic

"IOException: Resource has been deleted" is thrown in Deployment Portal when trying to access file attachments that contain reporting output.

The exception can occur if the PASW Collaboration and Deployment Services installation is running on WebLogic application server using JRockit rather than Sun JRE. If the exception occurs, reconfigure WebLogic to use Sun JRE. For more information, see WebLogic documentation.

Cascading parameters are not displayed correctly in reports when PASW Collaboration and Deployment Services is run with WebLogic 9.2 and 10 on Solaris 10.

-Djava.awt.headless=true startup argument must be added to the application server Java environment.

WebSphere

Miscellaneous errors occur during package installation (with Package Manager) into the repository using a WebSphere application server.

Make sure the latest vendor patches have been applied to the application server.

Server log is reporting encryption errors, such as exception com.ibm.crypto.provider.AESCipher.engineGetKeySize(Unknown Source)

The error occurs with WebSphere 6.1 Service Pack 19 and is caused by the incorrect password value. To correct the error, copy the value of platform.keystore.password from

<PASW Collaboration and Deployment Services installation directory>/platform/setupinfo.xml

to

<WEBSPHERE_HOME>/profiles/AppSrv01/config/cells/xi-wyueNode01Cell/nodes/xi-wyueNode01/servers/<server name>/server.xml

Upgrading to WebSphere 6.1 Service Pack 23 may also resolve encryption problems.

Appendix

B

Nativestore Schema Reference

The *nativestore.xsd* schema defines the structure of an XML file containing users and groups to be imported into the Deployment Manager. In addition, the file can specify obsolete users and groups that should be deleted.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
 <user userID="sbennett" password="sb9482" encrypted="false">
  <group>sales</group>
 </user>
 <user userID="lsanborn" password="ls7725" encrypted="false">
  <group>sales</group>
 </user>
 <user userID="lalger" password="la4011" encrypted="false">
  <group>analyst</group>
 </user>
 <user userID="cjones" password="cj2683" encrypted="false">
  <group>analyst</group>
 </user>
 <obsolete>
  <user>mmonroe</user>
  <user>bgmurphy</user>
  <group>jones project</group>
 </obsolete>
</nativestore>
```

nativestore Element

Child elements: user, obsolete

Root element for importing local users and their groups into the Deployment Manager.

user Element

Parent element: nativestore

Child elements: group, role

User to be added or updated.

Table B-1
Attributes for the user element

Name	Type	Use	Default	Description
userID	string	required	*no default value*	User ID that will be used to log in to the system.
password	string	optional	*no default value*	Usually a plain-text password. If the encrypted attribute is true, then this password is encrypted. It is generally not practical to use an encrypted password when importing. Passwords are encrypted when exporting from the server, but this is *not* exposed in the Deployment Manager user interface.
encrypted	boolean	optional	false	Indicates if the password is plain-text or encrypted. Encrypted passwords are exported from the native store (encryption is one-way, making it impossible to re-create a user's password). When importing from another system, passwords must be plain-text; the encrypted attribute is usually omitted.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
 <user userID="sbennett" password="sb9482" encrypted="false">
  <group>sales</group>
 </user>
</nativestore>
```

group Element

Type: string

Parent element: user

Groups associated with the user. If a group does not exist, it will be created automatically.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
 <user userID="sbennett" password="sb9482" encrypted="false">
  <group>sales</group>
 </user>
</nativestore>
```

role Element

Type: string

Parent element: user

Role associated with the user. If a role does not exist, it will *not* be added automatically.

obsolete Element

Parent element: nativestore

Child elements: user, group

Groups or users to be removed. Note that they may be loaded in "replace mode," which will automatically remove all groups and non-administrative users. In that mode, this element has no effect.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
 <obsolete>
  <user>mmonroe</user>
  <user>bgmurphy</user>
  <group>jones project</group>
 </obsolete>
</nativestore>
```

user Element

Type: string

Parent element: obsolete

The user ID to be removed. A user with administrative privileges cannot be removed.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
 <obsolete>
  <user>mmonroe</user>
 </obsolete>
</nativestore>
```

group Element

Type: string

Parent element: obsolete

Group name to be removed.

Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<nativestore>
  <obsolete>
    <group>jones project</group>
  </obsolete>
</nativestore>
```

Appendix

C

Oracle Database Configuration

When using an Oracle 10g or 11g database in conjunction with PASW Collaboration and Deployment Services, the following parameters and configurations must be followed. Changes are made to the *init.ora* and *spfile.ora* parameter files.

Table C-1
Oracle Database Parameters

Parameter	Setting
OPEN_CURSORS	150
NLS_CHARACTERSET	AL32UTF8
NLS_NCHAR_CHARACTERSET	AL16UTF16

Note: Both NLS_CHARACTERSET and NLS_NCHAR_CHARACTERSET should be set when creating the Oracle instance.

Appendix

D

Enabling Windows 64–bit Support

Enabling Windows 64-bit Support

Installing PASW Collaboration and Deployment Services4 on a 64-bit Windows system involves steps that are common to all platforms as well as steps specific to the application server. The following steps must be performed regardless of what application server is being used:

1. Install PASW Collaboration and Deployment Services4 into a 64-bit version of the application server. For more information, see Installing PASW Collaboration and Deployment Services in Chapter3 on p. 7.
2. Install the Microsoft Visual C++ Redistributable Package. The package can be found at *http://www.microsoft.com/downloads/details.aspx?familyid=90548130-4468-4BBC-9673-D6ACABD5D13B*. The package is used to execute applications developed in Microsoft Visual C++ on 64-bit platforms.
3. Replace the existing JRE in the */jre* directory of the application server installation path with a 64-bit JRE.
4. Continue with steps specific to your application server.

JBoss

The following changes must be made to the default PASW Collaboration and Deployment Services setup.

File Name	Find	Replace With
<PASW Collaboration and Deployment Services Installation Directory>/startserver.bat	mm console	set MM_INSTALL_DIR=%~dp0% mm64

Important! After completing these steps, it will no longer be possible to start PASW Collaboration and Deployment Services as a Windows service. Instead *startserver.bat* must be used. If it is necessary to configure PASW Collaboration and Deployment Services to start as a Windows service, see JBoss documentation (*http://www.jboss.org/community/docs/DOC-10679*).

WebLogic

The following changes must be made to the application server setup.

File Name	Find	Replace With
<WebLogic domain>/startWeblogic.cmd	windows	windows64
<WebLogic domain>/bin/setDomainEnv.cmd	set SUN_JAVA_HOME=	set SUN_JAVA_HOME= <64-bit JVM Java Home path>

Note: It may be necessary to set DOMAIN_PRODUCTION_MODE=true in the *<WebLogic domain>/startWeblogic.cmd* script if your 64-bit JVM fails to start.

WebSphere

1. Start WebSphere Application Server and login to the WebSphere administrator console.
2. Navigate to Environment -> Shared Libraries -> SPSSSharedLibrary
3. Change the native path from ${SPSSPLATFORM_DIR}/components/setup/jni/windows to ${SPSSPLATFORM_DIR}/components/setup/jni/windows64.
4. Navigate to Application servers -> Java and Process Management -> Process Definition -> Environment Entries -> PATH.
5. Change windows to windows64 in the value of the PATH.
6. Navigate to Application servers -> Java and Process Management -> Process Definition -> Java Virtual Machine -> Custom Properties -> java.library.path.
7. Change windows to windows64 in the value of the java.library.path.
8. Save the configuration and restart the server.

Oracle AS

The following changes must be made to the application server setup.

File Name	Find	Replace With
<Oracle installation directory>/opmn/conf/opmn.xml	windows	windows64 (only for the configuration for your particular OC4J instance)

Note: If the application server is not already configured to use a 64-bit JVM, it may be necessary to set your OC4J instance to use 64-bit JVM. To do this, add the java-bin option to *start-parameters* element for your particular instance in *<Oracle installation directory>/opmn/conf/opmn.xml*.

The following is an example of *<Oracle installation directory>/opmn/conf/opmn.xml* file of OC4J instance *PRH35* configured for Windows 64-bit:

```
<process-type id="PRH35" module-id="OC4J" status="enabled">
    <environment>
              <variable id="PATH" value="E:/Program Files/SPSSInc/Enterprise Repository/components/
      setup/jni/windows64;%PATH%"/>
    </environment>
      <module-data>
          <category id="start-parameters">
            <data id="java-bin" value="C:/Program Files/Java/jdk1.5.0_14/bin/java"/>
            <data id="java-options" value=" -Xmx512m -Xms128m
            -Djava.security.policy=$ORACLE_HOME/j2ee/PRH35/config/java2.policy
            -Djava.awt.headless=true -Dhttp.webdir.enable=false
            -XX:PermSize=512m -Djava.compiler=NONE
            -Doc4j.jmx.security.proxy.off=true
            -Dcom.spss.psapi.extensions.autoloadDirectory=
             &amp;quot;E:/Program Files/SPSSInc/Enterprise Repository/components/cr-adapter/clementine/ext/lib&amp;quot;
            -Djava.library.path=&amp;quot;E:/Program Files/SPSSInc/Enterprise Repository/components/setup/jni/windows64&amp;quot;"/>
            <data id="oc4j-options" value="-userThreads"/>
          </category>
          <category id="stop-parameters">
            <data id="java-options" value=
           "-Djava.security.policy=$ORACLE_HOME/j2ee/PRH35/config/java2.policy
            -Djava.awt.headless=true -Dhttp.webdir.enable=false"/>
          </category>
    </module-data>
    <start retry="2" timeout="600"/>
    <stop timeout="120"/>
    <restart retry="2" timeout="720"/>
    <port id="default-web-site" protocol="http" range="8080"/>
    <port id="rmi" range="12401-12500"/>
    <port id="rmis" range="12701-12800"/>
    <port id="jms" range="12601-12700"/>
    <process-set id="default_group" numprocs="1"/>
</process-type>
```

Appendix

E

SAP NetWeaver Configuration Notes

Additional configuration steps are required after the repository is installed and licensed on a system running SAP NetWeaver application server. They include modifying the system *PATH* variable and adding custom parameters to SAP NetWeaver configuration. For information about installing and licensing the repository, see Installing the Repository on p. 14

System Path Variable

- The host system *PATH* environment variable must be modified to include the directory containing OS-specific PASW Collaboration and Deployment Services licensing libraries; for Windows, it is *spsslic.dll*. The initial installation location of the libraries is *<PASW Collaboration and Deployment Services installation directory>/components/setup/jni/<operating system>*.
- *PATH* must also be modified to include all the paths enumerated in the [INCLUDE_PATHS] section of the generated properties file *<PASW Collaboration and Deployment Services installation directory>/setup/resources/netweaver/config/environment.properties*.

```
...
[INCLUDE_PATHS]
E:/PASW_CADS_4/components\modeler\bin
E:/PASW_CADS_4/setup/resources/netweaver/bin/psapi.rar
E:/PASW_CADS_4/setup/resources/netweaver/bin/csp.rar
...
```

Custom Parameters

- The following custom parameters must be defined with Application Server Configuration Tool.

Configtool
  VM Parameter
    System

- **com.spss.psapi.session.serverInstallationDirectory**
- **com.spss.psapi.extensions.autoloadDirectory**
- **csp.installationDirectory**

The parameters and the corresponding values are listed in the [JAVA_PROPERTIES] section of the generated properties file *<PASW Collaboration and Deployment Services installation directory>/setup/resources/netweaver/config/environment.properties*.

```
...
[JAVA_PROPERTIES]
com.spss.psapi.session.serverInstallationDirectory=E:/PASW_CADS_4/components/modeler
com.spss.psapi.extensions.autoloadDirectory=E:/PASW_CADS_4/components/modeler/ext/lib
```

```
csp.installationDirectory=E:/PASW_CADS_4/components/modeler
...
```

- Keystore path and password used during the repository installation must also be added as custom parameters.
  - **platform.keystore.file** The path of PASW Collaboration and Deployment Services keystore file.
  - **platform.keystore.password** Keystore password.
- Java Preferences Factory custom parameter must also be added.

  java.util.prefs.PreferencesFactory The parameter value must be set to java.util.prefs.WindowsPreferencesFactory.
- The SAP NetWeaver instance must be restarted for the parameters to take effect.

Index