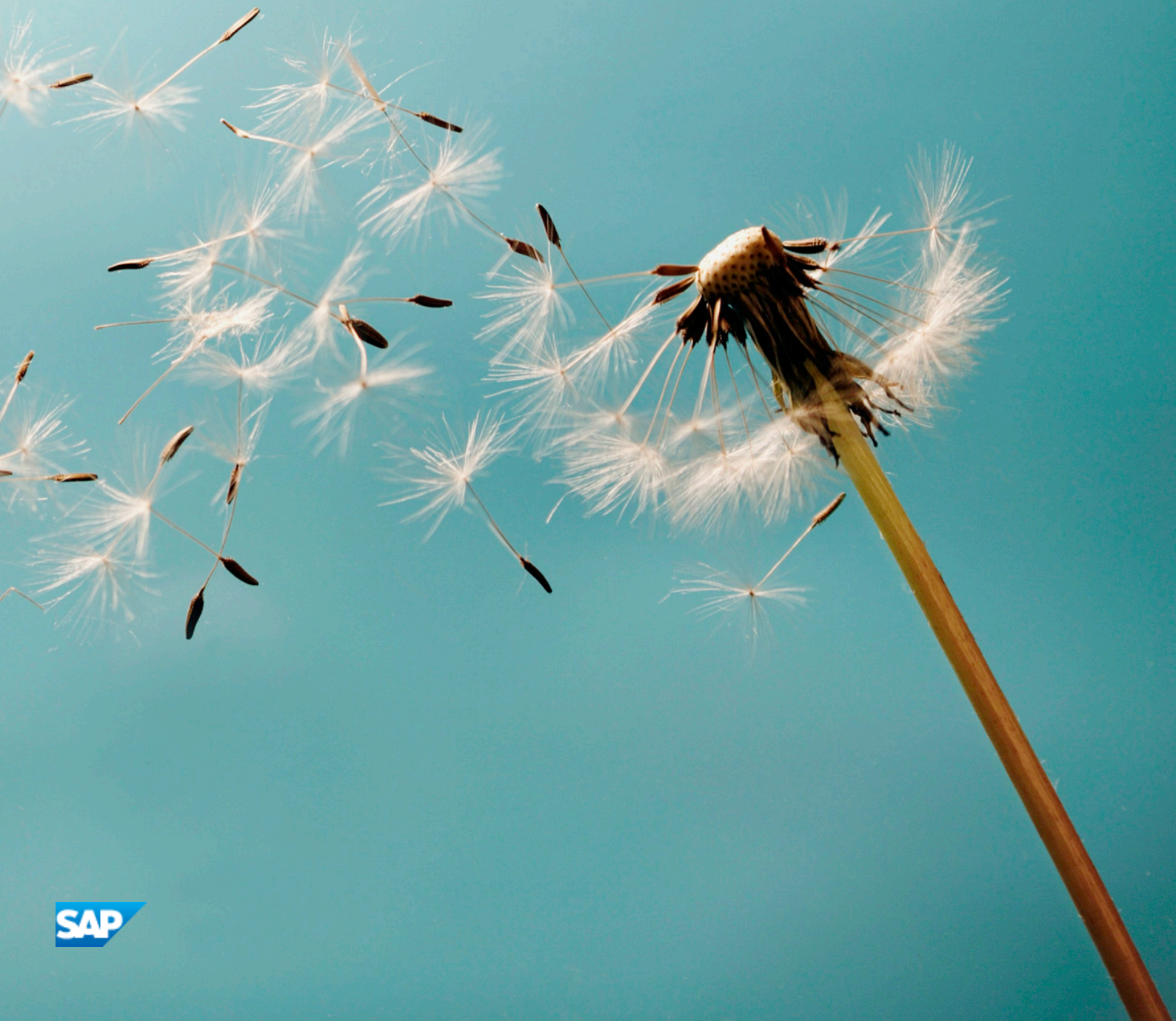
Administrator's Guide
Document version: 1.2 – 2015-05-21

CUSTOMER

SAP Strategy Management Interactive Publisher and Application Components 10.1



Document History

Caution

Before you start the implementation, make sure you have the latest version of this document. You can find the latest version on the SAP Help Portal at help.sap.com/bosm101.

The following table provides an overview of the most important document changes.

Table 1

Version	Date	Description
1.0	2013-10-01	Initial version
1.1	2015-05-04	Minor editorial correction to the topic "High Availability".
1.2	2015-05-21	Complete rewrite of the High Availability [page 34] chapter.

Content

1 Getting Started

The Java-based Interactive Publisher component of SAP Strategy Management provides connection pool, caching, and load balancing management for the Application Server component. These services are provided for the strategy management application business logic.

The application provides Web-based strategy management and is deployed on SAP NetWeaver.

About this Guide

This guide provides a starting point for managing and maintaining Interactive Publisher and the application components. It contains specific information for various tasks and lists the tools that you can use to implement them.

1.1 Important SAP Notes

> **Caution**
> Check regularly to see which SAP Notes are available for SAP Strategy Management.

Important SAP Notes

Table 2

SAP Note Number	Title
1905391	Central Note: SAP Strategy Management 10.1

1.2 Naming Conventions

In this documentation, the following naming conventions apply:

Table 3

Variables	Description
`<nw_server>:<port>`	Server name or IP address and port where SAP NetWeaver is installed and strategy management application components are deployed

2 Technical System Landscape

The following table lists more information about the technical system landscape available on the SAP Help Portal at http://help.sap.com/bosm101.

Table 4

Topic	Guide or Tool
System Landscape	*Master Guide for SAP Strategy Management*
Initial Configuration	*Installation Guide for SAP Strategy Management*
Post-Installation Configuration	*Server Configuration Help for SAP Strategy Management*
Security	*Security Guide for SAP Strategy Management*
Administration	*Administration* topic in the SAP Strategy Management application help

3 Monitoring of Interactive Publisher and Application Components

3.1 Trace and Log Files

The application has the following trace and log files:

Table 5

Component	Content	File	Path
Persistence Service	-	`defaultTrace.trc`	\usr\sap\<instance_DI>\J* \j2ee\cluster\server?\log
Print Service on NW	-	`defaultTrace.trc`	\usr\sap\<instance_DI>\J* \j2ee\cluster\server?\log

Configuring NetWeaver Java logs for the Java Persistence Service and the Print Service on NetWeaver:

You can change the severity of the Java Persistence Service logs using the NetWeaver log configuration.

1. Go to *http://<host>:<port>/nwa* > *Troubleshooting* > *Logs and Traces* > *Log Configuration.*
2. Set *Show* to *Tracing Locations*.
3. Set *Location* to *com.sap.poa.sbc.bui.persistence.*

   The logs are generated in the NetWeaver `defaultTrace.trc` file.

Enable SAP Logging

By default, the NetWeaver logging for the above functions goes to standard error. To have finer grained control, start the NetWeaver instance with the following Java system property: –
`Djava.util.logging.manager=com.logging.bridge.util.SapLogManager`

Set Java Virtual Machine (JVM) Property

1. Start AS Java Config Tool.
2. Go to *cluster-data* > *template* > *instance*.
3. Choose *VM Parameters* tab.
4. Choose *System* tab.
5. Choose *New*.
6. Add the name `java.util.logging.manager`
7. Add the value `com.sap.logging.bridge.util.SapLogManager`
8. Choose *File* > *Apply Changes*.
9. Restart NetWeaver.

Create a Log Destination

1. Start AS Java Config Tool.
2. Go to *cluster-data* > *template* > *instance* > *log configuration* > *destinations* > *applications_log*.
3. Choose *New*.
4. Enter the destination name, for example, `bui_trc`

5. Alter the pattern as required, for example, `./log/bui_trc_%g.log`
6. Choose *cluster-data* > *template* > *instance* > *log configuration* > *locations* > *Root Location* > *com* > *sap* > *poa*

> **Note**
> To be able to see the location in the Config Tool, you may first need to use the NetWeaver Administrator application to set the location level logging to a level other than ERROR.

7. Choose *Add*.
8. Select the previously created destination.
9. Choose *File* > *Apply Changes*.
10. Restart NetWeaver.

JPIP uses the SAP NetWeaver Log Viewer for tracing and logging.

3.1.1 Strategy Management Logging and Tracing

The strategy management application components that are deployed on SAP NetWeaver use the standard logging and tracing facilities in SAP NetWeaver Administrator.

Process

1. Set the log severity. For more information, see Setting the Log Severity [page 10].
2. Enable the trace. For information about enabling tracing, see Enabling the Log for the Interactive Publisher Worker Thread [page 11]
3. Run the application and check the trace. For more information, see Running the Application and Checking the Trace [page 11].

3.1.1.1 Setting the Log Severity

The default severity level for traces is *ERROR*. Traces can be dynamically switched on.

`PwRequest_httplog` is a location under `http` trace. You can set this parameter to `debug` to get a trace of queries in the application. The trace goes to the default location of `\log\defaultTrace_00.trc`.

For information about Log Viewer, see the SAP NetWeaver Administrator documentation.

Table 6

Location	Category
`/System/Server` `/System/Server/WebRequests` displays error or warning messages when processing Web requests.	`CATEGORY_SYSTEM_SERVER`
`/System/Database`	`CATEGORY_SYSTEM_DATABASE`
`/System/Network`	`CATEGORY_SYSTEM_NETWORK`

Location	Category
`/System/Security` `/System/Security/Authentication` logs the status of a logon attempt as either OK or failed, identifies the username attempting to log on, and the authentication stack.	`CATEGORY_APPLICATION`
`/Application`	`CATEGORY_APPLICATION`

Procedure

1. Log onto SAP NetWeaver as the administrator.
2. Go to *Troubleshooting* > *Logs and Traces* > *Log Configuration*.
3. From the *Show* dropdown list, choose *Tracing Locations*.
4. Navigate to `Root Location > com > sap > cpm> sm > jpip > PipTrace`.
5. Set the *Severity* to *Info* and save the configuration.

3.1.1.2 Enabling the Log for the Interactive Publisher Worker Thread

Prerequisites

You set the *Piptrace* severity to *Info* in SAP NetWeaver Administrator. For information, see .

Procedure

1. Start the *Tools* utility by typing this URL in a browser window:
   `http://<nw_server>:<port>/strategy/tools`
2. Select *Application Tracing*, and then select *Enable* for the *PIP Trace* setting.
3. Select the *Back* function on the Internet Explorer browser window to return to the *Tools* utility.
4. Select *JPIP Session Monitor* and then select *Restart JPIP*. This restarts JPIP to ensure that a new session is established with tracing enabled.
   For information about starting the JPIP Session Monitor, see Java (JPIP) Session Monitor. [page 15]

3.1.1.3 Running the Application and Checking the Trace

You can monitor the following locations: *Auth*, *Cache*, *Chart*, *Common*, *Dao*, *Dba*, *Ejb*, *http*, *jsp*. If there is a program exception, a log entry is created.

> **Note**
>
> When you are finished logging, make sure you disable the logging using the *Disable* setting in Tools

Procedure

1. In SAP NetWeaver Administrator, go to *Troubleshooting* > *Logs and Traces* > *LogViewer*.
2. Choose *Show View* > *Open View* > *System Logs and Traces*.

3.1.2 Logging of Application Configuration Settings

You can view the audit log of certain areas of the administration application. Whenever you add, modify, or rename entries in the following sections of the administration application, the settings are logged:

- *Manage Models*
- *Set Defaults*
- *Update User Responsibilities*
- *Delete Obsolete Items*

If you delete something, it is not included in the audit log.

The *AuditLogComments* Java System Property in the strategy management application in SAP NetWeaver causes a prompt to appear every time a strategy management administrator adds, modifies, or renames entries in one of the screens listed above. If the strategy management administrator enters a comment when prompted, the comment appears in the log next to the log activity. If this property is set to *No*, the strategy management administrator is not prompted to add comments and no comments appear in the log.

For information about setting the *AuditLogComments* Java System Property, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Configuring the Application Properties*.

Procedure

1. Open a browser and issue this URL in the *Address* box:

   `http://<nw_server>:<port>/strategy/tools`
2. Click *Auditor*.
3. From the *Application Module Name* dropdown list, select the item you want to audit and click *Search*.

   Audit details about the selection appear in the *Auditing Report* window. You can review detailed information by clicking each item.

   This table shows the fields and descriptions in the *Auditing Report*.

   Table 7

Field	Description
ID	Unique ID for auditing information

Field	Description
Application Name	Name of the administration application screen you are auditing.
Event	The event that triggered the auditing.
Description	Details of the event (parameters set, default settings, changed to settings)
Database ID	Database ID
User Name	Name of user logged in. The value represents the long name specified in SAP NetWeaver UME rather than the logon ID.
Created Date	Date and time the event occurred.
Last Updated Date	Date and time the setting was last changed.

3.2 SQL Auditing

The `SqlAuditingType` property in the global properties file controls whether to audit SQL statements.

Table 8

Property	Syntax	Description
`SqlAuditingType`	`SqlAuditingType= 0 | 1 | 2`	The database table `CPMS_SQL_AUDIT` is used for SQL auditing. If a statement gets executed from the strategy management application and it meets the set criteria for logging, then it gets logged in said table. `0`. Do not audit SQL statements `1`. Audit Insert, Update, and Delete statements `2`. Audit Insert, Update, Delete, Select statements

3.3 Activity and Resource Monitors

We recommend using resource consumption utilities such as the Microsoft Task Manager utility.

3.3.1 Application Component and Communication Monitors

3.3.1.1 Using CA Wily Introscope and Fiddler

For performance and resources utilization statistics, Application Components is instrumented for Wily's Introscope.

The following Application Component information is available in the Introscope:

- Peak memory consumption in megabytes (MB)
- CPU consumption in milliseconds

The following communication information is available in Fiddler:

- Number of round trips between front-end and application
- Data volume, in kilobytes (KB), transferred between front-end and application

For performance metrics monitoring, see Wily Introscope installation guide on SAP Service Marketplace at service.sap.com/instguides > *Installation & Upgrade Guides* > *SAP Components* > *SAP Solution Manager* > *Release 7.0 EhP1 or Release 7.1*. You can find the most up-to-date version of the installation guide there.

See also the following SAP Note:

Table 9

SAP Note Number	Title	Comment
797147	Wily Introscope Installation for SAP Customers	Installation and configuration details

Procedure

1. Set up the QryFilter servlet for CPU consumption and Peak Memory Consumption.
2. Start the Fiddler tool.
3. In SAP NetWeaver, go to *OpenSQLMonitors* and refresh to clear the screen.
4. Note the start time of the scenario. Run the scenario and capture the Wily's graph.
5. Right-click on the graph and export it to Microsoft Excel. Multiply by 15 and then sum the rows in the time range. This is in milliseconds.
6. Look at the graph and note down the max point. This is the peak memory measurement.
7. After running the scenario, Fiddler output is used to capture the Communication results as follows:
   - Number of round trips between front-end and application. This is located on the last line of the first column in the Fiddler screen.
   - Data volume, in KB, transferred between front-end and application. Select everything on the Fiddler screen, right-click to copy the summary and paste (special) into Excel. Perform a sum of the body row to get the data volume transferred in Bytes, which you can convert to KB.
8. In SAP NetWeaver, go to *OpenSQLMonitors*, select the output, and paste into Excel.
9. Sum the number of Selects, Inserts, Updates and Deletes.

3.3.1.2 Application Code Instrumentation (CA Wily)

CA Wily Introscope is used by SAP Solution Manager Diagnostics (SMD) as a third-party tool to enable Java application analysis and monitoring. One task of the Onboarding service for SAP Solution Manager involves configuring the Wily Introscope application and instrumentation of the product code for the strategy management application.

During the on-site visit, the CA Wily Introscope Agent (ISAgent) is installed and configured. This allows the agent to be started up with the J2EE server process which hosts the application on the same Java Virtual Machine. After the modification, the SAP J2EE engine has to be restarted.

This agent software reports the metrics derived from the application to the CA Wily server (known as “Enterprise Manager”), which generally resides on the SAP Solution Manager/Solution Manager Diagnostics host.

The CA Wily Introscope agent must be enabled to report certain metrics to the Enterprise Manager. This configuration is called Instrumentation.

3.3.1.3 Application Monitoring Configuration (CA Wily)

Application monitoring in Introscope is configured using Management Modules and Elements. Management Modules have to be created for each domain as a container object for Introscope Dashboards, but also for other Elements, for example, Metric Groupings, Alerts, or Report Templates.

Introscope dashboards are created to organize Introscope metrics in a meaningful set of views customized individually for each application to include in the Introscope environment. Dashboards present key performance indicators and alerts “at a glance” to enable the first-level and second-level support personnel’s application monitoring and root cause analysis tasks.

Introscope alerts are based on Introscope’s heuristic modeling of key performance indicators. Every key performance indicator has a matching heuristic metric, indicating the current state of the key performance indicator (for example, a value of 1 shows normal state, values greater 1 indicate that the current state of the heuristic's key performance is outside of normal ranges).

The strategy management application component uses resource monitors installed with SAP NetWeaver system database. The data is stored in the strategy management schema, which is composed of tables prefixed with `CPMS_`.

3.3.2 Interactive Publisher Monitors

3.3.2.1 Java (JPIP) Session Monitor

Use the *JPIP Session Monitor* to restart Interactive Publisher, restart a session, shut down an instance, and refresh the tree, session, and instance status.

You can also review the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time.

The *JPIP Session Monitor* is a Tools utility that you access using `http://<nw_server>:<port>/strategy/tools`. Then choose *JPIP Session Monitor*.

3.3.2.2 PAS Query Parameters

You can see internal usage using the `pipinfo` and `pipstats` administrator parameters shown below.

The `pipinfo` parameter displays information about the system. Use `pipinfo` to shut down, update, or restart a session based on an Application Server username and model connection. The available options for `pipinfo` are:

Syntax

```
pipinfo=
[shutdown&session=[<pasuser>/]<model_connection>|
restart&session=[<pasuser>/]<model_connection>|
update&session=[<pasuser>/]<model_connection>]
```

For example, use these parameters to shutdown the session for context HFPBM, where Guest is the user for this session:

```
pipcontext=HFPBM&pipinfo=shutdown&session=GUEST/HFPBM
```

Use these parameters to restart the session:

```
pipcontext=HFPBM&pipinfo=restart&session=GUEST/HFPBM
```

The `pipstats` parameter displays statistics about the cache and sorts the information based on the specified option. The available options for `pipstats` are:

Syntax

```
pipstats=[mon_sessionlist | mon_instancelist!<session_name>] [,<count>]
```

For more information about `pipinfo` and `pipstats`, see Administrator Parameters [page 73].

3.4 Component-Specific Monitoring

To monitor components centrally, you must use a central monitoring system (CCMS).

The persistence service components and the print service can be monitored using the CCMS GRMG (Generic Request and Message Generator) tool. It performs a connection check to the persistence and print services to check if the service is available. You specify the required configuration settings in a GRMG Customizing file.

For more information about the monitoring, see the Operations area on SAP Service Marketplace at service.sap.com/instguides > *SAP NetWeaver* > *<Release Version>* > *Operations*.

To configure the GRMG scenario, execute the following steps:

1. Create a file called grmg-customizing.xml containing the XML below. The following values need to be set:

   Table 10

Value	Description
%host%	The IP address of the host where the NW Java server is running
%port%	The port of the NW Java server
%bui_user_name%	This must be a username that exists on the NW Java server and that also has the BUI_USER role

Value	Description
%bui_user_password%	The password for the user

Syntax

```
 <customizing> <control> <grmgruns>X</grmgruns> <runlong/> <errorlog/> </
control> <scenarios> <scenario> <scenname>BUI_PS</scenname> <scenversion>001</
scenversion> <sceninst>100<sceninst> <scentype>URL</scentype> <scenstarturl>
http://%host%:%port%/sap/poa/sbc/ps/CORE/grmg </scenstarturl> <scenstartmod>Not
Used</scenstartmod> <scentexts> <scentext> <scenlangu>E</scenlangu>
<scendesc>GRMG: BUI Java Persistence Service</scendesc> </scentext> </scentexts>
<components> <component> <compname>GRMG_RT</compname> <compversion>001</
compversion> <comptype>Not Used</comptype> <comptexts> <comptext> <complangu>E</
complangu> <compdesc>Basic Authentication</compdesc> </comptext> </comptexts>
<properties> <property> <proptype>No_Encrypt</proptype>
<propname>LOGON_PASSWORD</propname> <propvalue>%bui_user_password%</propvalue> </
property> <property> <proptype>No_Encrypt</proptype> <propname>LOGON_USER</
propname> <propvalue>%bui_user_name%</propname> </property> <property>
<proptype>No_Encrypt</proptype> <propname>USE_BASIC_AUTHENTICATION</propname>
<propvalue>X</propvalue> </property> </properties> </component> <component>
<compname>PARAM</compname> <compversion>001</compversion> <comptype>Not Used</
comptype> <comptexts> <comptext> <complangu>E</complangu> <compdesc>Parameter</
compdesc> </comptext> </comptexts> <properties> <property> <proptype>No_Encrypt</
proptype> <propname>PARAM</propname> <propvalue>PARAM</propvalue> </property> </
properties> </component> </components> </scenario> <scenario> <scenname>BUI_PRS</
scenname> <scenversion>001</scenversion> <sceninst>100</sceninst> <scentype>URL</
scentype> <scenstarturl> http://%host%:%port%/sap/poa/sbc/prs/print/grmg </
scenstarturl> <scenstartmod>Not Used</scenstartmod> <scentexts> <scentext>
<scenlangu>E</scenlangu> <scendesc>GRMG: BUI Java Print Service</scendesc> </
scentext> </scentexts> <components> <component> <compname>GRMG_RT</compname>
<compversion>001</compversion> <comptype>Not Used</comptype> <comptexts>
<comptext> <complangu>E</complangu> <compdesc>Basic Authentication</compdesc> </
comptext> </comptexts> <properties> <property> <proptype>No_Encrypt</proptype>
<propname>LOGON_PASSWORD</propname> <propvalue>%bui_user_name%</propvalue> </
property> <property> <proptype>No_Encrypt</proptype> <propname>LOGON_USER</
propname> <propvalue>%bui_user_name%</propvalue> </property> <property>
<proptype>No_Encrypt</proptype> <propname>USE_BASIC_AUTHENTICATION</propname>
<propvalue>X</propvalue> </property> </properties> </component> <component>
<compname>PARAM</compname> <compversion>001</compversion> <comptype>Not Used</
comptype> <comptexts> <comptext> <complangu>E</complangu> <compdesc>Parameter</
compdesc> </comptext> </comptexts> </properties> <property>
<proptype>No_Encrypt</proptype> <propname>PARAM</propname> <propvalue>PARAM</
propvalue> </property> </properties> </component> <components> </scenarios> </
customizing>
```

2. Save your entries.
3. Log on to the Central Monitoring System (CEN).
4. Call up transaction GRMG. To do that, in the execution text field in the upper left corner, enter /nRZ20 and press `Enter`.

5. Choose *Upload* and select the relevant scenario.
6. Once the scenario is uploaded, choose *Edit/Delete* to verify that the content is correct.
7. Choose *Start*.

4 Management of Interactive Publisher and Application Components

SAP provides you with an infrastructure to help technical support consultants and system administrators effectively manage all SAP components and complete all tasks related to middleware technology.

4.1 Administration Tools of Software Components

Interactive Publisher has several mechanisms that help you maintain and optimize the strategy management application.

Strategy Management Tools

Table 11

Tool	Detailed Description
Administration Application	The administration application enables strategy management administrators to create the necessary components for the application and also maintain and optimize the application. For more information about the administration application, see the SAP Library application help for SAP Strategy Management at help.sap.com/epm. Review the topic *Administration*.
Tools utility	Use the *Tools* utility to access the following tools: *Add New Database* *UI Strings Administrator* *PAS Query* *Transporter* *Auditor* *Version* *JPIP Session Monitor* *Application Tracing* For information, see Tools Utility [page 20].

> **Note**
>
> A **model connection** is a named collection of Web authentication users associated with an Application Server user, which is associated with an Application Server dimensional model.

Logs on SAP JEE Server

The log messages on JEE server are written to the multiple log files present in the server installation directory. These files can be found at *\usr\sap\\\j2ee\cluster\server\log*. The log messages can also be viewed using the NetWeaver Administrator (NWA) Log Viewer. The NWA alias for JEE server is /nwa.

If NWA is not deployed on any server, contact the system administrator.

Before you look at the log in the Log Viewer, you must check the default log configuration set. Not all types of categories and locations are enabled on the server by default. To check the configuration, execute the following steps:

1. Open NWA.
2. Go to *Problem Management* > *Troubleshooting*.
3. Go to *Logs and Traces*.
4. Go to *Log Configuration*.
5. Set the *Severity* for your custom categories and save the configuration.

Check the Logs:

In the *Logs and Traces* tab of NWA you can find the link to Log Viewer.

The filters can be set for all the available logs in the viewer.

SAP System Tools for Managing Strategy Management

Table 12

Tool	Location	Detailed Description
SAP NetWeaver Administrator **Configuration**	Click *Configuration* > *Infrastructure* > *Java System Properties*. Click the *Applications* tab and filter on the strategy management application.	Use SAP NetWeaver Administrator to maintain the Java System Properties for the strategy management application. For information about how and why to modify these properties, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101.
SAP NetWeaver Administrator **UME**	Click *Configuration* > *Security* > *Identity Management*.	None

4.1.1 Administration Application

The administration application enables strategy management administrators to create the necessary components for the application and also maintain and optimize the application. Administrators can create model connections, set caching defaults, and set up scorecards for strategy management users.

For information about the administration application, see the SAP Library application help for SAP Strategy Management at help.sap.com/bosm101. Review the topic *Administration*.

4.1.2 Tools Utility

You can access the *Tools* utility by issuing this URL in the *Address* box of a browser window.

`http://<nwce_server>:<port>/strategy/tools`

The following information describes the tools in the *Tools* utility.

Table 13

Tool	Detailed Description
Add New Database tool	The installation provides two storage collections in the SAP NetWeaver System database for storing scorecard information. *Pwsample* contains sample scorecard information that you typically access after installation to explore a sample scorecard. *Pw* initially contains no scorecard data. You can use it to start implementing your scorecards. If your site acts as an application service provider and you must support multiple customers with separate data, you need to add and maintain another database. Use the *Add New Database* tool to add another storage collection to the SAP NetWeaver System database. For information, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Adding Another Storage Collection to the System Database*. When you add a storage collection, the name automatically appears on the *Launch* page so users can start an application and begin storing information in that storage collection. If you bypass the *Launch* page by using direct links to the applications, then you must set the *PwDatabase* Java System Property in the strategy management area of SAP NetWeaver to specify the default database to use. For more information, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Configuring the Application Properties*.
UI Strings Administrator tool	If your company requires you to use specific terminology that is different from the terminology used in the application, you can change the system text using the *UI Strings Administrator* tool. For information, see the *Server Configuration Help for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Customizing Application Strings*.
PAS Query tool	Enter Interactive Publisher parameters in the *PAS Query* tool to control caching, monitor caching statistics, monitor system information, and create model connections. See the sections about Interactive Publisher parameters later in this guide for more information about the parameter. For more information about the *PAS Query* tool, see PAS Query Tool [page 48].
Transporter tool	Use the *Transporter* tool to do any of the following: • Export strategy management application component data and model connection data from the SAP NetWeaver System database into a .ZIP file. You can customize the export by excluding a set of custom data from the export. You can exclude one or more of the following: KPI descriptions, objective descriptions, initiatives, model connections, comments, Audit log comments, and Model Designer data. By default, all data is included in the export unless you specify to exclude certain items.

Tool	Detailed Description
	Then import a .ZIP file containing strategy management application component data and model connection data into the SAP NetWeaver System database. For more information, see Backup and Restore [page 24] and Transport Change Management [page 39] • Periodically delete data during a transport. For more information, see SAP NetWeaver System Database Management [page 28].
Auditor tool	You can view the audit log of certain areas of the administration application. Whenever you add entries, modify entries, or rename entries in one of these administration screens, and click *Save*, the settings are logged: • *Manage Models* • *Set Defaults* • *Update User Responsibilities* • Delete Obsolete Items You can also set the *AuditLogComments* Java System Property that prompts the strategy management administrator to add a comment every time they modify something in one of the administrator functions listed above. The comment is added to the log next to the activity. For more information, see Logging of Application Configuration Settings [page 12].
Version tool	The *Version* tool displays the current version of the application. For more information, see Displaying Version Information [page 41].
JPIP Session Monitor tool	If you are running the Java (JPIP) version of Interactive Publisher, use the *JPIP Session Monitor* to restart Interactive Publisher, restart a session, shut down an instance, and refresh the tree, session, and instance status. You can also review the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time. For more information, see Java (JPIP) Session Monitor [page 15].
Application Tracing	The *Application Tracing* tool allows you to enable and disable tracing for Application Server and Interactive Publisher. Fore more information, see Enabling the Log for the Interactive Publisher Worker Thread [page 11].

In addition, you can use the *Create Named Queries* tool located at this URL to add all the SQL statements to the DB tables.

`http://<nwce_server>:<port>/strategy/tools/namedquery.jsp`

Although the import function of a transport usually updates the SQL statements, sometimes an `import.zip` is not provided, or sometimes an SQL statement is not added to the `import.zip`. When you are applying a hot fix or support package, you may be instructed to create named queries.

4.1.3 SAP NetWeaver Administrator

4.1.3.1 Java System Properties Configurations

Use SAP NetWeaver Administrator to maintain the Java System Properties for the strategy management application.

For information about how and why to modify these properties, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Configuring the Application Properties*.

4.1.3.2 Maintenance of Users in the User Management System

You manage users and system groups, LDAP users and system groups, and SAP BusinessObjects users and system groups in UME. Go to *Configuration* > *Security* > *Identity Management*.

For information about adding strategy management users to UME, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101.

4.2 Starting and Stopping

Table 14

Software Component	Start and Stop Sequences and Tools		
	Sequence	Tool	Detailed Description
Java Persistence Service and Printing Service	-	NetWeaver Administrator	Go to *http://<host>:<port>/nwa* > *Operations* > *Start & Stop* > *Java Applications*. Search for application `poa~sbc~bui~server~adapter~nw~eap`

Interactive Publisher maintains an active Application Server session for the duration of the session. The session is maintained even when a browser client logs off.

There are certain times where you may want to stop and start Interactive Publisher and Application Server sessions and subsystems.

- If you need to update an Application Server dimensional model, you need to stop all Interactive Publisher and Application Server sessions using that model.
- If you are upgrading from a previous version of the software, you need to stop and restart the SAP SM Listener Service and the WWW Service.

When you use Interactive Publisher to stop an Application Server session for a particular model, the dimensional model is unavailable for selection when creating new reports, and any existing reports that use that dimensional model are unavailable. In addition, each user's Work database for that model is automatically deleted.

Table 15

Tool	Detailed Description
JPIP Session Monitor	Restarts Interactive Publisher, restarts a session, shuts down an instance, and refreshes the tree, session, and instance status. Also shows the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time. You can sort a column in ascending or descending order, and you can control which columns are displayed by clicking the arrow in the column title.
`Pipinfo=restart`	Restarts all sessions.
`Pipinfo=restart &session=<model-connection>`	Restarts all sessions for a specified model connection.
`Pipinfo=restart &session=<pas-user>/ <model-connection>`	Restarts only those sessions with a specified Application Server user ID and model connection.
`Pipinfo=shutdown`	Shuts the session down.
`Pipinfo=shutdown &session=<model-connection>`	Stops a session (and instances of those sessions) of an Application Server user and model connection.
`Pipadmin=shutdown`	Stops Interactive Publisher and all subsystems.
`Pipadmin=restart`	Restarts Interactive Publisher and all subsystems and rereads data from the Interactive Publisher Registry/cache.

4.3 Backup and Restore

Backup and restore of Interactive Publisher files and application data is performed using the following tools shown in the table below. Backups and restores can be performed online while the systems are running. There is no need to shut down any systems.

Data storage locations:

Table 16

Data	Component	Location
Application data	Application Components	SAP NetWeaver system database
Configuration data	Application Components, Interactive Publisher	Java System Properties in SAP NetWeaver Administrator ▶ *Configuration* > *Infrastructure* > *Java System Properties* ▶

Data	Component	Location
Logs and trace files	Application Components	Default logging/tracing in SAP NetWeaver Administrator ▶ *Troubleshooting* ▶ *Logs and Traces* ▶ *Log Viewer* ▶
User data	Application Components, Interactive Publisher	UME in SAP NetWeaver

Backup:

Table 17

Tool	Command	Detailed Description
Microsoft Windows File Manager	▶ *Edit* ▶ *Copy* ▶	Used to back up the contents of the `\<install-dir>\SAP Strategy Management \InternetPub` directory, including the `\InternetPub` directory
Transporter tool	`http://<nw_server>:<port>/strategy/tools` Click *Transporter*. On the *Transporter* page, click *Export All Data* or *Exclude Custom Data*, and then choose *Export Database*.	Exports strategy management application component data and model connection data from the SAP NetWeaver System database into a ZIP file. For information about using the Transporter tool for software change management, see Software Change Management [page 39]. For information about using the *Transporter* tool during an upgrade, see the *Server Upgrade Guides* on the SAP Help Portal at help.sap.com/bosm101.
SAP NetWeaver Configuration	Use SAP NetWeaver Administrator's tools and guidelines for configuration of Java System Properties.	Java System Properties of the strategy management application must be backed up using tools and guidelines provided by SAP NetWeaver Administrator. For more information about Java System Properties, see Administration Tools of Software Components [page 19].
SAP NetWeaver UME	Use SAP NetWeaver Administrator's tools and guidelines for UME.	Management of strategy management users are in UME in SAP NetWeaver Administrator.

Restore:

Table 18

Tool	Command	Detailed Description
Microsoft Windows File Manager	▶ *Edit* ▶ *Paste* ▶	Used to restore the backed up contents of the `\<install-dir>\SAP Strategy Management\InternetPub` directory.
Transporter tool	`http://<nw_server>:<port>/strategy/tools` Click *Transporter*.	Imports a `ZIP` file containing strategy management application component data and model connection data into the SAP NetWeaver System database.

Tool	Command	Detailed Description
	On the *Transporter* page, click *Import into Database* function.	
SAP NetWeaver Administrator Configuration	Use SAP NetWeaver Administrator's tools and guidelines	Java System Properties of the strategy management application must be backed up using tools and guidelines provided by SAP NetWeaver Administrator.
SAP NetWeaver Administrator UME	Use SAP NetWeaver Administrator's tools and guidelines	Strategy management users that are stored in UME must be backed up using tools and guidelines provided by SAP NetWeaver Administrator.

4.4 Periodic Tasks

4.4.1 Periodic Tasks for Interactive Publisher

This table shows the periodic tasks for Interactive Publisher:

Table 19

Task	When to Perform the Task
Delete Work databases	Manually delete all Work databases periodically as needed if you do not use Interactive Publisher to shut down Application Server processes, or if you do not shut down the processes on a regular basis.
Add or remove users in a model connection or review session information related to a model connection	Perform these tasks on an as-needed basis according to the day-to-day activity in your organization.

4.4.1.1 Work Database Management

4.4.1.1.1 Clearing Work Databases

The Application Server Work database stores information about the user's current session. The information includes the user's current selections and User-Defined Hierarchies. When the user initiates a new session in the strategy management application, a new Work database is created.

Since Interactive Publisher may maintain many Work databases per Application Server process, it is important that these DB* Work databases get removed when Application Server is shut down.

If you do not use Interactive Publisher to shut down Application Server processes, or if you do not shut down the processes on a regular basis, you should manually delete all Work databases periodically. This way, the DB* databases will not clutter the Work database directory and use up disk space.

During the Interactive Publisher shutdown process, all Work databases are deleted by Interactive Publisher.

Procedure

1. In Microsoft Windows Explorer, navigate to the following location:

   <drive>:\Program Files (x86)\SAP Strategy Management\ApplicationServer\home
2. Select all the Work database files, right-click, and then choose *Delete*.

   Work databases have filenames in the format `<web_authentication_name><model_name>`. For example, TESTUSERJUICE.
3. Select all the DB* files, right-click, and then choose `Delete`.

4.4.1.1.2 Using Individual Work Databases or a Single Shared Work Database

By default, individual Work databases are created by Interactive Publisher for each user. This is a requirement in the application. If you have custom applications, you can control whether to use individual Work databases or use a shared Work database among all users. In situations where you are experiencing performance problems in your custom application, or you may have thousands of users with very large Work databases that may cause a disk space problem, you may want to use a single shared Work database.

Procedure

1. Access `regedit.jsp` by issuing this URL in a Web browser:

   `http://<nw_server>:<port>/strategy/tools/regedit.jsp`
2. Change the *IndividualWorkDatabase* value. A value of `1` means that you want the system to create an individual Work database in Application Server for each user who accesses the application. This is the default setting. A value of `0` means that you want the system to use a single Work database in Application Server for all users.

4.4.1.2 Model Connection Management

Model connections may need some periodic maintenance at your own discretion in the following areas:

- Adding or removing model connections
- Adding or removing users in model connections
- Reviewing session information related to model connections

The following table shows several methods of managing model connections.

Table 20

Tool	Detailed Description
Administration > *Manage Models* in the administration application	Create and manage model connections. Because this method of managing model connections involves using the administration application, this topic is discussed more fully in Periodic Tasks for the Application [page 29].

Tool	Detailed Description
`Pipadminex`	Use the `pipadminex` parameter to do any of the following: • Create a model connection and add users to it. • Delete sessions of a model connection or delete users in model connection sessions. • List users in a model connection session, or list model connection names and Application Server usernames. • Display session information. • Delete a user from a model connection and delete a session. • Rename a model connection or a session. For more information, see Pipadminex [page 74].
Transporter	Use the Transporter to export model connections from one version of strategy management and import them into another version of strategy management. This is useful when upgrading to a newer version. During the export of strategy management data, model connections are included by default. When you import the data, the model connections are added to the strategy management application. It is not possible to use the Transporter to delete individual model connections. Model connections exist in a protected list of tables in the SAP NetWeaver System database. Even if you choose *Delete all data in database before the import* during an import, the model connections are not deleted. If you need to delete a model connection, use the *Manage Models* link in the administration application.

4.4.1.3 SAP NetWeaver System Database Management

During a transport, you may want to delete data periodically. The Transporter utility has several options for deleting data.

You access the Transporter utility using the following URL:

`http://<nw_server>:<port>/strategy/tools/transporter.jsp`

Prerequisites

You are a strategy management developer or administrator who is familiar with the Strategy Management table schema.

Features

During a transport, you may want to delete data periodically:

- If you want to have a clean environment before importing data into the database, you can click the *Delete all data* option and then click *Delete Data* to delete all data.
- If you are ready to import data, and you did not previously delete data and you want to do so now, you can select *Yes* at the *Delete all data in database before the import* prompt.

> **Note**
>
> This setting does not delete model connections from the SAP NetWeaver System database. Model connections exist in a protected list of tables in the SAP NetWeaver System database. If you need to delete a model connection, use the *Manage Models* link in the administration application.

- If you want to delete the data of a particular database name, you can click the *Delete data by database name* option and then click *Delete Data*.
- If you want to delete the data of a particular table name because the table is obsolete or is not used in the database, you can click the *Delete data by table name* option and then click *Delete Data*. We do not recommend using this option. Deleting data by table name could introduce data integrity issues in the database.

4.4.2 Periodic Tasks for the Application

The strategy management administrator uses the *Administration* section in the administration application to perform application, system, and user maintenance.

For more information, the SAP Library application help for SAP Strategy Management on the SAP Help Portal at help.sap.com/bosm101 . Review the topic *Administration*.

This table shows the periodic tasks for the application:

Table 21

Task	When to Perform the Task
Notify subscribers when certain aspects of the scorecard have changed, such as a change in objective status, a change in KPI status, initiative milestones are due, a comment has been added.	Set up a nightly schedule in the *Scheduler* to run the *Scorecard Notifications* and *Initiative Notifications* activities.
Update temporary user reassignments that are set in the *Update User Responsibilities* screen.	Set up a nightly schedule in the *Scheduler* to review the start dates and end dates of all temporary assignments and update temporary user assignments accordingly.
Manage model connections as follows: • Create model connections • Add or remove strategy management users in a model connection • Add or remove Application Server users in a model connection	Perform these tasks on an as-needed basis according to the day-to-day activity in your organization.
Maintain data and workflow process in the Entry and Approval application	Perform these tasks on an as-needed basis according to the day-to-day activity in your organization.

Task	When to Perform the Task
Recreate certain reports or dashboards	Anytime a dimension, dimension member, attribute, or measure is removed from a dimensional model, you must recreate any reports and dashboards that use the model.

The tasks mentioned in this section are not dependent on each other and can be performed in any order. These tasks are performed periodically when the need arises.

These tasks do not affect system availability. Users can be running the administration application or the strategy management application while the administrator carries out these task.

4.4.2.1 System Defaults Maintenance

You may want to periodically maintain the system defaults used in the administration application.

The administrator of strategy management sets up and maintains system defaults.

Prerequisites

You are logged into the administration application as the strategy management administrator.

Features

Periodic Tasks

You might need to carry out one or more of these tasks periodically to maintain system defaults:

- Specify the users designated as strategy management administrators.

  Use *Administration* > *Set Defaults* in the administration application to set system defaults.

  For information about maintaining system defaults, see the SAP Library application help for SAP Strategy Management on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Administration* > *System Defaults Selection*
- Enable or disable Interactive Publisher tracing.

  Use the *Application Tracing* tool in the *Tools* utility to enable or disable tracing.

  For more information, see Tools Utility [page 20].

4.4.2.2 Model Connection Maintenance

A model connection is a named collection of Web authentication users associated with an Application Server user, which is associated with an Application Server dimensional model.

Use the *Administration* > *Manage Models* section of the administration application to create and maintain model connections.

Prerequisites

You are logged into the administration application as the strategy management administrator.

Features

Periodic Tasks

You may need to do the follow tasks periodically:

- Create, modify, or delete a model connection.
- View a model connection's definitions, the Application Server users for a model connection, or the Web authentication users assigned to an Application Server user.
- Add or remove an Application Server user for a model connection, or specify a different Application Server user for the model connection rather than the one selected.
- Add or remove a Web authentication user for a model connection.
- Test the connection to Application Server.

For information about maintaining model connections, see the SAP Library application help for SAP Strategy Management on the SAP Help Portal at help.sap.com/bosm101 . Review the topic *Administration* > *Model Connection Development*.

4.4.2.3 Schedule Maintenance

Use the Scheduler to set schedules that define how frequently to check for milestone due dates and objective and KPI status changes.

Use the *Scheduler* section of the administration application to create and maintain schedules.

Prerequisites

You are logged into the administration application as the strategy management administrator.

Features

Periodic Tasks

You may need to do the follow tasks periodically:

- Add, remove, or modify a schedule
- Enable or disable all Scheduler notifications for a task type

For more information about maintaining schedules, see the SAP Library application help for SAP Strategy Management on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Administration* > *Scheduler*.

4.4.2.4 Entry and Approval Application Maintenance

The Entry and Approval application allows users to manually enter data into a dimensional model and to monitor the data entry effort through a workflow process. Once the data entry process has been completed, the Entry and Approval application loads the data into the dimensional model.

Use *Entry and Approval* in the administration application to set defaults for the dimensional model and to launch the Entry and Approval application. Use the Entry and Approval application to maintain measure sets.

Prerequisites

You are the administrator of strategy management and an administrator of the Entry and Approval application.

Features

Periodic Tasks

You may need to do the following tasks periodically:

- Update the current period on a regular basis such as monthly, depending on how often data entry is occurring. You also update the date range depending on how wide a range (including history and future target) is wanted.
- Reset and re-release measure sets (for a new cycle of data entry and approval) -- this is in conjunction with resetting the current period and date range.
- Load the Application Server model.
- Enter target and historical data.
- Add new measure sets and release them.
- Reassign users to data entry and approval tasks.
- Change the list of users who are administrators. This might be necessary if new models are added or if the Entry and Approval administrator for a particular model needs to be updated.

For more information, see the SAP Library application help for SAP Strategy Management on the SAP Help Portal at help.sap.com/bosm101. Review the topic *Entry and Approval*.

4.4.3 Periodic Tasks for User Maintenance

A role is a named set of system users (strategy management users defined in SAP NetWeaver UME) and system groups with certain reporting and administrative permissions and accessibility into the administration application and the strategy management application. All the users in the role share the same functionality and access within the applications across all the contexts that are available to them.

The administrator of SAP NetWeaver creates and maintains roles to control each user's experience in the administration application and the strategy management application in terms of permissions, views, and accessibility.

Use the *Identity Management* section of SAP NetWeaver to create and maintain roles in terms of users, permissions and accessibility.

Prerequisites

You are logged into SAP NetWeaver as the administrator.

Features

Periodic Tasks

The following tasks may need to be performed periodically to maintain roles:

- Add or remove users in roles
- Change permissions of the role
- Change the tabs available to the role
- Add more roles
- Delete roles
- Rename roles

For more information about creating and maintaining roles, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101.

4.5 User Management

The strategy management application uses UME features in SAP NetWeaver to manage its users. For information about user management in the strategy management application, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101.

Table 22

Scenario(s)	Detailed Description	Tools to be Used
Assigning Roles to a Java Persistence user	A number of roles exist for the users of Java Persistence and these should be assigned to the principal based on what they should have access to	Identity Management in the NetWeaver Administrator should be used to configure roles, for example, http://<host>:50000/useradmin/index.jsp

5 High Availability

SAP Strategy Management is meant for large user deployments. To maximize system availability for large user deployments, you can implement a High Availability (HA) configuration. This allows certain system maintenance activities to occur without taking the application offline, and also provides an infrastructure to recover quickly from unexpected machine failures.

While there are many reasons why a high availability configuration is desirable, this section focuses on the configuration and deployment of the strategy management system in a high availability environment.

5.1 Architecture Considerations

The SAP Strategy Management application consists of several components working together. The components with relevance to high availability are Interactive Publisher and Application Server.

5.2 Recommended Cluster Configuration

There are multiple ways to install a high availability configuration of SAP NetWeaver. This depends on the number of nodes participating in the cluster as well as the configuration of the database component. For purposes of this discussion, the following specifications are used:

- A 64-bit version of Microsoft Windows Server 2008 for the operating system
- Microsoft SQL Server for the SAP NetWeaver System Database
- Microsoft SQL Server installed in a cluster configuration

This scenario adheres to the guidelines of the *Installation Guide for SAP NetWeaver Composition Environment* on Windows: MS SQL Server.

The SAP NetWeaver configuration has several options for installation and configuration. The choice of configuration for SAP NetWeaver and the SQL Server database should not matter as long as they are valid and supported configurations as defined by the installation guide.

The steps described in this discussion provide guidance for Interactive Publisher and Application Server.

5.3 Configuration Process

To configure your system for high availability, you create a separate resource cluster that collects all the associated strategy management resources together and permits them to be moved from node to node as a group.

Prerequisites

Software

- Microsoft Cluster (MSCS) configuration using the Windows Server family
- Microsoft SQL Server installed a high availability configuration
- SAP NetWeaver on Windows: MS SQL Server using high availability (MSCS) options

Hardware

- Fault Tolerant Disk Subsystem
- Two or more physical nodes

Process

1. Set up the strategy management cluster. For more information, see Setting Up a Strategy Management Cluster [page 35].
2. Install strategy management services. For more information, see Installing the Strategy Management Services [page 36].
3. Add strategy management services to the strategy management cluster. For more information, see Adding Strategy Management Services to the Cluster [page 36].
4. Update the Java System Properties. For more information, see Updating the Java System Properties [page 37].
5. Move `.INI` files to the shared disk. For more information, see Moving .INI Files to the Shared Disk [page 37].
6. Update the shortcut for the Application Server program. For more information, see Updating the Application Server Shortcut [page 37].
7. Set up a Link ID for certain implementations. For more information, see Setting Up a Link ID for Certain Implementations [page 38].

5.3.1 Setting Up a Strategy Management Cluster

Install Application Server and Interactive Publisher in their own group, which has a dedicated shared Physical Disk resource and a dedicated IP resource.

Procedure

1. To create a new resource cluster in the *Cluster Administrator*, select *File* > *New* > *Group*. Specify a group name, for example **SSM**, and click *Next*.
2. Select the nodes where the strategy management services run and click *Add* to add them to the group. Then click *Finish*. The *SSM* group is created.
3. Obtain an available IP Address from your network team and assign a unique name for use within your network.

   Create a new IP Address resource cluster in the *Cluster Administrator* by selecting the new `SSM` Group and choosing *File* > *New* > *Resource*.

   Select the Resource Type IP Address and enter the other parameters. This example uses the name SAP SSM IP. Click *Next*.

4. Enter the available IP Address and appropriate Subnet mask, and select the appropriate network. Click *Finish*.
5. Repeat the steps for the Physical Disk.

5.3.2 Installing the Strategy Management Services

Procedure

1. Log into one of the physical nodes and bring the new physical disk online using the *Cluster Administrator*.
2. If the *SSM* group is not assigned to your current node, right-click the *SSM* group name and move it to your current node.
3. Run the SSM Service installation (`xSSMSrv0x_0.exe`) and select the disk associated with the SSM physical group as the destination.
4. After installation, move the disk to the alternate node and log onto the second node.
5. Repeat the SSM Service installation and specify the same target disk. It will overwrite the files, but this is expected behavior.

5.3.3 Adding Strategy Management Services to the Cluster

You must make the strategy management services cluster-aware.

Prerequisites

The programs and services are created on each individual node.

Procedure

1. In the *Cluster Administrator*, right-click the *SSM* Group, and create a new resource. In the *New Resource* dialog box, select the resource type *Generic Service*. Assign it the name *SAP SSM Listener*, and click *Next*.
2. Select the possible owners and click *Add*. Then click *Next*.
3. Add the Physical Disk and IP Address as Resource Dependencies, and click *Next*.
4. Enter **`lssagent`** as the service name and click *Next*.
5. In the Registry Replication dialog, click *Add* and enter the key: `SOFTWARE\Wow6432Node\SAP\SSM\InternetPub`. Then click *Finish*.

5.3.4 Updating the Java System Properties

Procedure

1. Start the SAP NetWeaver Administrator.
2. Log on as administrator with the global password you provided when you installed SAP NetWeaver.
3. Select *Configuration*.
4. Click the *Infrastructure* tab.
5. Select *Java System Properties*.
6. In the *Templates* section, select the template that corresponds to your strategy management installation.
7. Click the *Applications* tab in the *Details* section.
8. In the *Name* column, type **`strategy`** and press `Enter` to list the strategy applications.
9. Select the name *xapps~cpm~sm~strategymanagement*.
10. In the *Name* column of *Extended Details*, enter the strategy management network name and press `Enter`. For example, **`pgepmssm`**.
11. Change `template.strategy` to the directory on the Cluster Disk. For example, `T:\Program files (x86)\SAP Strategy Management\InternetPub`.

5.3.5 Moving .INI Files to the Shared Disk

Procedure

1. Create the system environment variables `LSLINKINI` and `LSLINKCNF` on both nodes. Specify the directory only, for example, `T:\`.
2. Copy `LSDAL.INI` from the `c:\windows` directory to `T:\`.
3. Copy `LSDAL.CNF` from the `c:\windows` directory to `T:\`
4. Copy `lsserver.ini` from `c:\windows` to the Cluster Disk (e.g. `t:\lsspgx.ini`).
5. In the administration application, update the model connection definition by specifying `t:\lsspgx.ini` as the `INI` file.

   Also update the model connection to specify the SSM Cluster Network Name (e.g. `pgepmssm`) as the *PAS System Name*.

5.3.6 Updating the Application Server Shortcut

Procedure

1. Open the properties for the Application Server program, and add the text `"-inifile t:\lsspgx.ini"` to the strategy management target shortcut on both nodes.

2. Make sure the strategy management cluster is on the local node (this is dependent on the cluster disk resource `T:`).
3. Open `lsspgx.ini` in a text editor, and add this section:

```
[pgepmssm]
tcp_protocol=winsock
username=pipadmin
password=abcd1234
PROTOCOL=TCP
SERVICE=PILOT
CURSOR=LSSCMPTR
```

5.3.7 Setting Up a Link ID for Certain Implementations

If you use Entry and Approval or Model Designer, you must have a special Link ID called *ssm_cb_ea* to use for any of these connections. If you followed the instructions in the Installation Guide, you were instructed to create the special Link ID after installing the strategy management components.

Now you must modify a setting to adapt it for high availability.

Prerequisites

You have set up client access to the SAP NetWeaver System database. For more information, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Choose *Setting Up Client Access to the SAP NetWeaver System Database*. When using SQL Server as the system database, the 32-bit ODBC drivers for SQL Server are installed on the Windows server where Application Server is installed. The Application Server procedures are copied from the `\Strategy Management\InternetPub\procs \sqlsrvr_procs` directory to the parent directory `\Strategy Management\InternetPub\procs\`.

You have created a special Link ID called *ssm_cb_ea*. For more information, see the *Installation Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101. Choose *Creating a Link ID for Certain Implementations*.

Procedure

1. In *Administrative Tools*, make a copy of the *Data Sources (ODBC)* shortcut, and name it *Data Sources (ODBC) (32-bit)*.
2. Right-click the 32-bit shortcut and select *Properties.*
3. In the *Data Sources (ODBC) (32-bit)* dialog box, edit the target to be `%SystemRoot% \syswow64\odbcad32.exe`
4. Repeat the task on the other node(s).
5. When you created the *ssm_cb_ea* link ID in Application Server Administrator, you specified a value in the *WorkStation ID* text box of the *Link ID Properties* dialog box. The default value is the name of the current host.

   Change the value to the name of the cluster resource for SQLServer (for example, *PGEPMSQL*).

6 Software Change Management

Software Change Management standardizes and automates software distribution, maintenance, and testing procedures for complex software landscapes and multiple software development platforms. These functions support the project teams, development teams, and application support teams.

The goal of Software Change Management is to establish consistent, solution-wide change management that allows for specific maintenance procedures, global rollouts (including localizations), and open integration with third-party products.

This section provides additional information about the most important software components.

The following topics are covered:

- **Transport and Change Management** — Enables and secures the distribution of software changes from the development environment to the quality assurance and production environment.
- **Support Packages and SAP Notes Implementation** — Provide standardized software distribution and maintenance procedures.
- **Release and Upgrade Management** — Reduces the time, cost, and risk associated with upgrades.

6.1 Transport and Change Management

Application Components

The strategy management installation includes a *Transporter* tool to import and export business-related data objects between the development, quality management and production system instances. Use the *Transporter* to transport the contents of the SAP NetWeaver System database that stores the data for the strategy management application.

All objects and definitions created in the application are saved in records in the SAP NetWeaver System Database. This includes strategies, scorecards, objectives, initiatives, KPIs, comments, and so on. You can transport these objects and definitions to a different instance of the strategy management system.

The two-step process of moving an instance of the strategy management application from one system to another involves:

- Exporting the contents of the **source** instance into a system-generated `.ZIP` file using the *Transporter's Export Database* function.
- Importing the contents of the `.ZIP` file into the destination instance using the *Transporter's Import into Database* function.

Interactive Publisher

The database schema in the SAP NetWeaver system database has table `CPMS_DBVERSION` containing DB version information. The *Transporter* tool inserts bootstrap data into the table. The Transport data file has a `DBVERSION` field. The DB versions can be compared manually. This field can be used to match transport data to the right strategy management database version while transporting data across the landscape.

If there is a database version conflict, the *Transporter* tool may generate an error at runtime depending on the table/data. Error details are logged to the PipTrace.

For information about periodic tasks related to the SAP NetWeaver system database, see Periodic Tasks for Interactive Publisher [page 26].

Application component configuration settings are maintained in SAP NetWeaver so they do not need to be transported.

Users are maintained in SAP NetWeaver UME so they do not need to be transported.

Process

The transport and change management of Interactive Publisher and Application Components are handled as part of the same process. Activities include the following:

1. From the source system where Interactive Publisher is installed, copy any customized files to the same location on the destination system using an operating system tool for copying.
2. From the source system where the Software Component Archive is deployed in SAP NetWeaver, save the Java System Properties for the strategy management application.
3. From the source system where the administration application is installed, record the schedule definitions for each schedule you have set up.
4. On the source system, start the *Transporter* tool to export the contents of your SAP NetWeaver System database and model connections into a `.ZIP` file. The following URL starts the *Transporter* tool:

   `http://<source_nw_server>:<port>/strategy/tools`

   Click *Transporter* to start the *Transporter* tool.

   To export all data, choose *Export All Data*. Or, to exclude certain data, choose *Exclude Custom Data* and then specify the data you want to exclude. Then choose *Export Database*.
5. Copy the exported `zip` file from the original destination to the `\server\temp\SSM\import` directory on the SAP NetWeaver server. In a default installation, the full path is `\usr\sap\<SID>\J00\j2ee\cluster\server\temp\SSM\import`.
6. In the *Transporter* tool on the destination system, from the *Choose a database transport file* section, select the exported `zip` file you copied to the `\import` directory. Then choose *Import into Database* to import the contents of the `.ZIP` file.
7. On the destination system where the Software Component Archive is deployed in SAP NetWeaver Configuration Management, restore the Java System Properties for the application.
8. On the destination system, use the administration application to create new schedules using your recorded schedule definitions.
9. On the destination system, modify the model connections in the administration application because the IP address of the Application Server models are changed.

For more information, see the *Server Upgrade Guide* on the SAP Help Portal at help.sap.com/bosm101.

6.2 Support Packages and SAP Notes Implementation

You can find the available support packages by going to SAP Service Marketplace at service.sap.com/swdc > *Download* > *Support Packages and Patches* > *A — Index* > *S* > *SAP STRATEGY MANAGEMENT*.

The steps to apply a Support Package or a Hot Fix exist in the SAP Note for the Support Package or Hot Fix.

Critical limitations for a release are described in the Restrictions Note. The Central Note, and every Support Package Note and Hot Fix Note have a Related Notes tab that contains a link to the Restrictions Note.

6.3 Release and Upgrade Management

Table 23

Software Component	Release	Tool
Strategy management application	7.5, 10.0, 10.1	Transporter
Strategy management application	7.5, 10.0, 10.1	Fix File Store

Releases of the strategy management components are on SAP Service Marketplace at service.sap.com/swdc > *Download* > *Installations and Upgrades* > *A — Index* > *S* > *SAP STRATEGY MANAGEMENT*

For information about using the tools to upgrade your system or to transport from one system to another, see the *Server Upgrade Guide for SAP Strategy Management* on the SAP Help Portal at help.sap.com/bosm101.

6.3.1 Displaying Version Information

Procedure

1. Start the *Tools* utility by entering this URL in a browser window:
   `http://<nw_server>:<port>/strategy/tools`
2. To display the current version of strategy management, click *Version*.
3. To display complete version information about Interactive Publisher and Application Server, click *PAS Query* to start the PAS Query tool.
   Select the model connection, and then specify **`&result=version`** in the *Address* box.

Example

The following is example output:

```
SSM Version <version>
SP <support package number>
Interactive Publisher
Version <x>.<x>.<x> for Windows
Copyright (C) <year> SAP AG
Reference <xxxx> on <date> <time>
Application Server
Version <x>.<x>.<x> for Windows
Copyright (C) <year> SAP AG
Reference <xxxx> on <date> <time>
```

7 Troubleshooting

If you find that you are having troubles getting the products up and running, here are some basic troubleshooting steps to try.

7.1 Troubleshooting SAP NetWeaver Issues Related to Strategy Management

Logging and Tracing

The NetWeaver Java adapter use minimal logging and tracing as per the default NetWeaver logging and tracing level of ERROR.

Tracing and logging can be increased and output to separate log files using the standard NetWeaver procedures.

Here is the list of tracing locations used in the NetWeaver Java adapter:

- com.sap.analytics.aui
- com.sap.poa.sbc.bui
- com.sap.poa.sbc.fnd
- com.sap.poa.sbc.bui.core
- com.sap.poa.sbc.bui.core.persistence
- com.sap.poa.sbc.bui.persistence
- com.sap.poa.sbc.bui.persistence.jpa
- com.sap.poa.sbc.bui.persistence.repository
- com.sap.poa.sbc.bui.persistence.security
- com.sap.poa.sbc.bui.persistence.rest
- com.sap.poa.sbc.bui.usermanagement

Here is the list of standard NetWeaver logging categories used in the NetWeaver Java adapter:

- Application
- Performance

HTTP Sniffer to Identify Problem Areas

The application framework uses client-server communication with stable URL request patterns. This makes it easier to identify which back-end service has an issue if a problem occurs. Requests that are handled by the NetWeaver Java adapter have two distinct URL patterns:

- A URL with `sap/poa/sbc/um` indicates a user management request
- A URL with `sap/poa/sbc/ps` indicates a persistence service request

> **Note**
>
> Other requests are used to serve Flex content to the browser or serve/manipulate back-end data.

A HTTP sniffer allows the monitoring of the HTTP(S) requests from a browser and therefore can be used to help troubleshoot where a problem lies.

The HTTP requests and responses from a session can also be saved into log files. Examples of these tools include HTTPWatch and Fiddler.

Displaying User Preference Settings in NWA

There are four settings under the *Language and Region* section of *User Preferences*: language, time zone, data format, and number format.

The values for these settings are stored as custom attributes for each user.

The values are not visible by default in the NetWeaver Administrator (NWA) identity management screens, but can be made visible by executing the following configuration steps:

1. Open *Identity Management* in the NW Java AS NWA: *http://server:port/nwa/identity*
2. Choose *Configure*.
3. Go to the *User Admin UI* tab and choose *Modify Configuration*.
4. In the *Administrator - Managed Custom Attributes* field, enter the following value:
   `com.sap.poa.sbc.bui:builocaleid;com.sap.poa.sbc.bui:buitimezoneid;com.sap.poa.sbc.bui:buidateformatid;com.sap.poa.sbc.bui:buinumberformatid`
5. Save your changes.
6. A restart of the server may be required. Check the messages after saving.

After following these steps, there is a new tab called *Customized Information* visible in the *Identity Management* screen under *User Details*. This allows you to change the values directly in NWA.

For more information about *Adding Custom Attributes to the User Profile*, see SAP Help Portal at help.sap.com > *SAP NetWeaver*.

7.2 Java (JPIP) Session Monitor

Use the *JPIP Session Monitor* to restart Interactive Publisher, restart a session, shut down an instance, and refresh the tree, session, and instance status.

You can also review the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time.

The *JPIP Session Monitor* is a Tools utility that you access using `http://<nw_server>:<port>/strategy/tools`. Then choose *JPIP Session Monitor*.

7.3 Verifying the Virtual Directory is Working

Procedure

If you want to verify that a particular virtual directory is set up correctly, then display a simple image object from that virtual directory in a Web browser. For example, type this URL in a browser window to display an arrow:

`http://<nw_server>:<port>/strategy/files/greyarrow.gif`

If you are able to view the object in the browser, it means the virtual directory is set up correctly.

7.4 Anti-Virus Settings Verification

If you get unexpected errors in the application, try lowering the settings on your antivirus product or disable it temporarily to test if the errors are related to the anti-virus setting.

7.5 JavaScript Error

If you get a JavaScript error that says `Object not found`, this is generally caused because the Java JVM is unable to load a class file. Make sure the Sun JRE is version 1.6. The Java option in the Microsoft Windows Control panel should show the version information.

7.6 Copying Files and User Permissions

If you copy files in and out of folders on the server, you have to make sure you copy them in a way that does not modify user access to these files. Otherwise, users attempting to access a file may encounter problems.

8 Support Desk Management

Support Desk Management enables you to set up an efficient internal support desk for your support organization that seamlessly integrates your users, internal support employees, partners, and SAP Active Global Support specialists with an efficient problem resolution procedure. For support desk management, you need the methodology, management procedures, and tools infrastructure to run your internal support organization efficiently.

8.1 CA Wily Introscope Integration

To enable application analysis (including performance monitoring), CA Wily Introscope (IS) is integrated into SAP Solution Manager Diagnostics (SMD). SAP provides CA Wily IS instrumentation for SAP Strategy Management.

IS for Microsoft .NET is an application management solution for managed .NET applications, running on Microsoft's Common Language Runtime (CLR) environment. CA Wily IS offers Dashboards for performance and stability analysis. In addition, the Investigator provides a detailed view on all applications and environment metrics reported by the IS agent to the IS Enterprise Manager, which is the CA Wily IS server and part of SAP Solution Manager. User-specific interaction can be traced in CA Wily IS using the Transaction Trace.

Metrics, which are collected and reported through `tracers` defined in Probe Builder Directives `.pbd` files, define the information that is collected at runtime. The CA Wily IS .NET agent collects this information and reports it to the Enterprise Manager. The Enterprise Manager stores these metrics in its own database. You can view performance metrics using the IS Workstation or the IS WebView application.

Prerequisites

To enable IS for Strategy Management, install and configure the CA Wily IS .NET agent on the strategy management application server hosts.

For more about information about setting up and configuring CA Wily Introscope for strategy management, see SAP Note 1126554 as well as SAP Note 797147 and its attached FAQ document.

For more information about the installation, configuration, and use of SAP Solution Manager Diagnostics, visit SAP Service Marketplace at service.sap.com/diagnostics.

Procedure

1. Log on to Root Cause Analysis workcenter of SAP Solution Manager (transaction code `solman_workcenter`).
2. Select *System Analysis* from the detail navigation menu. Choose the query that contains the Strategy Management system or find it in *All Technical Systems*.
3. Select the Strategy Management system from the systems selection table.
4. Choose *CA Wily Introscope* and log on to the CA Wily IS WebView.

5. Choose *Start Introscope*, then log on to the Introscope WebView.
6. Do any of the following:
   - Select the *Console* tab to view Wily Dashboards.
   - Select the *Investigator* tab to view the Wily Investigator tree.
   - Select the *Transaction Viewer* tab to view Wily Transaction Trace.

8.2 Problem Message Handover

You can log problem messages on the SAP Support Portal on SAP Service Marketplace at service.sap.com/. Provide a detailed and reproducible problem description.

The following component strings are available:

Table 24

Component String	Component
EPM-SM	SAP Strategy Management
EPM-SM-APP	Applications
EPM-SM-EAI	Excel Add-In
EPM-SM-PAS	Application Server
EPM-SM-PIP	Interactive Publisher

9 Standard Parameters

This section covers all the standard parameters and values you can enter in the *Address* box of the *PAS Query* tool.

Prerequisites

9.1 PAS Query Tool

To execute parameters, use the *PAS Query* tool located at:

`http://<nw_server>:<port>/strategy/tools`

Then, click *PAS Query*.

Prerequisites

Any Web authentication user can access the *PAS Query* tool and use standard PAS Query parameters.

Administrators of strategy management can use the `pipadmin` parameters.

Features

Table 25

Field	Description
Context	Specify the model connection you want to work with. The *Address* box shows `context=<name>&` based on your selection in the *Context* dropdown list.
Address	Enter the parameters to execute, in the format: `<parameter>=<value>[&<parameter>=<value>]...` • `<parameter>` — PAS Query parameter to pass with the program. For information about parameters, see PAS Query Parameters [page 16] and Administrator Parameters [page 73]. • `<value>` — Valid value for the parameter. Specify multiple values for a parameter by using a plus sign (+) instead of a space. For example, `set=period+monthly`. The characters &, ?, +, %, and = are reserved. Do not use the reserved characters %, &, *, or + in a variable label. To use these characters for other purposes, use their hexadecimal representation, preceded by a percent sign (%). To use % as a percent sign, use %26, where 26 is the hexadecimal representation of the percent sign. Use multiple parameters by separating each one with an ampersand (&).

9.2 Syntax for Standard Parameters

Use the Interactive Publisher parameters to query the Application Server databases. To execute parameters for Interactive Publisher, use the PAS query tool. Issue this URL in a Web browser:

`http://<nw_server>:<port>/strategy/tools`

Then click *PAS Query*.

Prerequisites

Any Web authentication user can access the PAS Query tool and use standard Interactive Publisher parameters described in this section.

Features

Syntax

```
<parameter>=<value>[&<parameter>=<value>]…
```

This table describes the syntax to use in the Address text box of the PAS Query tool.

Table 26

Variable	Description
`<parameter>`	Interactive Publisher parameter to pass with the program. All parameters are described in this section. Many parameters are equivalent to commands in Application Server. For information about Application Server commands, procedures, and reports, see Application Server Help in the Application Server Administrator program. The parameters follow the same rules as in any HTML program. Use multiple parameters by separating each one with an ampersand (&).
`<value>`	Valid value for the parameter. Specify special characters and alphanumeric characters on the right side of the question mark (?). Specify multiple values for a parameter by using a plus sign (+) instead of a space. For example, `set=period+monthly`. The characters &, ?, +, %, and = are reserved. Do not use the reserved characters %, &, *, or + in a variable label. To use these characters for other purposes, use their hexadecimal representation, preceded by a percent sign (%). To use % as a percent sign, use %26, where 26 is the hexadecimal representation of the percent sign

9.3 Using Parameters

Interactive Publisher parameters are processed in the following order:

1. `&varname`, `initexecute`, and `initdql` are executed first regardless of their position in the *Address* box.
2. All other parameters except for template are then processed in the order they are listed.
3. The template parameter. If you have a result parameter that requires a template (text, report, or stream) and do not specify a template, the default header and footer templates are used.
4. Finally, Interactive Publisher clears any control variables that were set for the transaction.

Example

The following example shows how to use the `select`, `time`, `across`, and `down` parameters:

Syntax

```
select=variables+costs&select=product+output&time=yearly&across=variables,time&down=
product&result=table
```

The following example shows how to execute Application Server commands using the `dql` parameter and separating words with a plus sign (+). Although you should develop dynamic applications without using the `dql` parameter, it is provided for situations not covered by a parameter.

Syntax

```
dql=select+variables+cost+time yearly&result=text
```

The following example shows how to execute an Application Server procedure that contains Application Server `SELECT` commands. If an Application Server `smplproc` procedure contains commands such as these:

Syntax

```
select variables costs
select product
set period yearlyoutput
across variables,time down product
```

Then you can execute the procedure using the execute parameter and display the selections using:

Syntax

```
execute=smplproc&result=text
```

9.4 Across

Use `across` to display specified dimensions across the top of each page. Equivalent to the command `ACROSS` in Application Server.

Syntax

```
across=<across-dims>
```

The following section provides an overview of the variables.

Table 27

Variable	Syntax	Description
<across-dims>	across=<across-dims>	List of dimensions.

Example

```
across=variables,time
```

9.5 Attach

Use `attach` to attach an Application Server database. Equivalent to the `ATTACH` command in Application Server.

Syntax

```
attach=<database>
```

The following section provides an overview of the variables.

Table 28

Variable	Syntax	Description
<database>	attach=<database>	Name of Application Server database to attach.

Example

```
attach=APLIB
```

9.6 Charset

Changes the character set for the transaction.

Note

JPIP only supports the UTF-8 charset.

Features

Syntax

```
charset=<string>
```

The following section provides an overview of the variables.

Table 29

Variable	Syntax	Description
`<string>`	`charset=<string>`	Type of charset. For example, `iso-8859-`.

9.7 Context

Specifies the name of a model connection to use. The model connection must already be defined.

When you select the model connection from the dropdown list in the PAS Query tool, the context parameter is automatically added to the *Address* box as `context=<name>&` so you do not need to specify it. A model connection is a mapping of Web user authentication information and Application Server user access information. You use model connections to implement security in Interactive Publisher. Each model connection controls which users can access which types of information when they run your application.

Features

Syntax

```
context=<model connection>
```

The following section provides an overview of the variables.

Table 30

Variable	Syntax	Description
`<model connection>`	`context= <model connection>`	Name of the model connection

Example

```
context=juice
```

9.8 Decimalpoint

Specifies the character to use as a decimal point when directing a view of the data to a table, delimited text file, or a cellset in a custom template. Also use the `view=format` parameter with this parameter. The `decimalpoint` parameter controls the output for this transaction.

> **Note**
>
> This setting has no effect in the application, which use the decimal point character set in the Application Server dimensional model instead.

Features

> **Syntax**
>
> ```
> decimalpoint=<char> | locale | server
> ```

The following section provides an overview of the variables.

Table 31

Variable	Syntax	Description
`<char>`	`decimalpoint=<char>`	Overrides any decimal point set in the Application Server database with the character specified.
`locale`	`decimalpoint=locale`	Uses the Interactive Publisher server machine's locale setting.
`server`	`decimalpoint=server`	Uses the decimal point character set by the Application Server dimensional model on the server.

Example

```
execute=smplproc;external&decimalpoint=.&view=format&result=table
```

9.9 Delimiter

Specifies the character to use as a separator when directing a view of the data to a delimited text file.

Features

> **Syntax**
>
> ```
> delimiter=<char>
> ```

The following section provides an overview of the variables.

Table 32

Variable	Syntax	Description
<char>	delimiter=<char>	Overrides any decimal point set in the Application Server database with the character specified.

Example

```
mime=application/vnd.ms-excel&delimiter=,&result=delimited
```

9.10 Detach

Detaches a database. Same as the DETACH command in Application Server.

Features

Syntax

```
detach=<database>
```

The following section provides an overview of the variables.

Table 33

Variable	Syntax	Description
<database>	detach=<database>	Name of Application Server database to detach.

Example

```
detach=APLIB
```

9.11 Dimension

Selects and drills on a dimension and member. This parameter is useful when you create a form in which you want to vary one of the values based on user input. In a form, you should group these three parameters as one triplet if you reference multiple dimensions in this manner.

Features

Syntax

```
dimension=<dimension>[&<dimension>=<member>][&drill=[UP | DOWN | NONE]]
```

The following section provides an overview of the variables.

Table 34

Variable	Syntax	Description
`<dimension>`	`dimension=<dimension>`	Dimension to drill on.
`<member>`	`<dimension>=<member>`	Member of the dimension to drill on.
`UP`	`drill=UP`	Drills up on the dimension member.
`DOWN`	`drill=DOWN`	Drills down on the dimension member.
`NONE`	`drill=NONE`	Selects the dimension member. This is the default setting.

Example

```
dimension=product&product=juice&drill=down
```

9.12 Display

Displays an Application Server report with only the last specified down dimension. Executes the `DISPLAY` command in Application Server. This parameter is typically combined with the parameter `result=report`.

Features

Syntax

```
display=<report name>[:width][:length]
```

The following section provides an overview of the variables.

Table 35

Variable	Syntax	Description
`<report name>`	`display=<report name>`	Name of the Application Server report to display.
`<width>`	`display=<report name>:<width>`	Specifies the maximum number of characters that can appear on a line

Variable	Syntax	Description
<length>	display=<report name>:<length>	Specifies the maximum number of lines that can appear on a report.

Example

```
execute=smplproc;external&display=basica&result=display
```

9.13 Down

Displays the specified dimensions down the side of each page. Equivalent to the DOWN command in Application Server.

Features

Syntax

```
down=<dimension>[,<dimension>]
```

The following section provides an overview of the variables.

Table 36

Variable	Syntax	Description
<dimension>	down=<dimension>,<dimension>	Dimensions to display down the page. Each dimension is separated by a comma.

Example

```
down=product
```

9.14 Dql

Executes any command in Application Server's dimensional query language. This parameter is analogous to the command line in Application Server. This parameter is intended for future use in remote administration and may aid in the development of Web applications. Unlike other parameters, commands executed with this parameter are not saved.

> **Note**
>
> During development of an application, you may need to use this parameter to perform an operation not covered by another parameter. However, you should not provide access to this parameter in an application.

Features

> **Syntax**
>
> `dql=<command>`

The following section provides an overview of the variables.

Table 37

Variable	Syntax	Description
`<Application Server command>`	`dql=<command>`	Application Server command you want to execute.

Example

```
dql=show+settings&result=text
```

9.15 Execute

Executes an Application Server stored procedure.

Features

> **Syntax**
>
> `execute=<procedure name>`

The following section provides an overview of the variables.

Table 38

Variable	Syntax	Description
`<procedure name>`	`Execute=<procedure name>`	Name of Application Server procedure to execute.

Example

```
execute=smplproc;external&result=text
```

9.16 Exhibit

Displays information about sets, variables, and dimension members. Equivalent to the `EXHIBIT` command in Application Server.

Features

Syntax

```
exhibit=<Application Server EXHIBIT command parameters>
```

The following section provides an overview of the variables.

Table 39

Variable	Syntax	Description
`<Application Server EXHIBIT command parameters>`	`exhibit=<exh parameter>`	Specify any `EXHIBIT` command parameter to execute it.

Example

```
exhibit=dimension+PRODUCT&result=text
```

9.17 Labellen

Truncates row and column labels.

Features

Syntax

```
labellen=<number>
```

The following section provides an overview of the variables.

Table 40

Variable	Syntax	Description
`<number>`	`labellen=<number>`	Number of characters to truncate. If the row or column labels are larger than the number of `LabelLen` characters, the label is cut off at the specified number, and ends with a tilde character (~). Use `LabelLen` with `result=table` or `result=report`.

9.18 Maxcolumns

Specifies the maximum number of data columns to retrieve in a query. Use this parameter for Excel Add-In implementations where worksheets are limited to 256 columns. By specifying a number less than 256 for Excel Add-In implementations, it avoids buffer issues when a user tries to display more than 256 columns across the page. Use `maxcolumns` with a `result=cellset` parameter.

Features

Syntax

```
maxcolumns=<number>
```

The following section provides an overview of the variables.

Table 41

Variable	Syntax	Description
`<number>`	`maxcolumns=<number>`	Maximum number of columns to retrieve from a large query.

9.19 Maxrows

Specifies the maximum number of rows to retrieve in a query. When a query retrieves the specified number, the query is stopped. Use `maxrows` with a `result=table`, `result=report`, `result=delimited`, or `result=stream` parameter.

Note

The visual display of data in the Web browser and the actual number of rows in the source .html file may not be the same. For example, if you specify `result=report &maxrows=36`, you might only see 28 rows in the Web browser even though the source .html file has all 36 rows of data. This is because the source data may contain blank rows rendered in the .html as `<TR ALIGN="RIGHT"> </TR>` but are not rendered in the browser.

Features

Syntax

```
maxrows=<number>
```

The following section provides an overview of the variables.

Table 42

Variable	Syntax	Description
`<number>`	`maxrows=<number>`	Maximum number of rows to retrieve from a large query.

Example

```
execute=smplproc;external&result=stream&maxrows=200
```

9.20 Mime

Specifies how to format the data in cases other than displaying query results in the browser. For example you can use `result=delimited` and set a mime type to `Excel` and automatically launch results to Excel.

Features

Syntax

```
mime=<mime-type>/<mime-subtype>
```

The following section provides an overview of the variables.

Table 43

Variable	Syntax	Description
`<mime-type>`	`mime=<mime-type> /<mime-subtype>`	Specify a mime type.
`<mime-subtype>`	`mime=<mime-type> /<mime-subtype>`	Specify a mime subtype.

Example

```
mime=application/vnd.ms-excel&delimiter=,&result=delimited
```

9.21 Missing

Displays missing values with the specified value, up to 7 characters, when using the `result=table` method of displaying data. If you do not specify any value, missing values are represented as blanks.

Features

Syntax

```
missing=<value>
```

The following section provides an overview of the variables.

Table 44

Variable	Syntax	Description
<value>	missing=<value>	Value to display to represent missing values.

Example

missing=-

9.22 Order

Changes the order in which dimension members are displayed. Equivalent to the ORDER command in Application Server.

Features

Syntax

```
order=<dimension>+[alphabetic | labels]
```

The following section provides an overview of the variables.

Table 45

Variable	Syntax	Description
<dimension>	order=<dimension>	Dimension whose members to order.
alphabetic	order=<dimension>+ alphabetic	Orders members alphabetically.
labels	order=<dimension>+ labels	Orders members alphabetically based on their labels.

Example

```
order=CUSTOMER+ALPHABETIC
```

9.23 Pivot

Rotates the dimensions displayed across and down a table.

> **Note**
> The swapping of across and down dimensions occurs once only because this selection is saved.

Features

> **Syntax**
> `pivot=[on | off]`

The following section provides an overview of the variables.

Table 46

Variable	Syntax	Description
`on`	`pivot=on`	Swaps across dimensions with down dimensions.
`off`	`pivot=off`	Turns off all pivoting capabilities.

Example

```
initexecute=smplproc;external&across=product&down=time,var&pivot=on
```

9.24 Result

Controls the resulting output of your operation.

Features

> **Syntax**
> `result=[cellset|chart|context|delimited|remoteuser|report|stream|table|text|version]`

The following section provides an overview of the variables.

Table 47

Variable	Syntax	Description
`cellset`	`result=cellset`	Used with the `view=format` parameter to format the data. When `result=cellset`, you can use the `thousandsep` and `decimalpoint` parameters to override the thousands separator character and decimal point that is set in Application Server when using a custom template.

Variable	Syntax	Description
`chart`	`result=chart`	Charts the first matrix of the last down dimension and last across dimension in an Application Server view. You can chart additional matrices using the `result=cellset` parameter. Calculated columns that exist in the first matrix are not charted. The `result=chart` parameter does not chart information from an Application Server report. JPIP produces the same results for `result=chart | delimiited | table | text`
`context`	`result=context`	Selects the first available model connection for the current session. **Note** If you are setting up a default logon, and you do not specify `context=<name>` to start Interactive Publisher, `result=context` is used to select the first available model connection.
`delimited`	`result=delimited`	Downloads a delimited text file of the current view, which can be used in applications such as Microsoft Excel. When `result=delimited`, you can also use the `thousandsep` and `decimalpoint` parameters to override the thousands separator character and decimal point in Application Server. JPIP produces the same results for `result=chart | delimiited | table | text`. **Note** Use `result=delimited` with `view=script` to put a matrix into a tab-delimited JavaScript array variable called `PipResult`. Use this to export data in a report format from a matrix-oriented template such as Navigator.
`remoteuser`	`result=remoteuser`	Returns the Interactive Publisher remote user name. Use this parameter in a custom application when you want to create unique names by logon name in Interactive Publisher.
`report`	`result=report`	Produces HTML-formatted output of an Application Server report. This parameter is combined with the display parameter. Row and column labels may be truncated if you set a value using `labellen` or `labelview`.

Variable	Syntax	Description
stream	result=stream	Generates nonformatted output for use mainly with Basic Scripting, ActiveX, Java, or JavaScript programs.
table	result=table	Produces HTML-formatted output of an Application Server matrix. This result is analogous to the LIST command in Application Server. The view defaults for a table are: across, down, outline, suppress, and parent. JPIP produces the same results for result=chart | delimiited | table | text When using result=table, you can also use the thousandsep and decimalpoint parameters to override the thousands separator character and decimal point that is set in Application Server. If a dimension has multiple hierarchies, use result=table to display data where you can drill up or down on a hierarchy. Row and column labels may be truncated if you set a value using the labellen parameter or labelview parameter.
text	result=text	Transforms Application Server output into HTML-formatted text. JPIP produces the same results for result=chart | delimiited | table | text
version	result=version	Displays information about the current versions of Interactive Publisher and Application Server. This option is useful for testing connectivity. You must include a context=<model_connection> when using result=version.

Example

```
execute=smplproc;external&result=chart
execute=smplproc;external&attach=aplib&display=basica;aplib&result=report
execute=smplproc;external&result=stream
execute=smplproc;external&result=table
execute=smplproc;external&result=text
context=demo&result=version
```

9.25 Select

Selects a particular dimension or a dimension's members for data input, reporting, or output. Equivalent to the `SELECT` statement in Application Server.

Features

Syntax

```
select=dimension+<dimension name>
```

The following section provides an overview of the variables.

Table 48

Variable	Syntax	Description
`<dimension name>`	`select=dimension+ <dimension name>`	Name of the dimension to selection.

Example

```
select=dimension+product&result=text
```

9.26 Set

Defines the default set for an Application Server session. Equivalent to the `SET` command in Application Server.

Features

Syntax

```
set=<set command keywords>
```

The following section provides an overview of the variables.

Table 49

Variable	Syntax	Description
`<set command keywords>`	`set=<set command keywords>`	`SET` command keywords. For information, see the Help in Application Server Administrator.

Example

```
execute=smplproc;externalset=period+monthly&result=text
```

9.27 Super

Starts the Application Server Supervisor subsystem, where you can manage database and user information in `MASTERDB`. Equivalent to the `SUPERVISOR` command in Application Server.

Features

Syntax

```
super=<supervisor commands>
```

The following section provides an overview of the variables.

Table 50

Variable	Syntax	Description
`<supervisor commands>`	`super=<supervisor commands>`	Any Application Server `Supervisor` command

Example

```
super=create+user+pip
```

9.28 Textvarprefix

Allows you to use Application Server text variables in the Navigator template. This parameter identifies which Application Server variables are to be treated as text variables according to the prefix in the `SET VARIABLE BEFORE` command. If you do not specify this parameter and text variables exist, then Interactive Publisher treats any variables with a tilde (~) prefix as text variables.

Use this parameter with any of the following forms of output: `result=jscript`, `result=report`, and `result=delimited`, and `result=jscript`. It is not used in `result=cellset`.

Features

Syntax

```
textvarprefix=<value>
```

The following section provides an overview of the variables.

Table 51

Variable	Syntax	Description
<value>	textvarprefix= <value>	Same text variable prefix as the value specified in the Application Server SET VARIABLE BEFORE command.

9.29 Thousandsep

Specifies the character to use as a thousand when directing a view of the data to a table, delimited text file, or cellset in a custom template. You can also use the view=format parameter with this parameter. The thousandsep parameter controls the output for this transaction.

> **Note**
> This setting has no effect in the application, which use the thousands separator character set in the Application Server dimensional model.

Features

Syntax

```
thousandsep=[<char>|none|locale|server]
```

The following section provides an overview of the variables.

Table 52

Variable	Syntax	Description
<char>	thousandsep=<char>	Overrides any thousands separator set in the Application Server database using the value specified for <char>.
None	thousandsep=none	Omits using a thousand separator.
Locale	thousandsep=locale	Uses the locale setting of the Interactive Publisher machine.
Server	thousandsep=server	Uses the thousands separator character set by the Application Server dimensional model on the server.

Example

```
execute=smplproc;external&view=format&thousandsep=,&result=table
```

9.30 Time

Defines a periodicity that applies to all subsequent operations. Equivalent to the command `SET PERIOD` in Application Server.

Features

Syntax

```
time=[<yy[/mm[/dd]]-yy[/mm[/dd]]> | yty | yrago | current | previous | last]
```

The following section provides an overview of the variables.

Table 53

Variable	Syntax	Description
`<yy[/mm[/dd]]-yy[/mm[/dd]]>`	`time=<yy[/mm[/dd]]-yy[/mm[/dd]]>`	Range of dates in the form `yy[/mm[/dd]] - yy[/mm[/dd]]`. For example, 08/12/10-08/12/31 and 07-08 is all of 2007 and 2008.
`yty`	`time=yty`	The latest period back to the equivalent period one year ago.
`yrago`	`time=yrago`	The latest period one year ago.
`current`	`time=current`	The period set with a `SET LATEST` command.
`previous`	`time=previous`	The previous period.
`last`	`time=last`	The period set with a `SET LATEST` command.

Example

```
time=yearly
```

9.31 View

Controls what is displayed in a table. You can specify multiple view parameters in the *Address* box of the PAS Query tool. For example, `view=parent&view=format &view=linklabels&...` The default view options for a table are: `across`, `down`, `outline`, `noparent`, `noprotect`, and `suppress`.

Features

Syntax

```
view=[[no]across |  [no]down | cachecustom | custom | editcustom | format |
linklabels | linkrowlabels | linkcollabels | linkcells | meta | nbsp |
[no]outline | [no]parent | [no]protect | script | [no]suppress | time_span ]
```

The following section provides an overview of the variables.

Table 54

Variable	Syntax	Description
`[no]across`	`view=across` `view-noacross`	Displays column headings. Use no to omit column headings.
`[no]down`	`view=down` `view-nodown`	Displays row headings. Use no to omit row headings.
`cachecustom`	`view=cachecustom`	Caches User-Defined Hierarchies, which are not cached by default. If you know that a User-Defined Hierarchy's definitions will not change, use `view=cachecustom` parameter with `cache=yes`. This allows Interactive Publisher to cache the Application Server result sets that have User-Defined Hierarchies, and allows for a quicker response time.
custom	`view=custom`	Allows a user to create a user-defined hierarchy
`editcustom`	`view=editcustom`	Allows a user to edit or delete a user-defined hierarchy
`format`	`view=format`	Allows you to use the `thousandsep` and `decimalpoint` parameters to override the thousands separator character and decimal point set in Application Server.
`linklabels`	`view=linklabels`	Uses the onClick handler on the HTML <HREF> attribute to determine the action that an HREF link makes when a user clicks a column or row label in a table or report. Use this parameter to customize what happens when a user clicks any type of label, such as a measure or time. linklabels adds the dimension name, dimension member, and drill mode.
`linkrowlabels`	`view=linkrowlabels`	Uses the onClick handler on the HTML <HREF> attribute to determine the action that an HREF link makes when a user clicks a row label in a table or report. Use this parameter to customize what happens when a user clicks any type of row label, such as a measure, dimension, or time. Adds the following information about the view: dimension name, dimension member, and drill mode. It also performs the same function as `view=meta`.

Variable	Syntax	Description
`linkcollabels`	`view=linkcollabels`	Uses the onClick handler on the HTML <HREF> attribute to determine the action that an HREF link makes when a user clicks a column label in a table or report. Use this parameter to customize what happens when a user clicks any type of column label, such as a measure, dimension, or time. `Linkcollabels` adds the following information about the view: dimension name, dimension member, and drill mode.
`linkcells`	`view=linkcells`	Uses the onClick handler on the HTML <HREF> attribute to determine the action that an HREF link makes when a user clicks a data cell in a table or report. The `linkcells` value adds the following information about the view: dimension name, dimension member name, drill mode, column dimension name, column member name, column drill mode, and cell value.
`meta`	`view=meta`	Performs the same function as `view=script` and adds the following meta information about the view: long or short name, period, periodicity, timeset, and set latest date, across dimensions, down dimensions, and selected dimensions that affect the view, and the number of across, down, and selected dimensions. **Note** `View=meta` must be included when using the `onerror=jscript` parameter.
`nbsp`	`view=nbsp`	Adds a space to a report produced by a `result=report` parameter. Use the `nbsp` option to indent row labels in the report. Client browsers trim spaces in a report and left-justify any information. The `view=nbsp` parameter allows you to add the spaces into the report.
`[no]outline`	`view=outline` `view=nooutline`	Displays the headings as indented in a single column. Use no to display headings in a separate column.
`[no]parent`	`view=parent` `view=noparent`	Displays all higher levels of a dimension when drilling down. Use no to display only the drilled-down level.
`[no]protect`	`view=protect` `view=noprotect`	Disables auto-drilling. Use no to enable auto-drilling.
`script`	`view=script`	Uses the onClick handler on the HTML HREF attribute to determine the action that an HREF link makes when a user clicks a dimension member name in a table. Use this

Variable	Syntax	Description
		parameter to customize dynamic drill actions. To customize what happens when a user clicks a dimension member name in a report, use a `view=linkx` parameter. **Note** You can also use `view=script` with `result=delimited` to put an Application Server matrix into a tab-delimited JavaScript array variable called `PipResult`. Use this to export data in a report format from a matrix-oriented application such as Navigator.
`[no]suppress`	`view=suppress` `view=nosuppress`	Shows or hides the display of zeros.
`timespan`	`view=timespan`	Shows the start and end date tol be used in the current view when using `display=` and `result=report`.

Example

```
execute=smplproc;external&result=table&view=noacross
```

9.32 Varname

Sets the value of a control variable. Setting the control variable(s) occurs immediately after database commands. Equivalent to the commands `SET CONTROL` and `CLEAR` in Application Server. A control variable exists in Application Server only for the duration of the HTTP transaction.

Features

Syntax

```
&varname=<value>   or   _varname=<value>
```

The following section provides an overview of the variables.

Table 55

Variable	Syntax	Description
`<value>`	`&varname=<value>`	Value for the control variable. You can use either an ampersand (&) or an underscore (_) with this parameter.

Variable	Syntax	Description
		Interactive Publisher strips the ampersand or underscore character. The variable name must start with an underscore, otherwise it is ignored. The ampersand follows the Application Server convention for a control variable. To use the ampersand in this parameter, specify it by its hexadecimal encoding, %26. If you are using control variables with JavaScript or VBScript, use an underscore with this parameter, not an ampersand. To use a control variable that begins with an underscore, specify it with two underscores (__).

Example

```
%26varname=directory+full&dql=%26varname&result=text
_varname=directory+full&dql=_varname&result=text
```

10 Administrator Parameters

This section covers all the administrator parameters and values you can enter in the *Address* box of the PAS Query tool. For more information about the PAS Query tool, see PAS Query Tool [page 48].

10.1 Pipadmin

This syntax shows the complete `pipadmin` syntax for stopping, restarting Interactive Publisher, and interacting with the Interactive Publisher registry.

> **Note**
> You can use the JPIP Session Monitor to restart Interactive Publisher, restart a session, shut down an instance, and refresh the tree, session, and instance status. You can also review the show the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time. The JPIP Session Monitor is located on the *Tools* page at `http://<nw_server>:<port>/strategy/tools`.

Features

> **Syntax**
> `pipadmin=<options>`

The following section provides an overview of the variables.

Table 56

Variable <options>	Syntax	Description
`isadmin`	`pipadmin=isadmin`	Returns a `0` or `1` to identify whether the Interactive Publisher user is an administrator or not. A value of `1` means the user is an administrator. A value of `0` means the user is not an administrator.
`Restart`	`pipadmin=restart`	Restarts Interactive Publisher and all Interactive Publisher subsystems and rereads data from the Interactive Publisher Registry/cache.
`shutdown`	`Pipadmin=shutdown`	Stops Interactive Publisher and all subsystems.
`TestPasConn! <servername> '<port>' <serveruser> '<serverpassword>'`	`Pipadmin= TestPasConn! <servername> '<port>' <serveruser> '<serverpassword>' <service> '<inifile>'`	Tests the connection to Application Server. • `<servername>` — Server name where Application Server daemon is listening. • `port` — Server port where Application Server daemon is listening.

Variable <options>	Syntax	Description
<service> '<inifile>' <cmdline> '<pasuser>' <paspwd> '<usedb>' [<format>]	<cmdline> '<pasuser>' <paspwd> '<usedb>' [<format>]	• <serveruser > — User Id that Application Server daemon uses to launch Application Server process. • <serverpassword> — Password that Application Server daemon uses to launch Application Server process. • <service> — Service name that Application Server daemon uses to launch Application Server process. • <inifile> — Application Server inifile. • <cmdline> — Cmdline to pass to Application Server. • <pasuser> — Application Server user name. • <paspwd> — Application Server password. • <usedb> — use database that Application Server process loads as the Application Server USE DB. • <format> — Format for the return code. Can be either jscript, stream, or html.

10.2 Pipadminex

Creates and manages users and model connections.

Syntax

```
pipadminex=add_tc!<webuser>/<model_connection>'<pasuser>'<security>
pipadminex=add_td_elems!<pasuser>/<model_connection>'<pasuser>'
 <paspassword>'<servermachine>'<serverport>'<pasinifile>'<serveruser>'
 <serverpassword>'<maxinstances>'<mininstances>'<passervice>'<pasdb>'
 <security>'<webuser>
pipadminex=bulk_tc!<webuser>/<model_connection>'<pasuser>'<security>
pipadminex=deletecontext!<model_connection>
pipadminex=expandcontext&context=<model_connection>
pipadminex=expandsession!<model_connection>'<pasuser>
pipadminex=get_td_elems!<pasuser>/<model_connection>
pipadminex=listcontexts
pipadminex=listsessions
pipadminex=remove_tc!<pasuser>/<model_connection>
pipadminex=remove_td!<pasuser>/<model_connection>
pipadminex=renamecontext!<new_model_connection>'<old_model_connection>
pipadminex=renamecontext!<new_model_connection>/<newpasuser>'
 <old_model_connection>/<oldpasuser>
```

10.2.1 Adding a Web Authentication User to a Model Connection

Adds a Web authentication user to an existing model connection.

> **Note**
> Another method of performing the same function is to use *Administration* > *Manage Models* in the administration application.

This parameter is similar to `pipadminex=bulk_tc!`, except that `add_tc` allows you to add one user at a time, while the `bulk_tc` setting allows you to add multiple users at a time.

Usernames and passwords are stored as encrypted in the SAP NetWeaver System Database.

Prerequisites

Before you can add a user to a model connection, you must first create the model connection using `pipadmin=add_td_elems`. This parameter creates a record in the SAP NetWeaver System Database.

Features

> **Syntax**
> `pipadminex=add_tc!<webuser>/<model_connection>'<pasuser>'<security>`

The following section provides an overview of the variables.

Table 57

Variable	Description
`<webuser>`	Logon name for the Web authentication user
`<model_connection>`	Name of the model connection
`<pasuser>`	Name of the Application Server user to log on to Application Server **Note** Make sure the Application Server user is a user of the dimensional model you will be adding to the model connection definition. **Note** If you are running Application Server on a Linux/UNIX server, the UNIX or Linux user specified must have appropriate access to the `$ORACLE_HOME` directories, particularly `$ORACLE_HOME/lib32`. It is highly recommended to grant read and execute access to the directory structure under `$ORACLE_HOME` to ensure that there are no problems with Application Server connections to Oracle. You can run the `$ORACLE_HOME/install/changePerm.sh` script to ensure that the account used for client/server

Variable	Description
	connections has the appropriate access privileges to the Oracle client software. This script establishes Read access to most of the directories in `$ORACLE_HOME`.
`<security>`	Security for the session: `1`=security, `0`=no security

Example

This example shows the first part of the parameter:

```
pipadminex=add_tc!ellend/juice'admin'0
```

10.2.2 Creating a Model Connection

Use this parameter with `pipadminex=add_tc!` or `pipadmin=bulk_tc!` to create a model connection and add users to them in a batch process.

Each model connection is stored as an entry in the SAP NetWeaver System Database. Usernames and passwords are stored as encrypted in the SAP NetWeaver System Database.

> **Note**
>
> Another method of performing the same function is to use *Administration* > *Manage Models* in the administration application.

Features

Syntax

```
pipadminex=add_td_elems!<pasuser>/<model_connection>'<pasuser>'
<paspassword>' <servermachine>'<serverport>'<pasinifile>'<serveruser>'
<serverpassword>' <maxinstances>'<mininstances>'<passervice>'<pasdb>'
<security>'<webuser>
```

The following section provides an overview of the variables.

Table 58

Variable	Description
`<pasuser>`	Name of the Application Server user to log on to Application Server. The default is Guest, which provides user privileges, but not Supervisor privileges. Do not specify the same Application Server user name for multiple dimensional models.

Variable	Description
	Note Make sure the Application Server user is a user of the dimensional model you will be adding to the model connection definition. **Note** If you are running Application Server on a Linux/UNIX server, the UNIX or Linux user specified must have appropriate access to the `$ORACLE_HOME` directories, particularly `$ORACLE_HOME/lib32`. It is highly recommended to grant read and execute access to the directory structure under `$ORACLE_HOME` to ensure that there are no problems with Application Server connections to Oracle. You can run the `$ORACLE_HOME/install/changePerm.sh` script to ensure that the account used for client/server connections has the appropriate access privileges to the Oracle client software. This script establishes Read access to most of the directories in `$ORACLE_HOME`.
`<model_connection>`	Name of the model connection. This name must match the model connection name you specified earlier as the model connection value. Specify only alphanumeric characters (a-z, 0-9) up to a maximum size of 64 characters. You must not use special characters in the name.
`<pasuser>`	Name of the Application Server user to log on to Application Server. This name must be the same one you specified earlier as the `pasuser` value.
`<paspassword>`	Password associated with the Application Server user ID. If the user is Guest, you do not need to specify a password. If there is no password, leave a space. The empty argument for `paspassword` uses a space that the server should translate to a '+' character before it gets to the `pipadmin` code.
`<servermachine>`	Name of the machine on which Application Server is installed. The default is LocalHost, when no server sections exist in the `LSSERVER.INI` file. The list of the available machines comes from the `LSSERVER.INI` file in the Microsoft Windows directory.
`<serverport>`	Communications port for the UNIX or Microsoft Windows server machine running Application Server. This is the server port where the Application Server daemon is listening. The default is 8325
`<pasinifile>`	.INI file to use. The default is `LSSERVER.INI`.
`<serveruser>`	UNIX or Microsoft Windows Server user name authorized to run Application Server. This is the username that Application Server daemon uses to launch the Application Server process.
`<serverpassword>`	UNIX or Microsoft Windows Server password for the user name. This is the password that Application Server daemon uses to launch the Application Server process.
`<maxinstances>`	Maximum number of copies, or logons, of Application Server to allow for this model connection. The default setting is `5`. You can set up to 255 instances.

Variable	Description
	You must specify a number that is equal to or lower than the number defined for the Application Server user. For example, if an Application Server user is allowed 5 instances, you must enter a number that is no higher than 5. To find out the maximum number of instances available to an Application Server user, issue the `SUPERVISOR SHOW USERS` command in Application Server. To change the number of instances available to an Application Server user, issue the `SUPERVISOR CHANGE USER` command with the `MAXLOGIN` keyword. For information about these commands, see the Application Server online Help.
`<mininstances>`	Number of copies of Application Server to start up with this model connection. The default setting is `0`, which means that no copies start until the first query is executed.
`<passervice>`	Service specified in the .INI file. The default is `PILOT`
`<pasdb>`	Application Server `USE` database to use for this model connection. `Security` specifies whether to use Application Server security for this session: `1` = yes, `0` = no. **Note** You cannot use an Application Server dimensional model in the strategy management application that contains text variables. **Note** You must already have an Application Server model created.
`<security>`	Security for the session: `1`=security, `0`=no security.
`<webuser>`	Logon name for the Web authentication user.

Example

```
pipadminex=add_tc!webuser01/juice'admin&pipadminex=add_td_elems!admin/juice'
admin'kendall'8325'lsserver.ini'webuser01'$$$$$'5'1'PILOT'juice'0'pipadmin
```

10.2.3 Adding Multiple Users to a Model Connection

Adds multiple users to a model connection.

This parameter is similar to `pipadminex=add_tc`, except that `add_tc` allows you to add one user at a time, while `bulk_tc` allows you to add multiple users at a time.

Use this parameter with `pipadmin=add_td_elems!` to create model connections and add users as a batch process. This parameter creates a record in the SAP NetWeaver System Database.

Usernames and passwords are stored as encrypted in the SAP NetWeaver System Database.

> **Note**
>
> Another method of performing the same function is to use *Administration* > *Manage Models* in the administration application.

Prerequisites

Before you can add a user to a model connection, you must first create the model connection using `pipadmin=add_td_elems`. This parameter creates a record in the SAP NetWeaver System Database.

Features

> **Syntax**
>
> `pipadminex=add_tc!<webuser>/<model_connection>'<pasuser>'<security>`

The following section provides an overview of the variables.

Table 59

Variable	Description
`<webuser>`	Logon name for the Web authentication user. Add multiple users who can access this model connection by separating each name with a comma.
`<model_connection>`	Name of the model connection.
`<pasuser>`	Name of the Application Server user to log on to Application Server. **Note** Make sure the Application Server user is a user of the dimensional model you will be adding to the model connection definition. **Note** If you are running Application Server on a Linux/UNIX server, the UNIX or Linux user specified in the PAS User text box must have appropriate access to the $ORACLE_HOME directories, particularly $ORACLE_HOME/lib32. It is highly recommended to grant read and execute access to the directory structure under `$ORACLE_HOME` to ensure that there are no problems with Application Server connections to Oracle. You can run the `$ORACLE_HOME/install/changePerm.sh` script to ensure that the account used for client/server connections has the appropriate access privileges to the Oracle client software. This script establishes Read access to most of the directories in `$ORACLE_HOME`.
`<security>`	Security for the session: `1`=security, `0`=no security.

Example

```
pipadminex=bulk_tc!ellend,toml,scottl/juice'admin'0
```

10.2.4 Deleting Model Connection Sessions

Deletes sessions associated with a model connection.

Features

Syntax

```
pipadminex=DeleteContext!<model_connection>
```

The following section provides an overview of the variables.

Table 60

Variable	Description
<model_connection>	Name of the model connection whose sessions you want to delete.

10.2.5 Showing Users in Model Connection Sessions

Returns a tab-delimited string of all Interactive Publisher users in a model connection for all Application Server sessions. The rows are sorted in ascending order by server name. Each row shows the server name, Application Server user, and Interactive Publisher user.

Features

Syntax

```
pipadminex=ExpandContext&context=<model_connection>
```

The following section provides an overview of the variables.

Table 61

Variable	Description
<model_connection>	Name of the model connection to expand.

10.2.6 Showing Users in an Application Server Session

Returns a tab-delimited string of all Interactive Publisher users in an Application Server session for a specified model connection. The rows are sorted in ascending order by Interactive Publisher user name.

Features

Syntax

```
pipadminex=ExpandSession!=<model_connection>'<pasuser>
```

The following section provides an overview of the variables.

Table 62

Variable	Description
<model_connection>	Name of the model connection to use.
<pasuser>	Name of the Application Server user whose Interactive Publisher users you want to list.

10.2.7 Showing Application Server Session Information

Returns a single quote-delimited string of Application Server session information.

Features

Syntax

```
pipadminex=get_td_elems!<pasuser>/<model_connection>
```

The following section provides an overview of the variables.

Table 63

Variable	Description
<model_connection>	Name of the model connection for the session.
<pasuser>	Name of the Application Server user.

Results

The single quote-delimited string includes the following information:

Syntax

```
0||1 (error or OK)
pasuser/model_connection
pas_username
```

```
pas_pwd
server_machine
server_port
server_inifile
server_username
server_pwd
max_instances
min_instances
service
use_db
0 or 1 (not using security or using security)
```

10.2.8 Displaying Model Connection Names

Returns a tab-delimited string of model connection names.

Features

Syntax

```
pipadminex=listcontexts
```

10.2.9 Showing Connection Names and User Names

Returns a tab-delimited string of model connection names and Application Server user names, where each model connection appears in a new row, sorted in ascending order by model connection name.

Features

Syntax

```
pipadminex=listsessions
```

10.2.10 Deleting a User from a Model Connection

Deletes a user from a model connection.

Features

Syntax

```
pipadminex=remove_tc!<user>/<model-connection>
```

The following section provides an overview of the variables.

Table 64

Variable	Description
<model_connection>	Name of the model connection whose user you want to delete.
<user>	Web authentication username you want to remove from the model connection.

10.2.11 Deleting a Session

Deletes a session based on the model connection and Application Server username.

Features

Syntax

```
pipadminex=remove_td!<pasuser>/<model-connection>
```

The following section provides an overview of the variables.

Table 65

Variable	Description
<model_connection>	Name of the model connection whose session you want to delete.
<user>	Application Server username for the model connection.

10.2.12 Renaming a Model Connection

Renames a model connection.

Features

Syntax

```
pipadminex=renamecontext!<new_model_connection>'<old_model_connection>
```

The following section provides an overview of the variables.

Table 66

Variable	Description
<new_model_connection>	New name for the model connection.
<old_model_connection>	Name of model connection to rename.

10.2.13 Renaming a Session

Renames an Application Server user/model connection session.

Features

Syntax

```
pipadminex=renamecontext!<new_model_connection>/
<new_pas_user>'<old_model_connection>/<oldpasuser>
```

The following section provides an overview of the variables.

Table 67

Variable	Description
<new_model_connection>	New name for the model connection.
<new_pas_user>	Name of Application Server user to use in the new model connection session.
<old_model_connection>	Name of model connection to rename.
<old_pas_user>	Name of Application Server user to use in the new model connection session.

10.3 Pipinfo

Displays information about the system. Shuts down, updates, and restarts a session based on an Application Server username and model connection.

Note

You can also use the JPIP Session Monitor to restart Interactive Publisher, restart a session, shut down an instance, and refresh the tree, session, and instance status. You can also review the show the sessions, their status, number of busy instances, number of total instances, number of transactions, average process time, and average transaction time. The JPIP Session Monitor is located on the *Tools* page at `http://<nw_server>:<port>/strategy/tools`.

Features

Syntax

```
pipinfo=
[ shutdown&session=[<pasuser>/]<model_connection> |
restart&session=[<pasuser>/]<model_connection> |
update&session=[<pasuser>/]<model_connection> ]
```

The following section provides an overview of the variables.

Table 68

Variable	Syntax	Description
`shutdown&session= <model_connection>`	`pipinfo=shutdown& session= <model_connection>`	Shuts down all sessions connected with the specified model connection in the Interactive Publisher server.
`shutdown&session= <pasuser> <model_connection>`	`pipinfo=shutdown& session=<pasuser> <model_connection>`	Shuts down a specific session based on the specified model connection and Application Server user.
`restart&session= <model_connection>`	`pipinfo=restart& session= <model_connection>`	Restarts all sessions connected with the model connection
`restart&session= <pasuser> <model_connection>`	`pipinfo=restart& session=<pasuser> <model_connection>`	Restarts a specific session based on the specified model connection and Application Server user.

`update`	`pipinfo=update& session= <model_connection>`
	`pipinfo=update& session=<pasuser> <model_connection>`
		Allows an Interactive Publisher administrator to open an Application Server model for exclusive access for one transaction. Use this parameter in the following sequence.
		`pipinfo=shutdown& session=<pasuser> <model_connection>`
		`pipinfo=update& session=<pasuser> <model_connection>`
		`pipinfo=restart& session=<pasuser> <model_connection`

Example

This example shuts down all sessions connected with model connection juice:

```
pipinfo=shutdown&session=juice
```

The following example shuts down a specific user model connection:

```
pipinfo=shutdown&session=guest/juice
```

The following example restarts a specific user model connection:

```
pipinfo=restart&session=guest/juice
```

The following example restarts all sessions connected with model connection juice:

```
pipinfo=restart&session=juice
```

See also the `pipadmin` administrator parameter to stop and restart all the Interactive Publisher subsystems and reread data from the Registry/cache.

This sequence allows you to stop model connections associated with an Application Server model, use the model exclusively for updating purposes, and then restart the session and make it available to all model connections again. If anyone tries to use the model connection to this model during this process, an appropriate message appears. This allows you to perform an update without any connections occurring at the same time.

This result shows that the `Use` database is currently in read mode:

```
context=juice&dql=show data&result=text
```

Now you can use this parameter to shut down the session:

```
pipinfo=shutdown&session=guest/juice
```

This result shows that the `Use` database is now in Exclusive mode:

```
pipinfo=update&session=guest/juice&context=juice&dql=show data&result=text
```

Now you can use this parameter to confirm that the session is shut down:

```
context=juice&dql=show data&result=text
```

Now you can use this parameter to restarts the session:

```
pipinfo=restart&session=guest/juice
```

10.4 Pipstats

Displays session and instance information about the JPIP session.

> **Note**
>
> Another way to view information about the JPIP session is to use the JPIP Session Monitor located on the *Tools* page at `http://<nw_server>:<port>/strategy/tools`.

Features

Syntax

```
pipstats=mon_sessionlist|mon_instancelist!<as_user>/<context_name>
&format=json_object|json_array
[&attribute=<string>]
[&perfectformat]
```

The following section provides an overview of the variables.

Table 69

Variable	Syntax	Description
`mon_sessionlist`	`pipstats=mon_sessionlist`	Lists all JPIP sessions currently running or in shutdown mode. Use the `format` parameter to return the results of the Javascript code. This list shows the session data that is returned: **Name**: Name of the session **activeCount**: Number of instances whose status is PROCESSING **instanceCount**: Total instance number **status**: 1 Running, 0 Shutdown **trans**: Number of transaction executed **avgProcessTime**: Average process time of each request, counted right before transaction doRequest and right after the doRequest **avgTransTime**: Average process time of execute a transaction, counted right before transaction starts and right after transaction stops.
`mon_instancelist!<as_user>/<context_name>`	`pipstats=mon_instancelist!<as_user>/<context_name>`	Lists all JPIP instances of a JPIP session for a particular Application Server user and context. This list shows the instance data that is returned: **Name**: The name of the instance **Status**: 1 PROCESSING, 0 IDLE **Crf**: the value evaluated in LRFU algorithm, a instance with highest crf value will be returned when getting a LRFU instance. **Duration**: the time that this instance is occupied by last transaction **Trans**: number of transactions this instance served. **avgProcessTime**: Average process time of each request, counted right before transaction doRequest and right after the doRequest **avgTransTime**: Average process time of execute a transaction, counted right

Variable	Syntax	Description
		before transaction starts and right after transaction stops.
format	&format=json_object|json_array	For information, see Format [page 88].
attribute	&attribute=<string>	For information, see Attribute [page 89].
perfectformat	&perfectformat	For information, see Perfectformat [page 90].

Example

This example lists all JPIP sessions:

```
pipstats=mon_sessionlist&format=json_array
```

This example lists all JPIP instances of the session with user guest and context HFPBM, and returns the result in JSON Object format which is perfect-formatted:

```
pipstats=mon_instancelist!guest/
HFPBM&attribute=instances&format=json_object&perfectformat
```

10.4.1 Format

Formats output produced by the `pipstats` administrator parameters into the specified format, either in JSON Object mode or JSON Array mode.

Features

Syntax

```
format=[json_object | json_array]
```

The following table provides an overview of the variables.

Table 70

Variable	Syntax	Description
json_object	format=json_object	Returns the values as attributes of a JSON object. This parameter works together with the attribute parameter which defines the name of the JSON object.
json_array	format=json_array	Returns the values in array format.

Using `json_object`, to get the name value in Javascript, use `instances.name`:

```
{
```

```
instances : [
{
'name' : 'GUEST/HFPBM',
'activeCount' : '0',
'instanceCount' : '1',
'status' : '1',
'trans' : '6',
'avgProcessTime' : '1111.0',
'avgTransTime' : '0.0'
} ]
}
```

To return values in array format and to get the name value in Javascript if the returned JSON object is named as `ret`, then use `ret['name']`.

```
{
{
'name' : 'GUEST/HFPBM',
'activeCount' : '0',
'instanceCount' : '1',
'status' : '1',
'trans' : '6',
'avgProcessTime' : '1111.0',
'avgTransTime' : '0.0'
}
}
```

10.4.2 Attribute

Controls the object name of the returned `json_object`. This parameter works only with `pipstatus=<option>&format=json_object`.

Features

Syntax

```
attribute=<string>
```

The following section provides an overview of the variables.

Table 71

Variable	Syntax	Description
attribute	&attribute=<string>	Controls the object name of the returned json_object.

Example

This parameter:

```
attribute=instances
```

produces:

```
{instances:[{'name':'GUEST/HFPBM','activeCount':'0','instanceCount':'1','status' :
'1','trans':'6','avgProcessTime':'1111.0','avgTransTime':'0.0'}]}
```

This parameter:

```
attribute=others
```

produces:

```
{others:[{'name':'GUEST/HFPBM','activeCount':'0','instanceCount':'1','status' :
'1','trans':'6','avgProcessTime':'1111.0','avgTransTime':'0.0'}]}
```

10.4.3 Perfectformat

Formats the result of the JSON data in multiple lines and with indents so that it is easier to read. Use this parameter for debugging purposes.

If you do not use this parameter, the result is returned in a single line of text without any space characters. This produces the smallest result.

Features

Syntax

```
perfectformat
```

The following section provides an overview of the variables.

Table 72

Variable	Syntax	Description
perfectformat	perfectformat	Formats the result of the JSON data in multiple lines and with indents.

Example

If json is not present and you do not specify `perfectformat`, then the output appears like this:

```
{instances:[{'name':'GUEST/HFPBM','activeCount':'0','instanceCount':'1','status' :
'1','trans':'6','avgProcessTime':'1111.0','avgTransTime':'0.0'}]}
```

If json is present and you specify `perfectformat`, then the output appears like this:

```
{
instances : [
{'name' : 'GUEST/HFPBM',
'activeCount' : '0',
'instanceCount' : '1',
'status' : '1',
'trans' : '6',
'avgProcessTime' : '1111.0',
'avgTransTime' : '0.0'
} ]
}
```

Typographic Conventions

Table 73

Example	Description
<Example>	Angle brackets indicate that you replace these words or characters with appropriate entries to make entries in the system, for example, “Enter your **<User Name>**”.
Example > *Example*	Arrows separating the parts of a navigation path, for example, menu options
Example	Emphasized words or expressions
Example	Words or characters that you enter in the system exactly as they appear in the documentation
www.sap.com	Textual cross-references to an internet address
/example	Quicklinks added to the internet address of a homepage to enable quick access to specific content on the Web
123456	Hyperlink to an SAP Note, for example, SAP Note 123456
Example	• Words or characters quoted from the screen. These include field labels, screen titles, pushbutton labels, menu names, and menu options. • Cross-references to other documentation or published works
`Example`	• Output on the screen following a user action, for example, messages • Source code or syntax quoted directly from a program • File and directory names and their paths, names of variables and parameters, and names of installation, upgrade, and database tools
`EXAMPLE`	Technical names of system objects. These include report names, program names, transaction codes, database table names, and key concepts of a programming language when they are surrounded by body text, for example, `SELECT` and `INCLUDE`
EXAMPLE	Keys on the keyboard

www.sap.com