OVO Report Pack

Software Version: 1.0

HP OpenView Performance Insight

User Guide

May 2006

Introduction

The OVO Report Pack includes both a data collection datapipe and the rollup and reporting aspects normally found in a Report Pack. The data collection datapipe is an integrated part of the Report Pack.

The OVO Report Pack collects and reports the following information from OVOu servers:

- Message patterns based on OVOu Active and History message tables.

- Service status based on the OVOu Service Log table.

The version of the OVO Report Pack supports the collection of data from multiple OVOu servers.

The purpose of this document is to provide installation and usage information to knowledgeable OVPI administrators who are responsible for administering OVPI and this Report Pack.

Installation

Required Environment

The OVO Report Pack is designed to operate in the following environments:

- HP OVPI 5.0 SP 3 with Oracle 9.0.2.4 or Sybase 11.9.2.5 as the OVPI database.
- HP OVPI 5.1 SP 1 with Oracle 9.0.2.5 or Sybase 11.9.2.5 as the OVPI database.

The environment described above provides the Oracle connectivity required to connect to the OVOu Oracle database and extract the datasets needed. The following must be enabled to connect to the OVOu Oracle database:

- The OVOu server hostname must resolve to an IP address (ping the hostname).
- The OVOu Oracle database connection information must be added to OVPI via the OVPI Management Console Systems Information application.

When installing this Report Pack and connecting to the OVOu database for the first time, the following information must be provided:

- OVOu database server name (resolvable hostname).
- Port number that the OVOu database is accepting connections.
- Oracle SID for the OVOu database.
- OVOu database user name.
- OVOu database password.

Installation Steps Summary

The installation process involves the following steps:

- Place the OVO_Reporting folder in the {DPIPE_HOME}/Packages directory.
- Use OVPI's Package Manager utility to install the Report Pack.
- Configure OVPI server to connect to OVOu database.
- Installation verification.

NOTE: If an earlier version the OVO_Reporting Report Pack is already installed, you should uninstall that version before installing the new version.

Installing the OVO Report Pack

If you have not already done so, copy the OVO_Reporting directory and subdirectories to the OVPI packages directory ({DPIPE_HOME}) resulting in the following directory structure:

```
{DPIPE_HOME}/Packages/OVO_Reporting/OVO.ap
{DPIPE_HOME}/Packages/OVO_Reporting/OVO_Demo.ap
{DPIPE_HOME}/Packages/OVO_Reporting/OVO_Datapipe.ap
```

In the OVPI Management Console, use the Package Manager utility to step through the process of installing the OVO_Reporting Report Pack.

After installing the package, the installation can be verified by following the “Installation Verification” instructions later in this document.

1. Stop OVPI Timer.

    Windows: Select Settings > Control Panel > Administrative Tools > Services

    UNIX: As root, type one of the following:

        HP-UX: sh /sbin/init.d/ovpi_timer stop
        Sun: sh /etc/init.d/ovpi_timer stop

2. Log in to the system. On UNIX systems, log in as root.

3. Stop OVPI Timer and wait for processes to terminate.

4. Start Package Manager. The Package Manager welcome window opens.

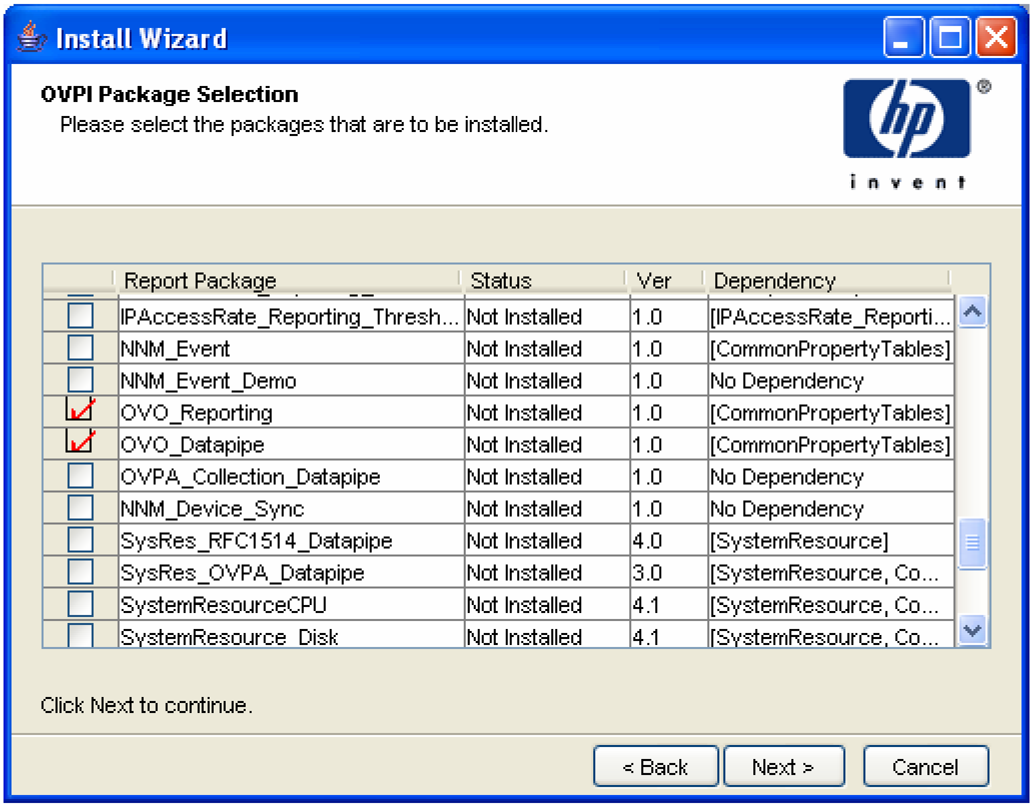
5. Click Next. The Package Location window opens.

6. Click Install. Approve the default installation directory or use the browse feature to select a different directory if necessary.

5. Click Next. The Report Deployment window opens. Accept the default for Deploy Reports; accept the default for application server name and port; type the username and password for the OVPI Application Server.

6. Click Next. The Package Selection window opens. Click the check box next to the following packages:

    OVO_Reporting
    OVO_Datapipe



OVPI Package Selection

Please select the packages that are to be installed.

	Report Package	Status	Ver	Dependency
☐	IPAccessRate_Reporting_Thresh...	Not Installed	1.0	[IPAccessRate_Reporti...
☐	NNM_Event	Not Installed	1.0	[CommonPropertyTables]
☐	NNM_Event_Demo	Not Installed	1.0	No Dependency
☑	OVO_Reporting	Not Installed	1.0	[CommonPropertyTables]
☑	OVO_Datapipe	Not Installed	1.0	[CommonPropertyTables]
☐	OVPA_Collection_Datapipe	Not Installed	1.0	No Dependency
☐	NNM_Device_Sync	Not Installed	1.0	No Dependency
☐	SysRes_RFC1514_Datapipe	Not Installed	4.0	[SystemResource]
☐	SysRes_OVPA_Datapipe	Not Installed	3.0	[SystemResource, Co...
☐	SystemResourceCPU	Not Installed	4.1	[SystemResource, Co...
☐	SystemResource_Disk	Not Installed	4.1	[SystemResource, Co...

Click Next to continue.

7. Click Next. The Type Discovery window opens. Disable the Type Discovery option.

8. Click Next. The Selection Summary window opens.

9. Click Install. The Installation Progress window opens and the install process begins. When the install finishes, a package install complete message appears.

10. Click Done.

11. Restart OVPI Timer.

Windows: Select Settings > Control Panel > Administrative Tools > Services

UNIX: As root, type one of the following:

HP-UX: sh /sbin/ovpi_timer start
Sun: sh /etc/init.d/ovpi_timer start

Configuring OVOu Database Connection

When installing the OVO Report Pack, the OVPI server must have connectivity information for the OVOu database. Without this information, the OVPI server and the ovou_datapipe won't be able to connect to the OVOu database. Configuration is done using the system option from the OVPI Management Console. Using the interface, add the OVOu database as an external database. The ovou_datapipe will use the information provided in the external database configuration, including user name and password, to connect to the OVOu database and export the required information.

The ovou_datapipe uses the following two files to determine which OVOu databases to export data from and how to connect to those databases:

{DPIPE_HOME}/data/ovou8servers.prp (list of OVOu8 servers to collect data from)
{DPIPE_HOME}/data/ovou7servers.prp (list of OVOu7 servers to collect data from)

{DPIPE_HOME}/data/systems.xml (connectivity information for databases OVPI can connect to)

The ovou[7|8]servers.prp is part of the OVO Report Pack and is put in place as part of the OVO Report Pack installation process.

When the ovou_datapipe is initiated and only the "-dataset" option is specified, the ovou_datapipe will collect data from all OVOu servers listed in the ovou[7|8]servers.prp. After looking up the list of servers to collect from in the ovou[7|8]servers.prp file, the ovou_datapipe looks up connectivity information for each listed OVOu server in OVPI's systems.xml file.

The following requirements must be met on the OVPI server for the ovou_datapipe to operate correctly:

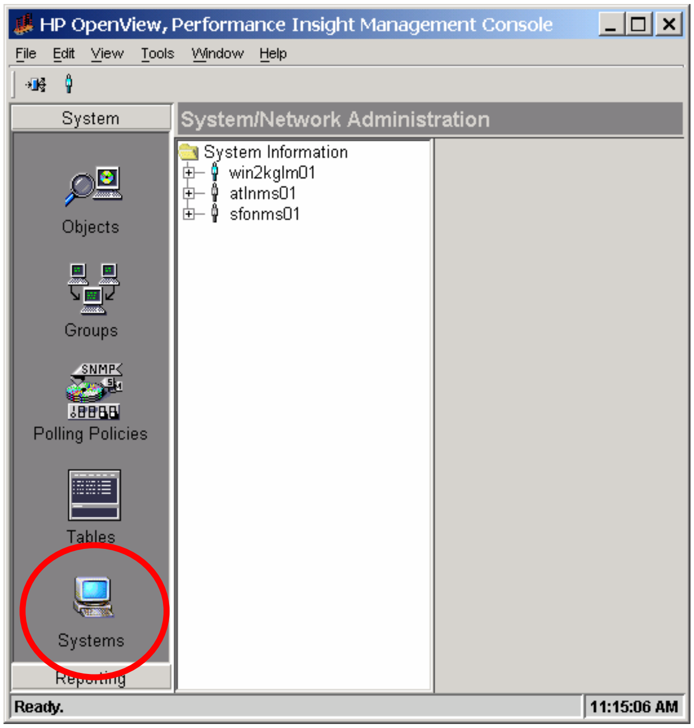
- The OVOU database server hostname must be pingable from the OVPI server.
- OVPI 5.x and appropriate service packs must be installed.
- When adding the OVOU database to OVPI as an "External Database" the following information is required:

  OVOu database server name (resolvable hostname).
  Port number that the OVOu database is accepting connections.
  Oracle SID for the OVOu database.
  OVOu database user name.
  OVOu database user's password.

Adding OVOu External Database to OVPI

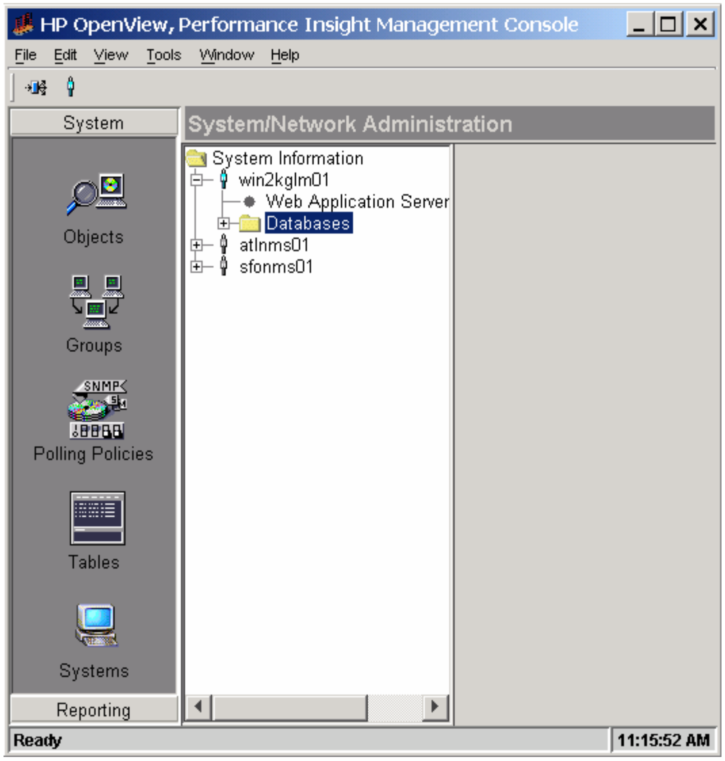
Start the OVPI Management Console and login as the OVPI administrator, trendadm.

Select the System Application in the Navigation Pane.

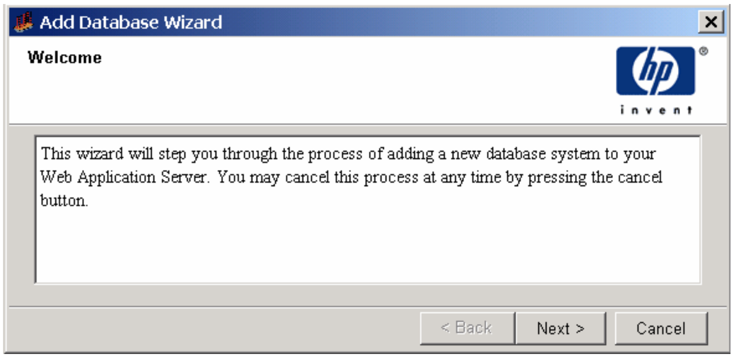


In the center pane under the "System Information" folder, open the container representing the local OVPI server.

Right-click over the "Database" folder within that container and select the "Add Database" option.



The Add Database Wizard welcome screen opens. Click Next.

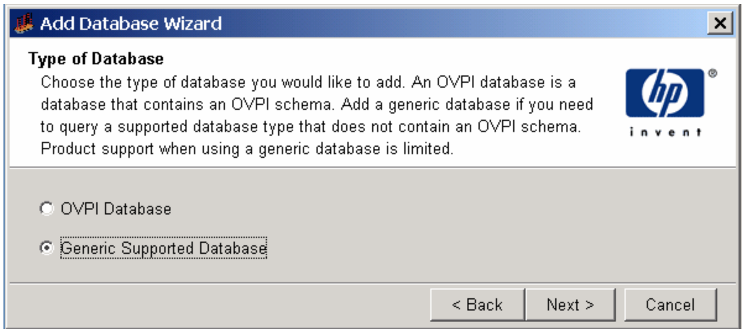


Add Database Wizard

Welcome

This wizard will step you through the process of adding a new database system to your Web Application Server. You may cancel this process at any time by pressing the cancel button.

In the Type of Database window, select the “Generic Supported Database” option. Click Next.



Add Database Wizard

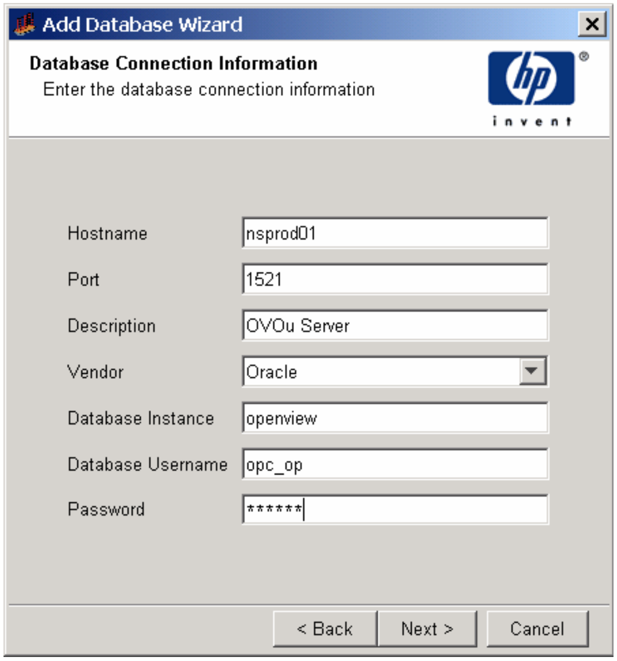
Type of Database

Choose the type of database you would like to add. An OVPI database is a database that contains an OVPI schema. Add a generic database if you need to query a supported database type that does not contain an OVPI schema. Product support when using a generic database is limited.

○ OVPI Database

● Generic Supported Database

In the Database Connectivity Information window, enter the required information for the OVOu database. Click Next.



Add Database Wizard

Database Connection Information

Enter the database connection information

Hostname	nsprod01
Port	1521
Description	OVOu Server
Vendor	Oracle
Database Instance	openview
Database Username	opc_op
Password	******

Hostname: IP Hostname of the OVOu server.

Port: The port the database is listening on.

Description: user specified description.

Vendor: Oracle.

Database Instance: SID of OVOu database.

Database Username: OVOu database login.

Password: OVOu database password for login.

Click Next.

In the Identify System window, enter a name and a description for this external database connection. The name specified here is a label that will be used for several purposes within the OVO Report Pack. The name specified here is only a label. The hostname of the server is not required here. The hostname can be entered in this field if desired but another label or name could be entered that is meaningful to the IT staff.

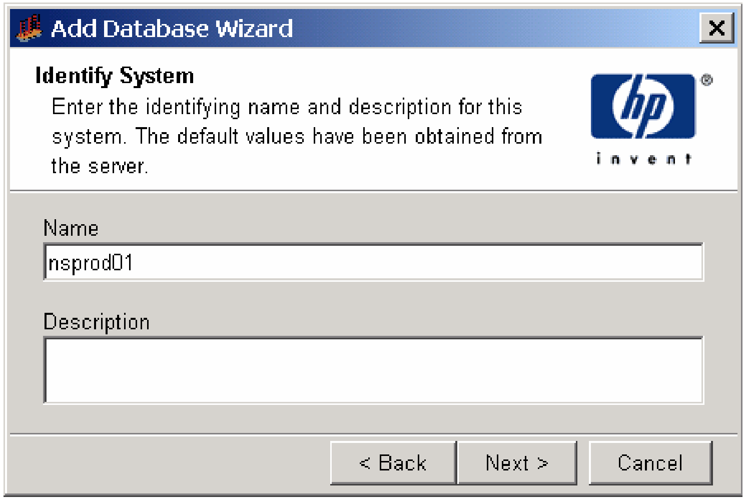
The name specified here is used for the following within the OVO Report Pack:

- Used by the OVOu Datapipe to determine connectivity information from systems.xml.
- Appears in the OVOu reports provided by the OVO Report Pack.

As mentioned, this name is referenced by the OVOu Datapipe to locate connectivity information for the OVOu database. When the OVOu Datapipe is initiated, it accesses the ovou[7|8]servers.prp file to determine the name of one or more OVOu servers to export data from. Each entry in the ovou[7|8]servers.prp file must match the name of an OVOu server as specified in the name field of the Identify System window. If an entry in the ovou[7|8]servers.prp file does not map to a valid name specified in this screen, data collection will fail for that OVOu server.

The name specified in this screen becomes part of the resulting entry in OVPI's systems.xml file and is contained in the <Name> field of the new entry.

Enter the correct information and click Next.



Add Database Wizard

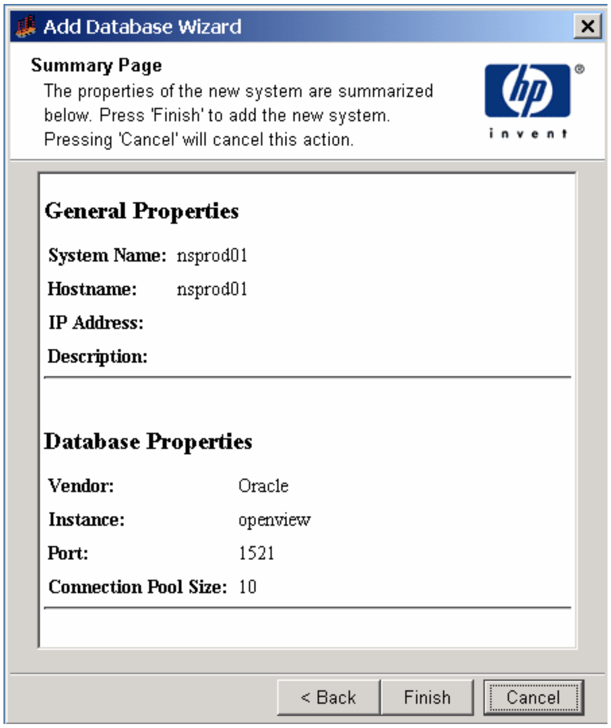
Identify System

Enter the identifying name and description for this system. The default values have been obtained from the server.

Name: nsprod01

Description:

Verify the parameters in the Summary Page window. Click Finish to complete the process.



Summary Page

The properties of the new system are summarized below. Press 'Finish' to add the new system. Pressing 'Cancel' will cancel this action.

General Properties

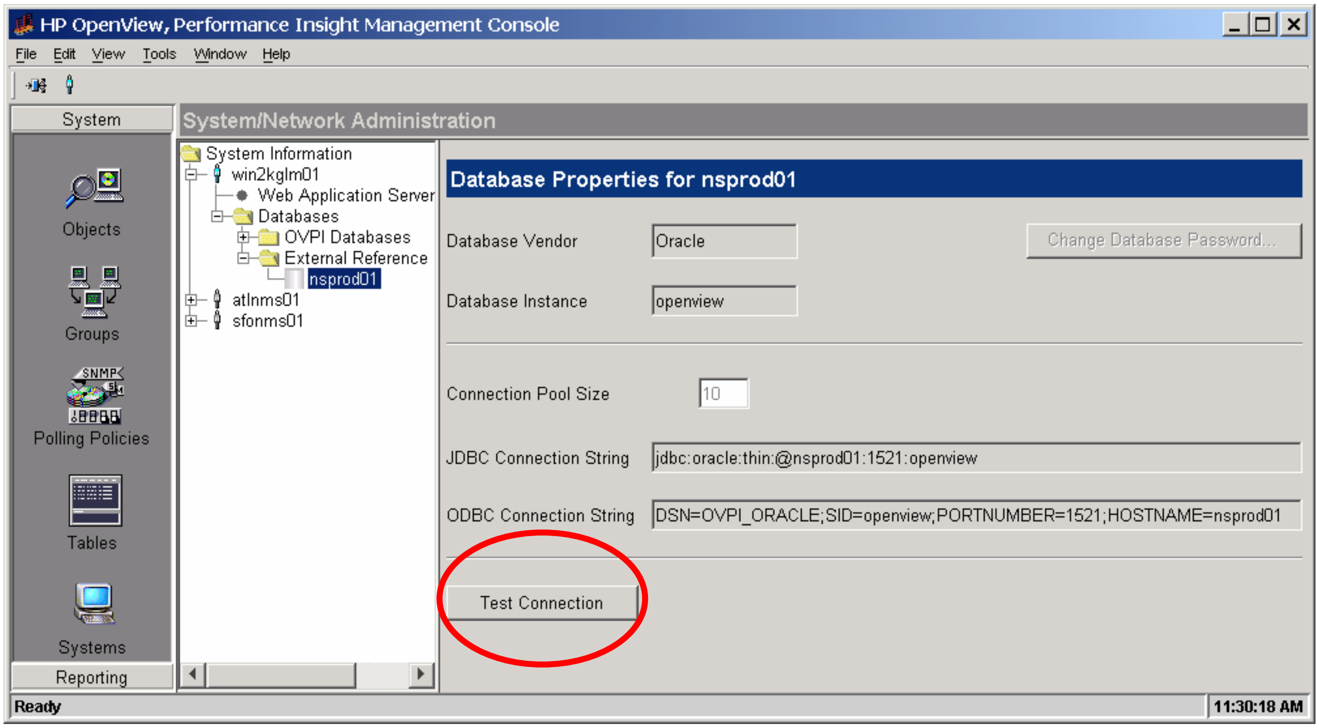
System Name:	nsprod01
Hostname:	nsprod01
IP Address:	
Description:	

Database Properties

Vendor:	Oracle
Instance:	openview
Port:	1521
Connection Pool Size:	10

Verifying External Database Connection to OVOu

Verify the connection just added by selecting this new database in the Systems Application. View the connectivity parameters.



Database Properties for nsprod01

Database Vendor: Oracle

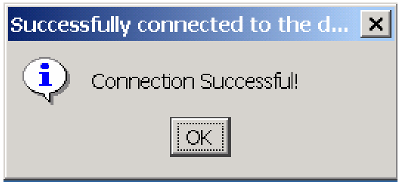
Database Instance: openview

Connection Pool Size: 10

JDBC Connection String: jdbc:oracle:thin:@nsprod01:1521:openview

ODBC Connection String: DSN=OVPI_ORACLE;SID=openview;PORTNUMBER=1521;HOSTNAME=nsprod01

Click the “Test Connection” button to verify that OVPI can successfully connect to the OVOu database.



Connection Successful!

Registering OVOu Servers with OVOu Datapipe

After adding the external database connection to OVPI using the Management Console, a file must be created containing a list of OVOu servers that the OVOu Datapipe should collect data from.

Use an editor to edit the following plain text file:

```
{DPIPE_HOME}/data/ovou[7|8]servers.prp
```

The file should contain a list of OVOu servers that data should be collected from. Each OVOu server should be listed on a separate line. If data is to be collected from a single OVOu server, the file will contain only one line. No other information is allowed in this file. Do not insert comments or any other information in this file. **NOTE:** The name specified in this file for each OVOu server must exactly match the name specified in the “Identify System” screen when the External Database Connection was added to the OVPI Management

Console's System application. For more information, refer to the section earlier in this document that covers that topic.

The name specified here is used for the following within the OVO Report Pack:

- Used by the OVOu Datapipe to determine connectivity information from systems.xml.
- Appears in the OVOu reports provided by the OVO Report Pack.

Configuring OVOu for Active Message Export

Instead of exporting the actual messages of the Active Message table, the OVOu Datapipe takes a snapshot of Active Message table current statistics based on the messages currently in the table. The results are summarized to hourly, daily, weekly, and monthly statistics similar to the HISTORY statistics at the same level but presenting a summary of "Current Active Messages" as compared to "New HISTORY".

To capture current Active Message statistics, three separate views are created within the OVOu database to enable the export of snapshots of current message statistics. Other levels of data are generated with the OVO Report Pack rollup from the data provided by these three levels.

The following export views are created for the element level indicated:

OVPI_ACT_ONA:	OVO_SERVER – NODE_NAME – APPLICATION
OVPI_ACT_ONS:	OVO_SERVER – NODE_NAME – SERVICE_NAME
OVPI_ACT_ONM:	OVO_SERVER – NODE_NAME – MSG_GROUP

These views must be created within the OVOu database before Active Message statistics can be exported from that OVOu server. The OVOu Datapipe should be used to create these views as follows:

NOTE: the OVO Report Pack must be installed before taking this step.

Login to the OVPI server as the OVPI administrator (trendadm on Unix)

Execute the following command at the command line:

```
ovou_datapipe -dataset create_view –database XYZ –server u8
```

For each OVOu database listed in the ovou[7|8]servers.prp file, four blocks of messages will be printed on the command line indicating that SQL statements are being executed to create four table views in each OVOu database.

Operation of the OVOu Export Process

The OVOu export process is very flexible and has several configuration parameters. The exporter connects to the OVOu database and exports information from the required tables. After the export process runs the first time, only new records are exported from the OVO Consolidated and Service Log tables.

The OVOu export process is driven by an OVPI trend_proc batch file. After installation, the file will be located in the following location:

```
{DPIPE_HOME}/scripts/OVO_Datapipe_Hourly.pro
```

This file drives three separate export processes, one for each of the three required OVOu datasets. The actual executable commands for the datasets are:

```
ovou_datapipe -dataset MSG
ovou_datapipe -dataset SRV_LOG
ovou_datapipe -dataset ACT
```

As shown above, the actual export program is called "ovou_datapipe". The program has several configurable parameters. However, in most cases only the "-dataset" parameter is needed. The information that ovou_datapipe requires to successfully connect to each OVOu database should have been added to the Report Pack's configuration when it was installed and is contained in the following files:

{DPIPE_HOME}/data/ovou8servers.prp	(list of OVOu8 servers to collect data from)
{DPIPE_HOME}/data/ovou7servers.prp	(list of OVOu7 servers to collect data from)
{DPIPE_HOME}/data/systems.xml	(list of servers and databases OVPI can connect to)

When the ovou_datapipe is initiated and only the "-dataset" option is specified, the ovou_datapipe will collect data from all OVOu servers listed in the ovou[7|8]servers.prp file. After looking up the list of servers to collect from in the ovou[7|8]servers.prp file, the ovou_datapipe looks up connectivity information for each in OVPI's systems.xml file.

Parameters for ovou_datapipe:

Parameter	Description
-dataset	<ACT|MSG|CFG|SRV_LOG|create_view|drop_view> ACT: export OVOu Active Message Statistics. HIST: export OVO Consolidated Messages. CFG: export OVOu channel configuration. SRV_LOG: export OVOu Service Log records. create_view: create OVOu Active Message Statistics database Views. drop_view: remove OVOu Active Message Statistics database Views.
-database	"Name" of OVOu server to connect to and export OVOu data from. The specified name must match an entry in OVPI's systems.xml file based on the "<Name>" tag field in the systems.xml file. This must also match an entry in the ovou[7|8]servers.prp file. Normally no OVOu server will be specified and data will be collected from all OVOu servers listed in the ovou[7|8]servers.prp file.
-datadir	By default, data is exported to $DPIPE_HOME/data/ImportData/OVO_Reporting. A different location can be specified using this option.
-prevdays	Specifies the number of days prior to today to collect data inclusive of today. Only used if the previous collection time is not stored in the maxtime.<TIME_TABLE>_<TIME_FIELD> file or for the first execution. The default value is 1 day. This parameter only applies to History and Service log data.
-debug	Set for Debug output.
-help	Display help information.
-version	Display ovou_datapipe version.
-V	Display ovou_datapipe version.
-server	Either u8 or u7 cooresponding to the database specified in -database

The first time the export process is run, an excessive amount of data could be exported from the OVOu database if the process is allowed to go back too far in time to locate records to export. The OVOu datapipe can be configured to go back a certain number of days to locate records for export. If no value for this option is specified, the default forces the export of messages only as old as midnight of the previous day.

The "-prevdays" option does not apply to all datasets. For example, the OVOu Active Message table export process does not actually export message records. Instead, a small number of records are captured that provide a snap shot of "current" record statistics for the OVOu Active Message table. In this case there is no need to specify a maximum number of days to include in the export process.

Note: OVOu export files are imported into OVPI and then moved to an archive directory by this module. The OVPI administrator should setup a process to clean out the OVOu export files from the following location on the OVPI server to avoid filling up the filesystem:

{DPIPE_HOME}/data/ImportData/OVO_Reporting/Archive

Installation Verification

The steps listed below are provided to assist the OVPI Administrator in verifying that the Report Pack is operating correctly. These steps are not required and can be skipped unless there is reason to believe that the installation may not have successfully completed.

Verifying Collection and Rollup Scheduling

The installation process adds entries to trendtimer.sched to ensure that processes are executed as needed by trendtimer. The following entries are added to trendtimer.sched:

```
# HP OVO Reporting Project
1:00+5 - - {DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/OVO_Datapipe_Hourly.pro
24:00+2:00 - - {DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/OVO_Datapipe_Daily.pro

### OVO Reporting Daily Processing
24:00+8:00 - - {DPIPE_HOME}/bin/trend_proc -f {DPIPE_HOME}/scripts/Daily_OVO_Rollup.pro
```

Verifying Deployed Reports

The installation process deploys reports to the specified OVPI server if that Package Manager option was selected during the installation process. Reports are provided in the following directory:

{DPIPE_HOME}/reports/deploy/system/OVO_Reporting

Verifying OVOu Data Export

After successfully completing the installation instructions above, data collection should be automatically performed for the first time at 5 minutes past the hour, based on the entries that were added by the installation process to trendtimer.sched for data collection (1:00+5 - -). This process exports the records from the OVOu database into flat files on the OVPI server. The OVOu data is later imported into the OVPI database and summarized by the daily rollup process specific to each dataset.

Note: The first time the ovou_datapipe is run, data records received since midnight of the previous day are exported by default. This behavior can be overridden if desired. To force the export process to include records from further back in time, specific “-prevday” option and specify the appropriate number of days. This only affects the first time the export process is executed for a specific dataset. After that, the ovou_datapipe keeps track of the last exported timestamps and only exports new records that have arrived since that time.

Note: If this option is desired, the “-prevday” option must be added to the ovou_datapipe statements before the datapipe is executed the first time.

The following command can be executed at the command line to initiate the ovou_datapipe export process manually:

```
trend_proc -f {DPIPE_HOME}/scripts/OVO_Datapipe_Hourly.pro
```

After the successful completion of the above command, the exported data should be located in the following directory:

```
{DPIPE_HOME}/data/ImportData/OVO_Reporting
```

The exported OVOu data in that directory should be in files with names similar to the following:

```
OVO_ACT_ONA_<ovou server name>_<date>_<time>.dat
OVO_ACT_ONM_<ovou server name>_<date>_<time>.dat
OVO_ACT_ONS_<ovou server name>_<date>_<time>.dat
OVO_MSG_HIST_<ovou server name>_<date>_<time>.dat
OVO_MSG_ACT<ovou server name>_<date>_<time>.dat
OVO_SRV_LOG_<ovou server name>_<date>_<time>.dat
OVO_CFG_LOG_<ovou server name>_<date>_<time>.dat
```

Verifying Data Import into OVPI

The OVOu data is imported into the OVPI database by the daily rollup process specific to each dataset. Each daily rollup process imports the appropriate dataset files and then steps through the rollup of that dataset.

Note: The data is imported into OVPI once per day as part of the daily rollup process.

The import process should cause the following to occur:

The following data tables in the OVPI database should be populated with records.

R_OVO_MSG_BASE
R_OVO_ACT_BASE
R_OVO_SRVLOG_BASE
R_OVO_CONFIG_BASE

* Records will only be added to these tables if the OVOu server has received new records for those datasets.

Verify this in Table Manager by navigating to the OVO_Reporting Category and locating the tables listed above.

The oldest and most recent time stamps can be determined for the import tables by executing the following SQL statement in SQLplus, ISQL, or SQL Advantage:

```
select ‘R_OVO_MSG_BASE’, min(ta_period), max(ta_period) from R_OVO_MSG_BASE
select ‘R_OVO_ACT_BASE’, min(ta_period), max(ta_period) from R_OVO_ACT_BASE
select ‘R_OVO_SRVLOG_BASE’, min(ta_period), max(ta_period) from R_OVO_SRVLOG_BASE
select ‘R_OVO_CONFIG_BASE’, min(ta_period), max(ta_period) from R_OVO_CONFIG_BASE
```

This should return valid date/times in the second and third column if data has been imported.

The OVOu export files should have been archived to the following location by the OVPI data import process:

```
{DPIPE_HOME}/data/ImportData/OVO_Reporting/Archive
```

If the OVOU export files are in the "OVO_Reporting" directory but not in the "Archive" directory, the OVOU export ran successfully but the import process was not successful. Errors relating to the import process can be found in the {DPIPE_HOME}/log/trend.log file.

Verifying Hourly Data Rollup

After several hours of data have been imported into the OVO Report Pack's Base tables within the OVPI database, the hourly rollup process can be initiated as follows:

Change directory to {DPIPE_HOME}/scripts

Execute the following commands:

```
trend_proc -f Daily_OVO_ACT_Rollup.pro
trend_proc -f Daily_OVO_MSG_Rollup.pro
trend_proc -f Daily_OVO_SRVLOG_Rollup.pro
trend_proc -f Daily_OVO_CFG_Rollup.pro
```

Use Table Manager to verify that the hourly data tables have been populated or execute the following SQL statements:

```
select 'SH_OVO_MSG', min(ta_period), max(ta_period) from SH_OVO_MSG
select 'SH_OVO_ACT', min(ta_period), max(ta_period) from SH_OVO_ACT
```

These statements should return valid date/times in the second and third column if data has been rolled up to the hourly level.

Note: The individual steps of any trend_proc file can be executed at the command line as long as the environment variables are defined correctly and slashes are in the correct direction (/).

Verifying Daily Data Rollup

After a complete day of data has been imported into the OVO Report Pack's Base tables within the OVPI database, the daily rollup process can be initiated again using the same command as specified above:

Change directory to {DPIPE_HOME}/scripts

Execute the following commands:

```
trend_proc -f Daily_OVO_ACT_Rollup.pro
trend_proc -f Daily_OVO_MSG_Rollup.pro
trend_proc -f Daily_OVO_SRVLOG_Rollup.pro
trend_proc -f Daily_OVO_CFG_Rollup.pro
```

Use Table Manager to verify that the hourly data tables has been populated or execute the following SQL statements:

```
select ‘SD_OVO_MSG’, min(ta_period), max(ta_period) from SD_OVO_MSG
select ‘SD_OVO_ACT’, min(ta_period), max(ta_period) from SD_OVO_ACT
select ‘SD_OVO_SRVLOG’, min(ta_period), max(ta_period) from SD_OVO_SRVLOG
```

These statements should return valid date/times in the second and third column if data has been rolled up to the hourly level.

The daily rollup processes also force the summarization to the weekly, monthly and baseline level, if enough time has passed and enough data has been collected to summarize to those levels. Data will only rollup to the next time interval for time periods that are complete. April’s monthly data can not be generated until the conclusion of the month of April.

Verify those tables using the following SQL statements:

```
select ‘SW_OVO_MSG’, min(ta_period), max(ta_period) from SW_OVO_MSG
select ‘SM_OVO_MSG’, min(ta_period), max(ta_period) from SM_OVO_MSG

select ‘SW_OVO_ACT’, min(ta_period), max(ta_period) from SW_OVO_ACT
select ‘SM_OVO_ACT’, min(ta_period), max(ta_period) from SM_OVO_ACT

select ‘SW_OVO_SRVLOG’, min(ta_period), max(ta_period) from SW_OVO_SRVLOG
select ‘SM_OVO_SRVLOG’, min(ta_period), max(ta_period) from SM_OVO_SRVLOG
```

Data Tables & Views

This solution uses a large number of data tables and data table views. This section provides information on the purpose and usage of each.

Setting Table Retentions

After the installation and operation have been verified, it is recommended that the retention period for each level of data be examined and modified as needed. Each table is configured to retain data for the following periods by default:

R_OVO_MSG_BASE:	7 days
SH_OVO_MSG:	60 days
SD_OVO_MSG:	700 days
SW_OVO_MSG:	700 days
SM_OVO_MSG:	700 days
R_OVO_ACT_BASE:	7 days
SH_OVO_ACT:	60 days
SD_OVO_ACT:	700 days
SW_OVO_ACT:	700 days
SM_OVO_ACT:	700 days
R_OVO_SRVLOG_BASE:	7 days
SH_OVO_SRVLOG:	60 days
SD_OVO_SRVLOG:	700 days
SW_OVO_SRVLOG:	700 days
SM_OVO_SRVLOG:	700 days
R_OVO_CONFIG_BASE:	30 days

Setting data retention levels too high could result in performance degradation as the database grows larger. If the database reaches 90% of capacity, OVPI processing will stop. Be very careful when setting table retentions levels.

Import Tables

OVOu records are imported into the following tables:

History Messages:	R_OVO_MSG_BASE, K_OVO_MSG_BASE
Active Message Statistics:	R_OVO_ACT_BASE, K_OVO_ACT_BASE
Service Log Entries:	R_OVO_SRVLOG_BASE, K_OVO_SRVLOG_BASE
Configuration Entries:	R_OVO_CONFIG_BASE,K_OVO_CONFIG_BASE

Import Tables – History Message

OVO Consolidated Message OVPI Import Tables: R_OVO_MSG_BASE, K_OVO_MSG_BASE

OVPI Field	Data Type	OVO Table(s)	OVO Table(s)	OVO Field	Example Data
		HISTORY	ACTIVE		
OVO_SERVER	Vchar(16)				
ACKN_AFTER_TT_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	ACKN_AFTER_TT_FLAG	
ACKN_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	ACKN_TIME	
ACKN_USER	Vchar(20)	OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	ACKN_USER	
APPLICATION	Vchar(254)	OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	APPLICATION	
AUTO_ACKN_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	AUTO_ACKN_FLAG	
AUTO_ANNO_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	AUTO_ANNO_FLAG	
AUTO_CALL		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	AUTO_CALL	
AUTO_NODE_ID		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	AUTO_NODE_ID	
AUTO_STATUS		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	AUTO_STATUS	
		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	CMA_FLAG	
DESCRIPTION	Vchar(254)	OPC_COND, OPC_TRAP_COND, OPC_MONITOR_COND	OPC_COND, OPC_TRAP_COND, OPC_MONITOR_COND	DESCRIPTION	
CONDITION_ID	Vchar(36)	OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	CONDITION_ID	
DUPL_COUNT		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	DUPL_COUNT	
ESCALATE_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	ESCALATE_FLAG	
FORWARD		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	FORWARD	
IP_ADDRESS		OPC_NODE_NAMES	OPC_NODE_NAMES	IP_ADDRESS	
LAST_TIME_RECEIVED		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LAST_TIME_RECEIVED	
LOCAL_RECEIVING_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_RECEIVING_TIME	
LOCAL_ACKN_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_ACKN_TIME	
LOCAL_AGT_CREATION_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_AGT_CREATION_TIME	
LOCAL_CREATION_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_CREATION_TIME	
LOCAL_LAST_TIME_RECEIVED		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_LAST_TIME_RECEIVED	
LOCAL_RECEIVING_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_RECEIVING_TIME	
LOCAL_UNBUFFER_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOCAL_UNBUFFER_TIME	
LOG_ONLY_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	LOG_ONLY_FLAG	
MESSAGE_GROUP		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MESSAGE_GROUP	
MESSAGE_NUMBER		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MESSAGE_NUMBER	
MESSAGE_TYPE		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MESSAGE_TYPE	
MSG_GEN_NODE_ID		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MSG_GEN_NODE_ID	
MSG_KEY		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MSG_KEY	

MSG_SOURCE_NAME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MSG_SOURCE_NAME
MSG_SOURCE_TYPE		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	MSG_SOURCE_TYPE
NODE_ID		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	NODE_ID
NODE_NAME	Vchar(64)	OPC_NODE_NAMES	OPC_NODE_NAMES	NODE_NAME
NOTIFICATION_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	NOTIFICATION_FLAG
NOTIFY_SERVICES		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	NOTIFY_SERVICES
OBJECT		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OBJECT
OP_INIT_ACKN_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OP_INIT_ACKN_FLAG
OP_INIT_ANNO_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OP_INIT_ANNO_FLAG
OP_INIT_CALL		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OP_INIT_CALL
OP_INIT_NODE_ID		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OP_INIT_NODE_ID
OP_INIT_STATUS		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	OP_INIT_STATUS
RECEIVING_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	RECEIVING_TIME
SERVICE_NAME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	SERVICE_NAME
SEVERITY		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	SEVERITY
TROUBLE_TICK_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	TROUBLE_TICK_FLAG
UNBUFFER_TIME		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	UNBUFFER_TIME
UNMATCHED_FLAG		OPC_MSG_MESSAGES	OPC_ACT_MESSAGES	UNMATCHED_FLAG
cma_name		opc_MSG_cust_attrib	opc_act_cust_attrib	cma_name
cma_value		opc_MSG_cust_attrib	opc_act_cust_attrib	cma_value
service_id		opc_notif_schedule	opc_notif_schedule	service_id
status_flag		opc_trouble_ticket	opc_trouble_ticket	status_flag
TEMPLATE_ID	Vchar(36)	opc_templ_options	opc_templ_options	TEMPLATE_ID
TEMPLATE_NAME		opc_templ_options	opc_templ_options	TEMPLATE_NAME
TEMPLATE_DESCRIPTION	Vchar(254)	opc_templ_options	opc_templ_options	TEMPLATE_DESCRIPTION
APPLICATION_ID	Vchar(254)	Opc_msg_Messages	Opc_act_messages	APPLICATION_ID

Import Tables – Active Message

OVOu Active Message OVPI Import Tables: R_OVO_ACT_BASE, K_OVO_ACT_BASE

COLUMN NAME	**Data Type**	**Description**	**SOURCE TABLE**	
OVO_SERVER	varchar(128)	OV server records were exported from		Created by export process

NODE_NAME
(based on NODE_ID)
	varchar(254)
	varchar2(1024)
			OPC_NODE_NAMES	All records
				Is it possible for this value to be null in the OVO DB? Should we check for this

				and sub-in IP ADDRESS?
IP_ADDRESS (based on NODE_ID)	varchar(32)		OPC_NODE_NAMES	All records
NODE_ID ID of node where event occurred	varchar2(36)		OPC_ACT_MESSAGES	All records
APPLICATION	varchar2(254)	ONA records only	OPC_ACT_MESSAGES	ONA records only
SERVICE_NAME	**varchar2(2048)**	ONS records only	OPC_ACT_MESSAGES	ONS records only
MSG_GROUP	varchar2(32)	ONM records only	OPC_ACT_MESSAGES	ONM records only
EXPORT_TIME	date	Becomes ta_period Ta_period format 1		Created by export process
OVO_GMT_OFFSET	number(12)			All records
CNT_MESSAGES	gauge/float	Count of current ACTIVE messages	OPC_ACT_MESSAGES	All records
CNT_UNKNOWN	gauge/float	With severity = 1 = unknown	OPC_ACT_MESSAGES	All records
CNT_NORMAL	gauge/float	With severity = 2 = normal	OPC_ACT_MESSAGES	All records
CNT_WARNING	gauge/float	With severity = 4 = warning	OPC_ACT_MESSAGES	All records
CNT_CRITICAL	gauge/float	With severity = 8 = critical	OPC_ACT_MESSAGES	All records
CNT_MINOR	gauge/float	With severity = 16 = minor	OPC_ACT_MESSAGES	All records
CNT_MAJOR	gauge/float	With severity = 32 = major	OPC_ACT_MESSAGES	All records
MIN_AGE_MSG	gauge/float	Minimum age of current ACTIVE messages	OPC_ACT_MESSAGES	All records
MAX_AGE_MSG	gauge/float	Maximum age of current ACTIVE messages	OPC_ACT_MESSAGES	All records
AVG_AGE_MSG	gauge/float	Average age of current ACTIVE messages	OPC_ACT_MESSAGES	All records
TOT_AGE_MSG	gauge/float	Total age of current ACTIVE messages	OPC_ACT_MESSAGES	All records
CNT_AGE_5_MIN	gauge/float	Messages less than 5 minutes old	OPC_ACT_MESSAGES	All records
CNT_AGE_5_10_MIN	gauge/float	>= 5 minutes and < 10 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_10_30_MIN	gauge/float	>= 10 minutes and < 30 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_30_60_MIN	gauge/float	>= 30 minutes and < 60 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_60_120_MIN	gauge/float	>= 60 minutes and < 120 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_120_720_MIN	gauge/float	>= 120 minutes and < 720 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_720_1440_MIN	gauge/float	>= 720 minutes and < 1440 minutes	OPC_ACT_MESSAGES	All records
CNT_AGE_GT_1440_MIN	gauge/float	> 1440 minutes	OPC_ACT_MESSAGES	All records
CNT_CURR_BUFFERED	gauge/float	Count current buffered messages	OPC_ACT_MESSAGES	All records
CNT_PREV_BUFFERED	gauge/float	Count messages previously buffered	OPC_ACT_MESSAGES	All records
CNT_DUPL_COUNT	gauge/float	Total duplicates received	OPC_ACT_MESSAGES	All records
CNT_DUPL_MSG	gauge/float	Count of messages receiving duplicates	OPC_ACT_MESSAGES	All records
CNT_ESCALATE_FLAG	gauge/float	Count flagged as escalated	OPC_ACT_MESSAGES	All records
CNT_AUTO_ACKN_FLAG	gauge/float	Count flagged as auto acknowledge	OPC_ACT_MESSAGES	All records
CNT_LOG_ONLY_FLAG	gauge/float	Count flagged as log only	OPC_ACT_MESSAGES	All records
CNT_NOTIFICATION_FLAG	gauge/float	Count flagged to send notification	OPC_ACT_MESSAGES	All records
CNT_TROUBLE_TICK_FLAG	gauge/float	Count flagged to generate trouble ticket	OPC_ACT_MESSAGES	All records
CNT_ACKN_AFTER_TT_FLAG	gauge/float	Count flagged to auto acknowledge after	OPC_ACT_MESSAGES	All records

		generating trouble ticket		
APPLICATION_ID	Number	Unique ID for long application names	NA	ONA
NODE_IP_ADDR	Number	IP Address in long form	OPC_NODES	All records
SERVICE_NAME_ID	Number	Unique ID for long service names	NA	ONS

Import Tables – Service Log

OVOu Service Log OVPI Import Tables: R_OVO_SRVLOG_BASE, K_OVO_SRVLOG_BASE

COLUMN NAME	Data Type	COLUMN SAMPLE DATA	OVO SOURCE TABLE	Comment
OVO_SERVER	varchar(128)			Created by export process
SERVICE_NAME	**varchar2(2048)**	SeM_SPI:Server@@nsprod01	OPC_SERVICE_LOG	
SEVERITY	number(3)	4	OPC_SERVICE_LOG	1…Unknown, 2…Normal, 4…Warning, 8…Critical 16…Minor, 32…Major
START_TIME	number(12)	1108624087	OPC_SERVICE_LOG	
LOCAL_START_TIME	date	4/5/2005 10:03:43 PM	OPC_SERVICE_LOG	**TA_PERIOD**
END_TIME	number(12)	1108624087	OPC_SERVICE_LOG	
LOCAL_END_TIME	date	4/5/2005 11:03:43 PM	OPC_SERVICE_LOG	
DURATION	number(12)	1108624087		Created by export process
OVO_GMT_OFFSET	number(12)			Created by export process
EXPORT_TIME	number(12)	1108594373		Created by export process

Import Tables – Config

OVOu Service Log OVPI Import Tables: R_OVO_SRVLOG_BASE, K_OVO_SRVLOG_BASE

Column Name	OVOU Source Table	Comment
dsi_key_id_		Standard OVPI numeric element ID
ta_period		Standard OVPI time stamp
delta_time		Standard OVPI column not used in this case
ta_samples		Standard OVPI column. Always 1 in this case
OVO_SERVER		Created by export process
Type		Record type indicator TT or NOTIF
NOTIF_SVC_ID	Opc_notif_services	Service Identifier
NOTIF_SVC_NAME	Opc_notif_services	Notif Service Name

FLAG	opc_trouble_ticket	Active/Inactive
PROGRAM	Opc_notif_services opc_trouble_ticket	External program
DAY	opc_notif_schedule	Day for Notif
STARTING	opc_notif_schedule	Start time for Notif channel
ENDING	opc_notif_schedule	End time for Notif channel

Reporting Tables

Reporting Tables – History Message Property Table

All OVO Consolidated Message summarized data tables are associated with the K_OVO_MSG property table.

K_OVO_MSG

Column Name	Data type	Population Method
dsi_key_id		Maintained by OVPI
dsi_target_name		Record type indicator based on trend_sum by variables: O: OVO_SERVER ON: OVO_SERVER - NODE_NAME OA: OVO_SERVER - APPLICATION OS: OVO_SERVER - SERVICE_NAME OM: OVO_SERVER - MSG_GROUP OU: OVO_SERVER – ACKN_USER OT: OVO_SERVER – TEMPLATE OTD: OVO_SERVER – TEMPLATE - DESCRIPTION ONA: OVO_SERVER – NODE_NAME - APPLICATION ONS: OVO_SERVER – NODE_NAME - SERVICE_NAME ONM: OVO_SERVER – NODE_NAME - MSG_GROUP All Msgs executive summary across all messages A: APPLICATION S: SERVICE_NAME M: MSG_GROUP U: ACKN_USER T: TEMPLATE TD: TEMPALTE – DESCRIPTION N: NODE NT: NODE – TEMPLATE NS: NODE – SERVICE NA: NODE – APPLICATION NM: NODE – MSG GROUP
dsi_table_key		Due to custom rollup, all by variables, except for GRP_LEVEL and the interval by variable, are concatenated together and inserted into dsi_table_key
dsi_descr		Set to same value as dsi_table_key
GRP_LEVEL		Same as dsi_target_name
OVO_SERVER	Varchar(128)	trend_sum populates based on by variable "OVO_SERVER"

NODE_NAME	Varchar(254)
	Varchar2(1024)
		trend_sum populates based on by variable "NODE_NAME"

APPLICATION	Varchar2(254)	trend_sum populates based on by variable "APPLICATION"
SERVICE_NAME	Varchar2(2048)	trend_sum populates based on by variable "SERVICE_NAME"

ACKN_USER	Varchar2(20)	trend_sum populates based on by variable “ACKN_USER”
MESSAGE_GROUP	VARCHAR(255)	trend_sum populates based on by variable “MSG_GROUP”
MSG_SOURCE_NAME	VARCHAR(255)	trend_sum populates based on by variable “TEMPLATE”
APPLICATION_ID	number	Unique has for application
COND_DESCRIPTION_ID	number	Unique hash for condition description
SERVICE_NAME_ID	number	
CONDITION_DESCR	VARCHAR(255)	trend_sum populates based on by variable “CONDITION”
OTHER standard columns found in OVPI RPs		

Reporting Tables – History Message Data Tables

The following OVO Consolidated Message rollup tables are used for reporting:

Destination Table	**Rollup Processing Populating Table**
SH_OVO_MSG	populated by several rollup processes. SH_OVO_MSG_???.sum
SD_OVO_MSG	populated by trend_sum using SD_OVO_MSG.sum
SW_OVO_MSG	populated by trend_sum using SW_OVO_MSG.sum
SM_OVO_MSG	populated by trend_sum using SM_OVO_MSG.sum

OVO Consolidated Message metrics provided by time interval:

Metric	Hour	Day	Week	Month
CNT_MESSAGES	X	X	X	X
CNT_UNKNOWN	X	X	X	X
CNT_NORMAL	X	X	X	X
CNT_WARNING	X	X	X	X
CNT_CRITICAL	X	X	X	X
CNT_MINOR	X	X	X	X
CNT_MAJOR	X	X	X	X
MIN_RECEIVE_DELAY	X	X	X	X
MAX_RECEIVE_DELAY	X	X	X	X
AVG_RECEIVE_DELAY	X	X	X	X
TOT_RECEIVE_DELAY	X	X	X	X
MIN_AGE_MSG	X	X	X	X
MAX_AGE_MSG	X	X	X	X
AVG_AGE_MSG	X	X	X	X
TOT_AGE_MSG	X	X	X	X
CNT_AGE_5_MIN	X	X	X	X
CNT_AGE_5_10_MIN	X	X	X	X
CNT_AGE_10_30_MIN	X	X	X	X
CNT_AGE_30_60_MIN	X	X	X	X
CNT_AGE_60_120_MIN	X	X	X	X
CNT_AGE_120_720_MIN	X	X	X	X
CNT_AGE_720_1440_MIN	X	X	X	X
CNT_AGE_GT_1440_MIN	X	X	X	X
CNT_PREV_BUFFERED	X	X	X	X
CNT_DUPL_COUNT	X	X	X	X
CNT_DUPL_MSG	X	X	X	X
CNT_ESCALATE_FLAG	X	X	X	X
CNT_AUTO_ACKN_FLAG	X	X	X	X
CNT_LOG_ONLY_FLAG	X	X	X	X
CNT_NOTIFICATION_FLAG	X	X	X	X
CNT_TROUBLE_TICK_FLAG	X	X	X	X
CNT_ACKN_AFTER_TT_FLAG	X	X	X	X

Reporting Tables – Active Message Property Table

All OVOu Active Message summarized data tables are associated with the K_OVO_ACT property table.

K_OVO_ACT

Column Name	Data type	Population Method
dsi_key_id		Maintained by OVPI
dsi_target_name		Record type indicator based on trend_sum by variables: O: OVO_SERVER ON: OVO_SERVER - NODE_NAME OA: OVO_SERVER - APPLICATION OS: OVO_SERVER - SERVICE_NAME OM: OVO_SERVER - MSG_GROUP ONA: OVO_SERVER – NODE_NAME - APPLICATION ONS: OVO_SERVER – NODE_NAME - SERVICE_NAME ONM: OVO_SERVER – NODE_NAME - MSG_GROUP All Msgs executive summary across all messages A: APPLICATION S: SERVICE_NAME M: MSG_GROUP
dsi_table_key		Due to custom rollup, all by variables, except for GRP_LEVEL and the interval by variable, will be concatenated together and inserted into dsi_table_key
dsi_descr		NULL. Not used
GRP_LEVEL		Same as dsi_target_name
OVO_SERVER	varchar(128)	trend_sum populates based on by variable "OVO_SERVER"

NODE_NAME	varchar(254)
	varchar2(2048)
		trend_sum populates based on by variable "NODE_NAME"

APPLICATION	varchar2(254)	trend_sum populates based on by variable "APPLICATION"
SERVICE_NAME	varchar2(2048)	trend_sum populates based on by variable "SERVICE_NAME"
MSG_GROUP	varchar2(32)	trend_sum populates based on by variable "MSG_GROUP"
APPLICATION_ID	Number	Unique hash for the application
SERVICE_NAME_ID	Number	Unique hash for the service
OTHER standard columns found in OVPI RPs		

Reporting Tables – Active Message Data Tables

The following OVOu Active Message rollup tables are used for reporting:

Destination Table	Rollup Processing Populating Table
SH_OVO_ACT	populated by several rollup processes. SH_OVO_ACT_???.sum
SD_OVO_ACT	populated by trend_sum using SD_OVO_ACT.sum
SW_OVO_ACT	populated by trend_sum using SW_OVO_ACT.sum
SM_OVO_ACT	populated by trend_sum using SM_OVO_ACT.sum

OVOu Active Message metrics provided by time interval:

Metric	Hour	Day	Week	Month
CNT_MESSAGES	X	X	X	X
CNT_UNKNOWN	X	X	X	X
CNT_NORMAL	X	X	X	X
CNT_WARNING	X	X	X	X
CNT_CRITICAL	X	X	X	X
CNT_MINOR	X	X	X	X
CNT_MAJOR	X	X	X	X
MIN_AGE_MSG	X	X	X	X
MAX_AGE_MSG	X	X	X	X
AVG_AGE_MSG	X	X	X	X
TOT_AGE_MSG	X	X	X	X
CNT_AGE_5_MIN	X	X	X	X
CNT_AGE_5_10_MIN	X	X	X	X
CNT_AGE_10_30_MIN	X	X	X	X
CNT_AGE_30_60_MIN	X	X	X	X
CNT_AGE_60_120_MIN	X	X	X	X
CNT_AGE_120_720_MIN	X	X	X	X
CNT_AGE_720_1440_MIN	X	X	X	X
CNT_AGE_GT_1440_MIN	X	X	X	X
CNT_CURR_BUFFERED	X	X	X	X
CNT_PREV_BUFFERED	X	X	X	X
CNT_DUPL_COUNT	X	X	X	X
CNT_DUPL_MSG	X	X	X	X
CNT_ESCALATE_FLAG	X	X	X	X
CNT_AUTO_ACKN_FLAG	X	X	X	X
CNT_LOG_ONLY_FLAG	X	X	X	X
CNT_NOTIFICATION_FLAG	X	X	X	X
CNT_TROUBLE_TICK_FLAG	X	X	X	X
CNT_ACKN_AFTER_TT_FLAG	X	X	X	X

Reporting Tables – Service Log Property Table

All OVOu Service Log summarized data tables are associated with the K_OVO_SRVLOG property table.

K_OVO_SRVLOG

Column Name	Population Method
dsi_key_id	Maintained by OVPI
dsi_target_name	trend_sum populates based on by variable “OVO_SERVER”
dsi_table_key	trend_sum populates based on by variable “SERVICE_NAME”
dsi_descr	NULL. Not used
OVO_SERVER	trend_sum populates based on by variable “OVO_SERVER”
SERVICE_NAME	trend_sum populates based on by variable “SERVICE_NAME”

Reporting Tables – Service Log Data Tables

The following OVOu Service Log rollup tables are used for reporting:

Destination Table	Rollup Processing Populating Table
SD_OVO_SRVLOG	populated by trend_sum using SD_OVO_SRVLOG.sum
SW_OVO_SRVLOG	populated by trend_sum using SW_OVO_SRVLOG.sum
SM_OVO_SRVLOG	populated by trend_sum using SM_OVO_SRVLOG.sum

OVOu Service Log metrics provided by time interval:

Metric	Day	Week	Month
CNT_MESSAGES	X	X	X
DURATION_TOTAL	X	X	X
DURATION_UNKNOWN	X	X	X
DURATION_NORMAL	X	X	X
DURATION_WARNING	X	X	X
DURATION_CRITICAL	X	X	X
DURATION_MINOR	X	X	X
DURATION_MAJOR	X	X	X

Identifying Active and History Message Element Types

When viewing and building reports based on Active or History Message data, it is important to understand how the list of elements can be filtered down to the element type of interest. Each summarized data table contains records for all element types or grouping levels. Refer to the description provided earlier for the K_OVO_ACT and K_OVO_MSG property tables for a list of element types available in each dataset. In these two property tables, two columns can be used to specify which element type should be included in a specific report:

dsi_target_name
GRP_LEVEL

These two columns contain the same information, which is a short character string that should be used by all reporting and processing as a record type indicator.

The dsi_target_name and GRP_LEVEL columns contain the same value, which should be one of the following possible values:

Element Type Identifier **grp_level**	**Element Type Description**
O	OVO_SERVER
ON	OVO_SERVER - NODE_NAME
OA	OVO_SERVER – APPLICATION
OS	OVO_SERVER - SERVICE_NAME
OM	OVO_SERVER - MSG_GROUP
OU (History only)	OVO_SERVER – ACKN_USER
OT	OVO_SERVER - TEMPLATE
OTD	OVO_SERVER – TEMPLATE - DESCRIPTION
ONA	OVO_SERVER – NODE_NAME - APPLICATION
ONS	OVO_SERVER – NODE_NAME - SERVICE_NAME
ONM	OVO_SERVER – NODE_NAME - MSG_GROUP
All Msgs	executive summary across all messages
A	APPLICATION
S	SERVICE_NAME
M	MSG_GROUP
U (History only)	ACKN_USER
T	TEMPLATE
N	NODE
NA	NODE – APPLICATION
NM	NODE – MSG GROUP
NS	NODE – SERVICE
NT	NODE – TEMPLATE
ONT	SERVER – NODE – TEMPLATE
TD	TEMPLATE - CONDITION

When building a report for a particular element type, the drilldown source chart must have constraints on the dsi_target_name column to ensure that only elements of the desired type appear in the report. The required reporting constraints are defined based on the possible contents of the dsi_target_name column provided in the table above. For example, to display a list of all elements of the type “OVO_SERVER - NODE_NAME”, the following constraint should be added to the chart in the Report Builder:

dsi_target_name = ‘ON’   (record type indicator for “OVO_SERVER - NODE_NAME”)

See the definition of K_OVO_MSG and K_OVO_ACT provided earlier in this document for more details on the possible values of dsi_target_name.

Two columns are required to uniquely identify each element within each property table:

dsi_target_name	Element type indicator described above.
dsi_table_key	Unique identity of the element.

The dsi_table_key column can contain a long and sometimes cryptic string.   Separate columns are provided that contain a more human readable identity for each element. The specific columns used to clearly identify a given element depend on the type of element. The following table provides a mapping between the element type and the reporting columns that should be used to display the element’s identity within reports:

The identity columns are contained in an OVPI property tables K_OVO_MSG and K_OVO_ACT. When creating report in the OVPI Report Builder, property table columns of this appear in the available column list when an associated data table is selected.

GRP_LEVEL	**OVO_SERVER**	**NODE_NAME**	**APPLICATION**	**SERVICE_NAME**	**MSG_GROUP**	**ACKN_USER (History Only)**	**TEMPLATE**	**CONDITION DESCRIPTION**
O	**X**							
ON	**X**	**X**						
OA	**X**		**X**					
OS	**X**			**X**				
OM	**X**				**X**			
OU (History only)	**X**					**X**		
OT	**X**						**X**	
OTD	**X**						**X**	**X**
ONA	**X**	**X**	**X**					
ONS	**X**	**X**		**X**				
ONM	**X**	**X**			**X**			
All Msgs								
A			**X**					
S				**X**				
M					**X**			
U (History only)						**X**		

T							**X**	
TD							**X**	**X**
N		**X**						
NA		**X**	**X**					
NM		**X**			**X**			
NS		**X**		**X**				
NT		**X**					**X**	
ONT	**X**	**X**					**X**	

Creating Drilldown Links in Reports

Drilldown link definition within reports can be simple or complex. Normally reports are built with a Tabular Chart at the top listing elements of interest. This chart is the "**drilldown source**". When an element is selected in the chart, the "**drilldown destination**" graphs in the lower part of the report are updated to reflect the selected element.

In the OVO Report Pack, if the drilldown source and drilldown destination charts are to display the same element type, a simple "link by element ID" link can be created between the two charts. A custom drilldown link is not required in that case. For example, if the drilldown source tabular chart contains a list of OVO_SERVER-APPLICATION (OA) elements, and the drilldown destination graph is intended to display a long term trend for the OVO_SERVER-APPLICATION element selected in the drilldown source chart, both charts will display the same element type and a simple "link by element ID" link can be used between the two charts.

If the drilldown source and drilldown destination charts are to display different element types, a simple "link by element ID" link can not be used between the two charts. A "custom" drilldown relationship must be defined based on the specific element types displayed in the charts. An example should clarify the issues.

Drilldown Links Between Different Element Types

This example provides information on creating drilldown links from the OVO_SERVER-APPLICATION (OA) level to the OVO_SERVER-NODE-APPLICATION (ONA) level. This type of drilldown analysis enables the user to first view a list of problematic applications and then drilldown to a list of nodes that have generated messages for the selected application so that the user can determine which nodes are experiencing the most problems with the selected application.

OVO_SERVER-APPLICATION → OVO_SERVER-NODE-APPLICATION

Linking from OA to ONA:

Drilldown Source: (Tabular listing **Applications by OVOu Server**)

Build from SD_OVO_MSG for daily.
Display the OVO_SERVER and APPLICATION columns in exception list
Constraint DSI_TARGET_NAME = 'OA'

Drilldown Destination: (Tabular listing **Nodes for selected Applications/OVOu Server**)

Build from SD_OVO_MSG for daily.
Display the NODE_NAME column.
Constraint DSI_TARGET_NAME = 'ON' (OVO_SERVER - NODE)

The resulting list of nodes should only include nodes that generated messages for the selected **Application and OVOu Server**.

Insert Link Relationship:

Custom link definition:

```
APPLICATION = APPLICATION
OVO_SERVER = OVO_SERVER
```

Appendix A: History and Active Message Rollup Definition

A sophisticated method is used to summarize to multiple groupings or element/record types. All summarization processes populate a single set of destination tables. Using fewer tables within the solution simplifies administration and support issues.

Defining Unique Element IDs

Within OVPI, two columns are used to uniquely identify each element within a given property table. Those columns are:

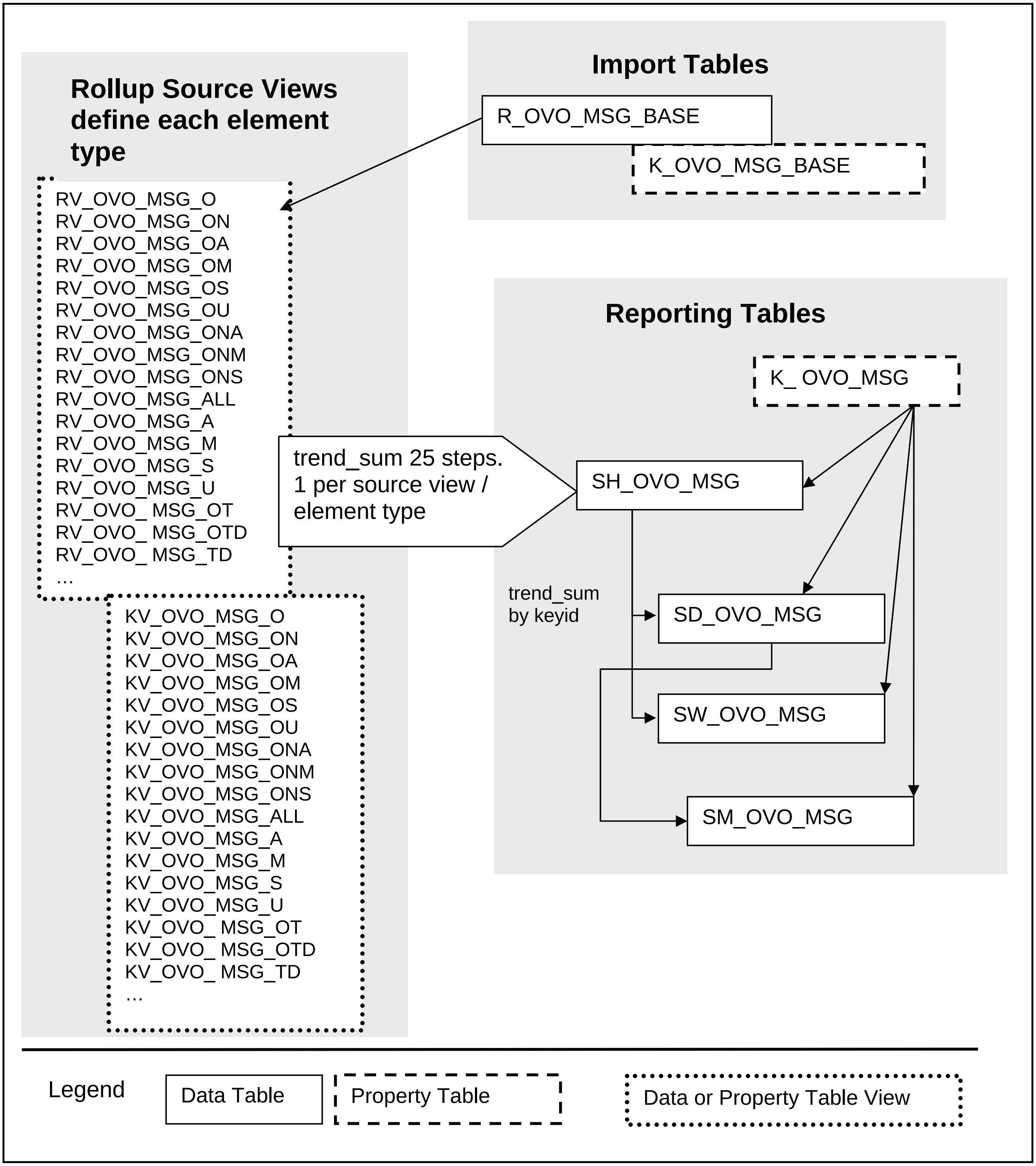
dsi_target_name
dsi_table_key

Care must be taken to ensure that dsi_target_name and dsi_table_key are populated with a unique value for each unique element when summarizing data for multiple element types into a single set of destination tables. In the case of OVOU HIST data, 14 different element types of data are summarized into one set of destination tables.

To ensure the dsi_target_name and dsi_table_key are populated with unique values in the destination property table, database views are used as the source data table and the source property table by the trend_sum rollup processes. A different pair of data/property table views is used for each element type. Data is rolled up from these views to the hourly level. From there, data is summarized to daily, weekly, and monthly intervals in using only “keyid” as the element level by variable because the element grouping levels/types are already defined in the hourly data and do not have to be “redefined” when rolling up from that table.

In the rollup source database views, a “Element Type” column called GRP_LEVEL is populated with the acronym of the element type presented by each property table view. This GRP_LEVEL column is used as the first by_variable in each hourly trend_sum rollup process, which becomes dsi_target_name in the destination property table.

Rollup Diagram – Consolidated Messages

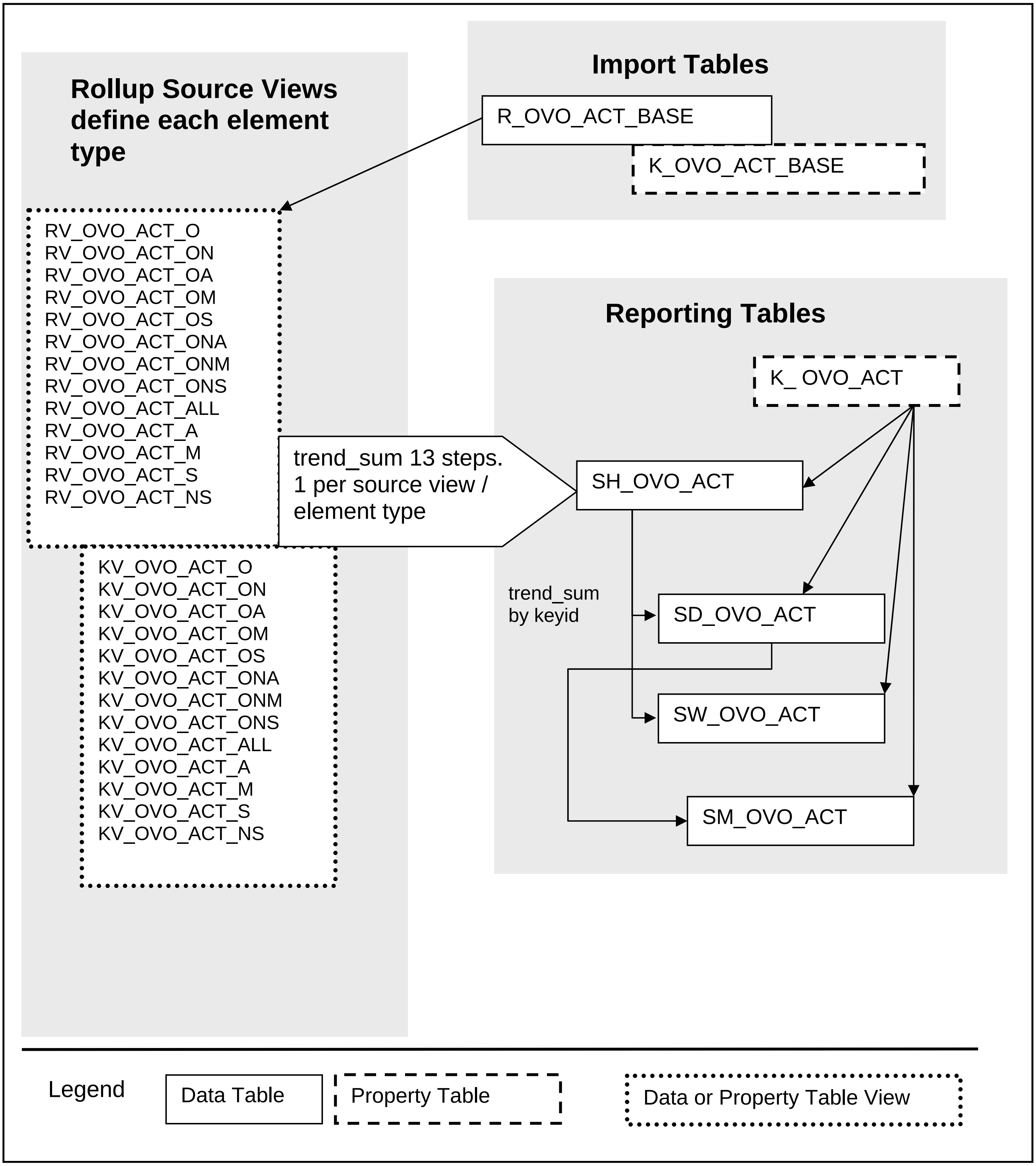


Hourly Rollup Source Table Views – History Messages

The **GRP_LEVEL** column of each rollup source view is defined as:

GRP_LEVEL Indicator	**Grouping Level**	**Rollup Source Views**
O	OVO_SERVER	RV_OVO_MSG_O KV_OVO_MSG_O
ON	OVO_SERVER - NODE_NAME	RV_OVO_MSG_ON KV_OVO_MSG_ON
OA	OVO_SERVER - APPLICATION	RV_OVO_MSG_OA KV_OVO_MSG_OA
OS	OVO_SERVER - SERVICE_NAME	RV_OVO_MSG_OS KV_OVO_MSG_OS
OM	OVO_SERVER - MSG_GROUP	RV_OVO_MSG_OM KV_OVO_MSG_OM
OU	OVO_SERVER – ACKN_USER	RV_OVO_MSG_OU KV_OVO_MSG_OU
ONA	OVO_SERVER – NODE_NAME - APPLICATION	RV_OVO_MSG_ONA KV_OVO_MSG_ONA
ONS	OVO_SERVER – NODE_NAME - SERVICE_NAME	RV_OVO_MSG_ONS KV_OVO_MSG_ONS
ONM	OVO_SERVER – NODE_NAME - MSG_GROUP	RV_OVO_MSG_ONM KV_OVO_MSG_ONM
All Msgs	executive summary across all messages	RV_OVO_MSG_ALL KV_OVO_MSG_ALL
A	APPLICATION	RV_OVO_MSG_A KV_OVO_MSG_A
S	SERVICE_NAME	RV_OVO_MSG_S KV_OVO_MSG_S
M	MSG_GROUP	RV_OVO_MSG_M KV_OVO_MSG_M
U	ACKN_USER	RV_OVO_MSG_U KV_OVO_MSG_U
T	TEMPLATE	RV_OVO_MSG_T KV_OVO_MSG_T
TD	TEMPLATE - CONDITION	RV_OVO_MSG_TD KV_OVO_MSG_TD
OT	OVO_SERVER – TEMPLATE	RV_OVO_MSG_U KV_OVO_MSG_OT

Rollup Diagram – Active Messages



Hourly Rollup Source Table Views – Active Messages

The **GRP_LEVEL** column of each rollup source view is defined as:

GRP_LEVEL Indicator	**Grouping Level**	**Rollup Source Views**
O	OVO_SERVER	RV_OVO_ACT_O KV_OVO_ACT_O
ON	OVO_SERVER - NODE_NAME	RV_OVO_ACT_ON KV_OVO_ACT_ON
OA	OVO_SERVER - APPLICATION	RV_OVO_ACT_OA KV_OVO_ACT_OA
OS	OVO_SERVER - SERVICE_NAME	RV_OVO_ACT_OS KV_OVO_ACT_OS
OM	OVO_SERVER - MSG_GROUP	RV_OVO_ACT_OM KV_OVO_ACT_OM
ONA	OVO_SERVER – NODE_NAME - APPLICATION	RV_OVO_ACT_ONA KV_OVO_ACT_ONA
ONS	OVO_SERVER – NODE_NAME - SERVICE_NAME	RV_OVO_ACT_ONS KV_OVO_ACT_ONS
ONM	OVO_SERVER – NODE_NAME - MSG_GROUP	RV_OVO_ACT_ONM KV_OVO_ACT_ONM
A	APPLICATION	RV_OVO_ACT_A KV_OVO_ACT_A
S	SERVICE_NAME	RV_OVO_ACT_S KV_OVO_ACT_S
M	MSG_GROUP	RV_OVO_ACT_M KV_OVO_ACT_M
All Msgs	executive summary across all messages	RV_OVO_ACT_ALL KV_OVO_ACT_ALL
NS	NODE_NAME – SERVICES	RV_OVO_ACT_NS KV_OVO_ACT_NS

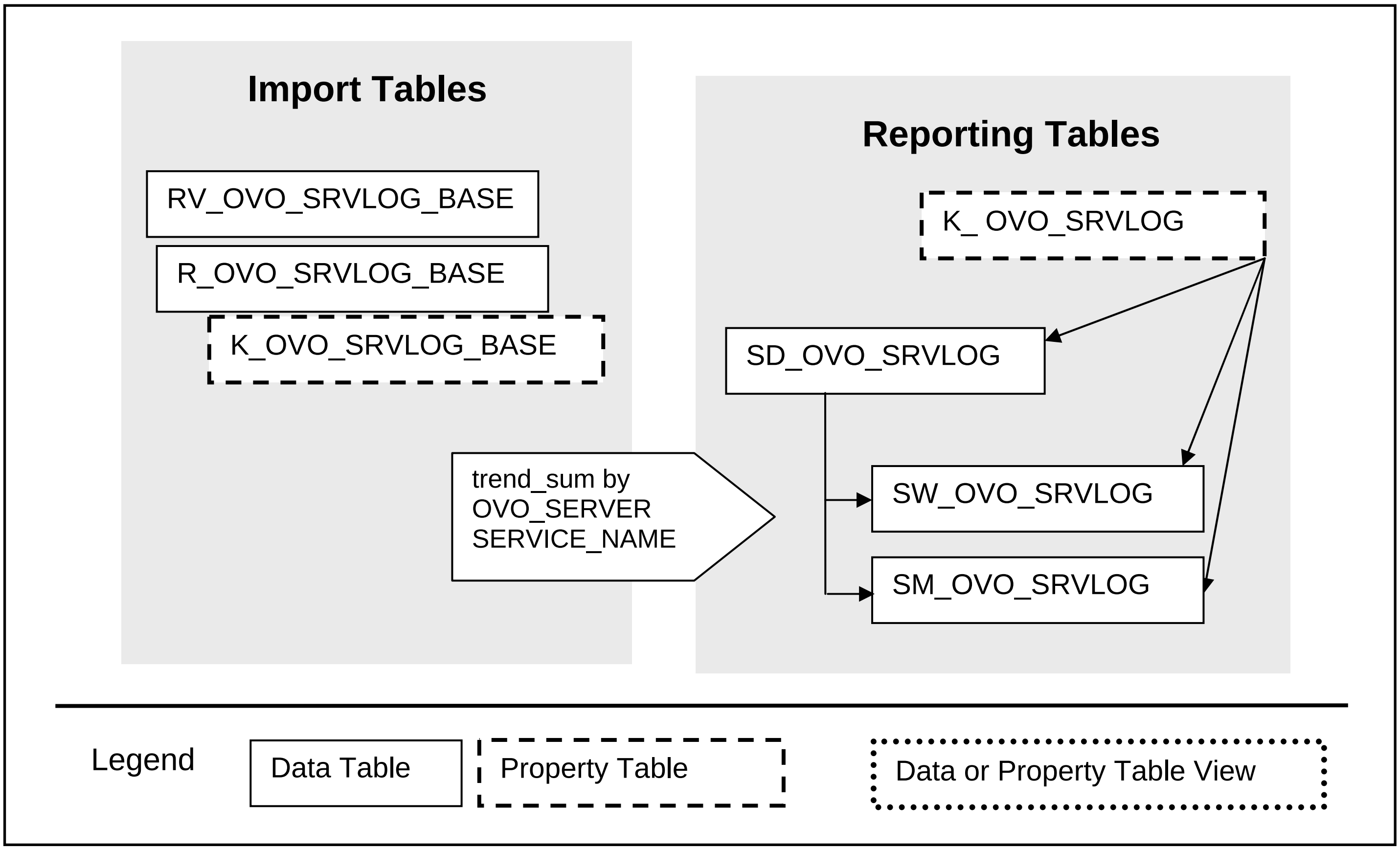
Note that 12 separate trend_sum files are used in 12 separate trend_sum rollup steps to rollup each Element Type to the hourly level. The trend_sum file names:

SH_OVO_ACT_<??>.sum

(<??> = O, ON, OA, OS, OM, ONA, ONS, ONM, All, A, S, M)

Appendix B: Service Log Rollup Definition

The rollup process used for Service Log data is a straightforward trend_sum rollup as depicted in the diagram below.

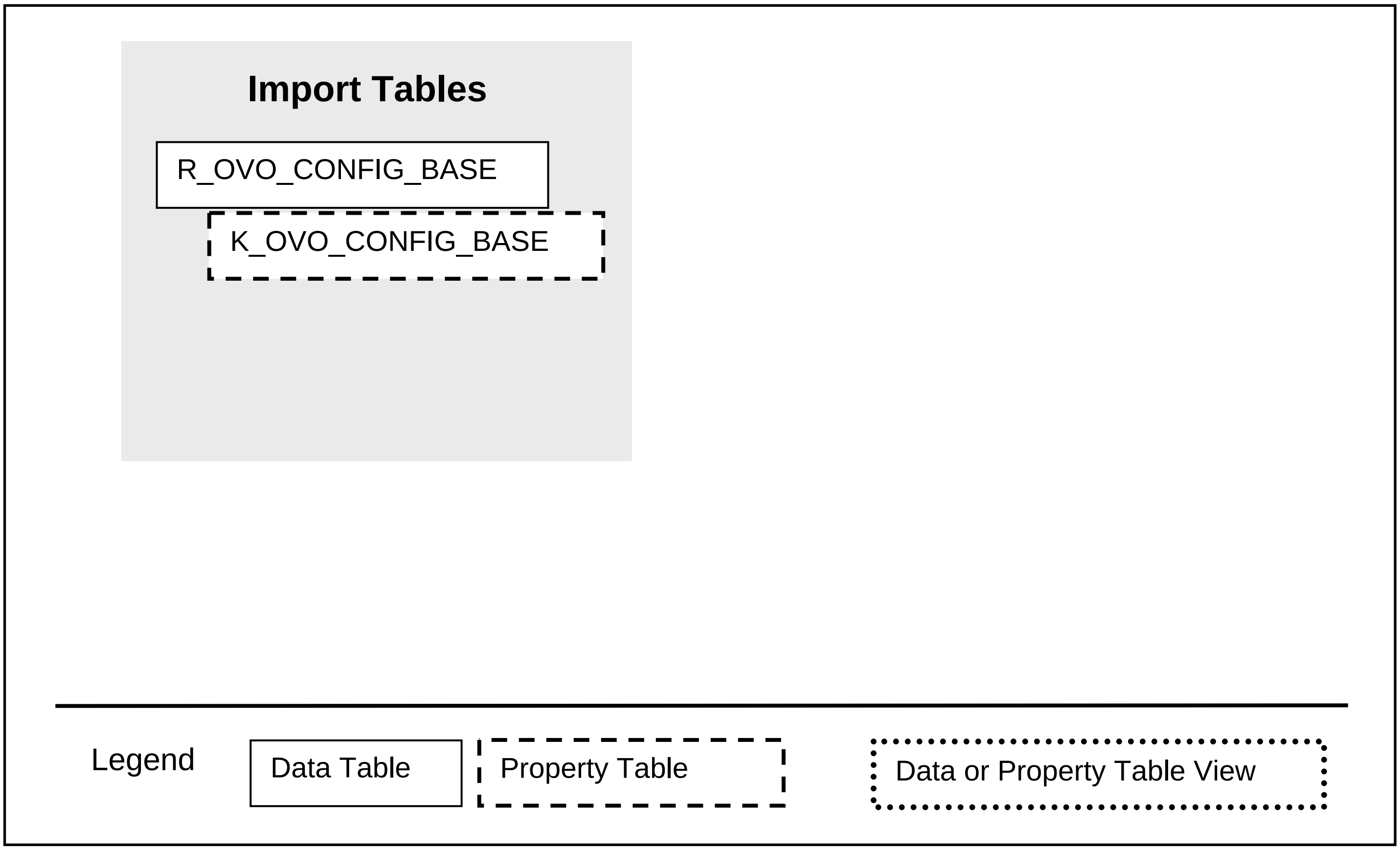


All reporting tables containing OVOU Service Log data are associated with the K_OVO_SRVLOG property table. Within this property table, elements are uniquely identified as follows:

dsi_target_name = OVO_SERVER
dsi_table_key = SERVICE_NAME

Appendix C: Configuration Rollup Definition

The collection process used for Configuration data is straightforward. It is simply a data import into a raw table. Due the the nature of the data, no rollups are performed. It is simply used as a history table.



Appendix D: Uninstalling the OVO Report Pack

Follow these steps to uninstall OVO Report Pack:

Warning: If you remove a report pack, the associated tables and all the data in those tables will be deleted. If you want to preserve the data in those tables, archive the data before removing the report pack.

1. Log in to the system. On UNIX systems, log in as root.

2. Remove the Active Message table Views from each OVOu server’s database using the following command on the OVPI server:

```
ovou_datapipe -dataset drop_view
```

3. Stop OVPI Timer and wait for processes to terminate.

   Windows: Select Settings > Control Panel > Administrative Tools > Services
   UNIX: As root, type one of the following:
      HP-UX: sh /sbin/init.d/ovpi_timer stop
      Sun: sh /etc/init.d/ovpi_timer stop

4. Start Package Manager. The Package Manager welcome window opens.

5. Click Next. The Package Location window opens.

6. Click Uninstall.

7. Click Next. The Report Undeployment window opens. If the reports for this Report Pack were deployed from this server, accept the defaults for Undeploy Reports, Application Server Name, and Port. Otherwise, clear the check box and skip to step 8.

8. Type your username and password for the OVPI Application Server.

9. Click Next. The Package Selection window opens. Click the check box next to the following package:

   OVO_Reporting

10. Click OK.

11. Click Next. The Selection Summary window opens.

12. Click Uninstall. The Progress window opens and the removal process begins. When the uninstall process is complete, a package removal complete message appears.

13. Click Done to return to the Management Console.

14. Restart OVPI Timer.
   Windows: Select Settings > Control Panel > Administrative Tools > Services
   UNIX: As root, type one of the following:

HP-UX: sh /sbin/init.d/ovpi_timer start

Sun: sh /etc/init.d/ovpi_timer start