System Resource OVPA Datapipe

Software Version: 3.30

HP Performance Insight 5.40

User Guide

February 2009

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software Support Online provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Contents

1 Overview

The System Resource Report Pack contains a main package and multiple sub-packages. The main package and the sub-packages do not share the same datapipes. The main package can use three datapipes:

- SysRes RFC1514 Datapipe 4.10
- OVPA Collection Datapipe 1.10
- SysRes OVPA Datapipe 3.30

Each sub-package has a built-in datapipe. For more information about the data collected by each built-in datapipe, refer to the *System Resource Report Pack User Guide*.

SysRes 1514 Datapipe

The SysRes RFC 1514 Datapipe collects the following data from the Host Resources MIB:

- CPU utilization
- Memory utilization
- Number of page outs
- Run queue length
- Swap utilization

OVPA Collection Datapipe

The OVPA Collection Datapipe performs the following tasks:

- Discovers PA or EPC performance agents
- Inserts pa_collect statements into the trendtimer.sched file
- Populates the OVPA type group for use in the collection process

The discovery process runs once a day. It is controlled by the OVPA_Collection_Daily.pro file. The discovery process attempts to check all of the nodes in ksi_managed_node table. Because SNMP discovery is only capable of discovering SNMP-pollable devices, you may want to add PA or EPC systems using the create node tool.

Once the discovery process starts, it will write to the pa_discovery.data file in the DPIPE_HOME/data directory. The pa_discovery.data file contains the systems the discovery process finds. After the discovery process is complete, ee_collect loads the pa_discovery.data file into PI. The data is placed in the K_padatasources table.

SysRes OVPA Datapipe

The SysRes OVPA Datapipe collects performance data from two agents:

- HP Performance Agent (HP PA)
- Embedded Performance Component of HP Operations (HP OA)

HP OA, which is also known as EPC, ships with OpenView Operations for Unix (HPOU) 7.00 and OpenView Operations for Windows (HPOW) 7.00. If you are using HPOU 7.00 or HPOW 7.00, the SysRes OVPA Datapipe collects metrics from EPC. If you are using an earlier version of HP Operations, UNIX or Windows, the SysRes OVPA Datapipe collects metrics from HP PA.

The metrics gathered by HP PA are different from the metrics gathered by EPC. For details about how the metrics vary, refer to *Metrics for HP Performance Agent and Operations Agent.* You can download this document from the HP Software support site. It appears in two places on the Product Manuals Search page, under *Operations for Windows,* and also under *Operations for UNIX*.

The SysRes OVPA Datapipe has one prerequisite, the OVPA Collection Datapipe.

The following table outlines recent enhancements to the SR OVPA Datapipe.

Version	Release Date	Features/Enhancements
1.00	May 2003	Initial release.
2.00	October 2003	Oracle support.
2.50	April 2004	Defect fixed: • ROSmm36178: related to redundant collections and constraint violation.
2.60	November 2004	Node discovery moved to SysRes OVPA Collection Datapipe.
2.80	June 2005	Defect fixed: • QCCR1000214002: related to incorrect system availability data.
3.00	May 2006	Defects fixed: • QXCR1000301194 • QXCR1000247763TBD
3.10	April 2007	Defects fixed: • QXCR1000414003 • QXCR1000393795 New upgrade package: • UPGRADE_SysRes_OVPA_Datapipe_to_31

Version	Release Date	Features/Enhancements
3.20	October 2007	Defect fixed: • QXCR1000452475 - PI not able to collect INTERVAL metric from Coda A.07.50.06 New metrics collected: • LS Role (HOST/GUEST/STAND) • LS Type (HPVM or AIX LPAR) Role and Type columns added to this property table: • K_OVPA_config Role and Type columns added to these data tables: • xSR_OVPA_config • RSR_OVPA_config Modified the following SQL: • update_device_properties.sql New upgrade package: • UPGRADE_SysRes_OVPA_Datapipe_to_32
3.30	Feb09	New upgrade package: • SysRes_OVPA_Datapipe_Upgrade_to_33 Defect fixes: • QXCR1000786806 - K_OVPA_CONFIG table does not get updated • QXCR1000791798 - Discrepancies between Reported System Availability and reality • QXCR1000793411 - Missing data and information for the File System Reports

Sources for Additional Information

The following documents are related to this manual:

- *SysRes OVPA Datapipe Release Statement*
- *OVPA Collection Datapipe Release Statement*
- *SysRes RFC1514 Datapipe Release Statement*
- *Metrics for HP Performance Agent and Operations Agent, January 2005*
- *HP Performance Agent Dictionary of Operating System Performance Metrics*
- *HP Performance Agents Metrics Help Text*
- *System Resource Report Pack User Guide*
- *PI Report Packs, CD-ROM Release Notes, February 2009*

The user guides for PI and the user guides for the report packs and datapipes that install on PI can be downloaded from here:

http://h20230.www2.hp.com/selfsolve/manuals

The user guides for PI are listed under **Performance Insight**. The user guides for report packs and datapipes are listed under **Performance Insight Reporting Solutions**. Every user guide indicates the month and year it was posted to the web. If a manual is revised and reposted, the date will change. Since revised manuals are reposted from time to time, be sure to compare your PDF to the web edition and download the web edition if it is newer.

2 Installation and Agent Discovery

This chapter covers the following topics:

- Guidelines for a smooth install
- Prerequisites
- Installing the SysRes OVPA Datapipe 3.30
- Verifying system discovery
- Configuring a remote poller

Guidelines for a Smooth Install

The reporting solutions created for PI consist of two installable packages, a report pack and a datapipe. Some report packs have multiple datapipes. When you install the datapipe, you configure PI to collect a specific type of performance data at a specific polling interval. When you install the report pack, you configure PI to summarize and aggregate the data collected by the datapipe.

The report pack CD-ROM contains report packs, datapipes, and several packages that are shared by multiple report packs. If the contents of the CD were extracted to the Packages directory on your system, then every package, including the Sys Res OVPA Datapipe, is available for installation and can be installed by starting Package Manager and following the on-screen instructions. If the contents of the CD-ROM have not been extracted, follow the extract procedure later in this chapter.

The SysRes OVPA Datapipe is not a prerequisite for the System Resource Report Pack. You may install the SysRes OVPA Datapipe and the System Resource Report Pack at the same time, or you may install the SysRes OVPA Datapipe later, after you install the report pack. The SysRes OVPA Datapipe has one prerequisite, the OVPA Collection Datapipe. If you do not select the OVPA Collection Datapipe for installation, Package Manager will select and install this datapipe for you, automatically.

Installing the SysRes OVPA Datapipe

Follow these steps to install the SysRes OVPA Datapipe:

Task 1: Extract packages from the report pack CD

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

3 Insert the report pack CD in the CD-ROM drive. On Windows, a Main Menu displays automatically; on UNIX, mount the CD, navigate to the top-level directory for the CD drive, and type the setup command.

4 Select PI report packs by typing **1** in the choice field and pressing Enter. The install script displays a percentage complete bar. When extraction finishes, the install script starts Package Manager. The Package Manager welcome window opens.

5 When prompted, type your PI username and password.

Task 2: Install the datapipes

If you just extracted files from the product CD, start at step 2. Otherwise, begin at step 1.

1 From the Management Console select **Tools > Package Manager**. The Package Manager welcome window opens.

2 Click **Next.** The Package Location window opens.

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

4 Click **Next.** The Report Deployment window opens. Accept the default for Deploy Reports; accept the default for application server name and port. Type your username and password for the PI Application Server.

5 Click **Next.** The Package Selection window opens.

6 If you are currently running version 3.20, click the check box next to the upgrade package:

SysRes_OVPA_Datapipe_Upgrade_to_33

7 If you are installing the OVPA Datapipe for the first time, click the check boxes next to these packages:

SysRes OVPA Datapipe 3.3

OVPA Collection Datapipe 1.1

8 Click **Next.** The Type Discovery window opens.

9 Disable the default to run Type Discovery immediately after package installation.

10 Click **Next.** The Selection Summary window opens.

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

12 Click **Done** to return to the Management Console.

13 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`**

Verifying Agent Discovery

There two ways to find out whether any systems have been discovered. The easier way is to open Table Viewer from the Management Console and look at what the property table contains. The other way is to log into the database and run the following command:

```
SELECT * FROM K_padatasources
```

If no rows appear, no systems have been discovered.

If there are systems in K_padatasources, those systems should also be in the OVPA type group. Systems are placed into the OVPA type group by the following SQL stored procedure:

```
SR_OVPA_SetOVPAType
```

SR_OVPA_SetOVPAType runs once a day. It is executed by the following SQL script:

```
execute_SR_OVPA_SetOVPAType.sql
```

The script is located in the SR_OVPA_Daily.pro file.

To see the contents of the OVPA type group, use the Group Manager feature within the Management Console. If there are no systems in the OVPA type group, follow these steps:

1. From the Management Console, select Polling Policy Manager.
2. Use Polling Policy Manager to add systems to the polling group.
3. Verify that pa_collect is able to collect statistics from the systems you added; if statistics are being collected from a system, you know that the system is running an OVPA/EPC agent.
4. If you cannot collect data from a system in the OVPA type group, remove that system from the group.

If you already have systems defined in PI, you can discover the systems that support OVPA/EPC by navigating to the DPIPE_HOME/scripts directory and running the following command:

```
trend_proc -f OVPA_Collection_Daily.pro
```

Configuring a Remote Poller

Manual steps are required to configure a remote poller for OVPA. At the system where you installed the SysRes OVPA Datapipe and the OVPA Collection Datapipe, follow these steps:

1. Stop OVPI Timer on the remote poller.
2. Navigate to the DPIPE_HOME/data directory.
3. Copy the pa_rpt.cnfg file to the DPIPE_HOME/data directory on the remote poller.
4. Navigate to the DPIPE_HOME/collect/SR directory.
5. Copy the bcpg files to the DPIPE_HOME/collect/SR directory on the remote poller.
6. Using the trendtimer.sched file as a template, add the pa_collect entries to the trendtimer.sched file on the remote poller.

7 Create polling policies on the PI system the remote poller is using.

8 Restart OVPI Timer on the remote poller.

It is important to have the bcpg files in place before running the first pa_collect on the remote poller. If for some reason pa_collect runs before the bcpg files are in place, you will need to remove the *.def files in the DPIPE_HOME/collect/SR/60 directory on the remote poller. If you install any of the sub-packages that come with System Resource, you will need to copy those bcpg files over as well, and create the collection polices for the remote pollers.

3 Data Collection and Tables

This chapter covers the following topics:

- Data collection
- Data tables, property tables, and the associated TEEL files
- Metrics collected by the SysRes OVPA Datapipe
- Daily and hourly processes

Data Collection

The SysRes OVPA Datapipe collects 5 types of metrics. All collections except one, Configuration, occur hourly.

Metric	**Frequency**
Global	Hourly
Filesystem	Hourly
Transaction	Hourly
Application	Hourly
Configuration	Daily

If a system is in the OVPA type group, the datapipe collects all five metric types from the system.

All collections except the configuration collection are history-styled collections. When the system is polled, the datapipe returns multiple rows back and each row has a distinct *ta_period*. The first time the collection runs, all rows beginning from midnight the previous day will be returned. The next collection is smaller. It returns data that has not already been collected.

The history property tables contain the last *ta_sysuptime* value in the lastPoll column. For each collection that follows the first collection, the poller requests data that meets the following criterion: time stamp > value of *lastPoll*

To change the granularity of the data, use the **-E** option for *pa_collect*. For example if you want data granularity of 15 minutes instead of an hour, type this command:

```
pa_collect -i 60 -E 15
```

Changing the frequency of polling is not recommended.

Data Tables and Property Tables

The following tabls are defined by TEEL files:

- Raw tables
- Rate tables
- Property tables
- History property tables

Use the information below to find out which TEEL file defines each table.

Data Tables (Raw and Rate)

Raw Table	Rate Table	Property Table	TEEL File
xSR_OVPA_config	RSR_OVPA_config	K_OVPA_config	SysResOVPA_config.teel
xSR_OVPA_filesystem	RSR_OVPA_filesystem	K_OVPA_filesystem	SysResOVPA_filesystem.teel
xSR_OVPA_global	RSR_OVPA_global	K_OVPA_global	SysResOVPA_global.teel
xSR_OVPA_application	RSR_OVPA_application	K_AppOVPA_application	SysResOVPA_application.teel
xSR_OVPA_transaction	RSR_OVPA_transaction	K_TranOVPA_tran	SysResOVPA_transaction.teel

Property Tables

Property Table	TEEL File
K_OVPA_config	prop_SysResOVPA_config.teel
K_OVPA_filesystem	prop_SysResOVPA_filesystem.teel
K_OVPA_global	prop_SysResOVPA_global.teel
K_AppOVPA_application	prop_AppOVPA_app.teel
K_TranOVPA_tran	prop_TranOVPA_tran.teel
K_padatasources	prop_SysResOVPA_padatasources.teel

History Property Table Matrix

Property Table	History Property Table	Teel File
K_OVPA_filesystem	K_OVPA_filesystem_history	prop_SysResOVPA_filesystem_history.teel
K_OVPA_global	K_OVPA_global_history	prop_SysResOVPA_global_history.teel
K_AppOVPA_application	K_AppOVPA_app_history	prop_AppOVPA_app_history.teel
K_TranOVPA_tran	K_TranOVPA_tran_history	prop_TranOVPA_tran_history.teel

Metrics Collected by the SysRes OVPA Datapipe

The SysRes OVPA Datapipe collects metrics from the following metric classes:

- Configuration
- Filesystem
- Application
- Transaction
- Global

Configuration Metrics

The collection table is RSR_OVPA_config. Supported metrics vary from platform to platform and from OVPA to EPC.

Metric	Description
GBL_COLLECTOR	Collector name and version
GBL_LS_ROLE	Options are HOST/GUEST/STAND
GBL_LS_TYPE	Vendor; options are HPVM or AIX LPAR
GBL_MACHINE	System type
GBL_MEM_FREE	Amount of memory not allocated in the system
GBL_MEM_PHYS	Amount of physical memory in the system
GBL_NUM_CPU	Number of CPUs on the system
GBL_NUM_DISKS	Number of disks on the system
GBL_NUM_NETWORKS	Number of LAN interfaces on the system
GBL_OSNAME	Operating system
GBL_OSRELEASE	Current release of the operating system
GBL_OSVERSION	Operating system version
GBL_SYSTEM_ID	System ID

Filesystem Metrics

The collection table is RSR_OVPA_filesystem. Supported metrics vary from platform to platform and from OVPA to EPC.

Metric	Description
FS_DEVNAME	Path name string of the current device
FS_DEVNO	Internal device number that the OS associates with a disk device
FS_DIRNAME	The filesystem name or drive letter
FS_TYPE	The filesystem type
FS_MAX_SIZE	Maximum size of the filesystem in MBs
FS_SPACE_UTIL	Percentage of the file system space in use during the interval
INTERVAL	Number of seconds in the measurement interval

Application Metrics

The collection tables RSR_OVPA_application. Supported metrics vary from platform to platform and from OVPA to EPC.

Metric	Description
APP_NAME	The name of the application (up to 20 characters). The name comes from the parm file where the applications are defined. The application called “other” captures all processes not aggregated into applications specifically defined in the parm file. If no applications are defined in the parm file, then all process data will be reflected in the “other” application. HP-UX If the parm file switch to log PRM group data, instead of application data, is in effect (indicated by APP_PRM_LOGGING_MODE = 1 and the log statement in the parm file includes application=prm), then this name is the PRM groupname defined in the HP-UX Process Resource Manager configuration file. WinNT The name of the Windows module for this application.

Metric	Description
APP_CPU_TOTAL_UTIL	The percentage of the total CPU time devoted to processes in this group during the interval. This indicates the relative CPU load placed on the system by processes in this group. On a system with multiple CPUs, this metric is normalized. That is, the CPU used over all processors is divided by the number of processors online. This represents the usage of the total processing capacity available. Large values for this metric may indicate that this group is causing a CPU bottleneck. This would be normal in a computation-bound workload, but might mean that processes are using excessive CPU time and perhaps looping.
APP_CPU_TOTAL_TIME	The total CPU time, in seconds, devoted to processes in this group during the interval. On a system with multiple CPUs, this metric is normalized. That is, the CPU used over all processors is divided by the number of processors online. This represents the usage of the total processing capacity available.
APP_MEM_VIRT	The approximate size (in KB) of virtual memory for processes in this group that were alive at the end of the interval. HP-UX / SunOS This is the sum of the virtual memory region sizes for all processes in this group. Since this virtual memory size for each process includes shared regions, such as library text and data, the shared regions are counted multiple times in this metric. For example, if two processes are attached to a 10MB shared region, then 20MB is reported in this metric. This value is not affected by the reference count. As such, this metric can overestimate the virtual memory being used by processes in this group when they share memory regions. WinNT The size (in KB) of paging file space used for processes in this group during the interval. This is the sum of the pagefile space used for all processes in this group. Groups of processes may have working set sizes (APP_MEM_RES) larger than the size of their pagefile space.

Metric	Description
APP_ALIVE_PROC	An alive process is one that exists on the system. APP_ALIVE_PROC is the sum of the alive-process-time/ interval-time ratios for every process belonging to a given application. The following diagram shows two processes over a 4-second interval. Note the difference between active processes, which consume CPU time, and alive processes which merely exist on the system. ---------- Seconds ---------- 1 2 3 4 Proc ---- ---- ---- ---- ---- A live live live live B live/CPU live/CPU live dead Process A is alive for the entire interval but consumes no CPU. A's contribution to APP_ALIVE_PROC is 4*1/4. A contributes 0*1/4 to APP_ACTIVE_PROC. B's contribution to APP_ALIVE_PROC is 3*1/4. B contributes 2*1/4 to APP_ACTIVE_PROC. Thus, for this interval, APP_ACTIVE_PROC equals 0.5 and APP_ALIVE_PROC equals 1.75. Because a process may be alive but not active, APP_ACTIVE_PROC will always be less than or equal to APP_ALIVE_PROC.
APP_ALIVE_PROC (con't)	SunOS / WinNT This metric is derived from sampled process data. Since the data for a process is not available after the process has died on this operating system, a process whose life is shorter than the sampling interval may not be seen when the samples are taken. Thus this metric may be slightly less than the actual value. Increasing the sampling frequency captures a more accurate count, but the overhead of collection may also rise.
APP_ACTIVE_PROC	The number of seconds during the interval that this application had windows in the active state.
INTERVAL	Number of seconds in the measurement interval.

Transaction Metrics

The collection table is RSR_OVPA_transacation. Supported metrics vary from platform to platform and from OVPA to EPC.

Metric	Description
TT_ABORT	Number of aborted transaction.
TT_ABORT_WALL_TIME_PER_TRAN	Average time in sec per aborted transaction.
TT_APP_NAME	The registered ARM application name.

Metric	Description
TT_COUNT	Number of completed transactions.
TT_NAME	The registered transaction Name.
TT_SLO_COUNT	Total number of page outs to the disk per second.
TT_SLO_PERCENT	Number of successful packets per second received through all network interfaces.
TT_SLO_THRESHOLD	Number of successful packets received through all network interfaces.
TT_WALL_TIME_PER_TRAN	Number of successful packets per second sent through all network interfaces.
INTERVAL	Number of seconds in the measurement interval.

Global Metrics

The collection table is RSR_OVPA_global. Supported metrics vary from platform to platform and from OVPA to EPC agents.

Metric	Description
GBL_ACTIVE_PROC	The total alive-process-time/interval-time ratios of every process that is active (uses any CPU time) during an interval.
GBL_ALIVE_PROC	The total alive-process-time/interval-time ratios for every process.
GBL_CPU_TOTAL_TIME	The total time, in seconds, that the CPU was not idle.
GBL_CPU_TOTAL_UTIL	Percentage of time the CPU was not idle.
GBL_DISK_PHYS_IO	The number of physical IOs.
GBL_MEM_PAGEOUT_RATE	The total number of page outs to the disk per second.
GBL_NET_IN_PACKET_RATE	Number of successful packets per second received through all network interfaces.
GBL_NET_IN_PACKET	Number of successful packets received through all network interfaces.
GBL_NET_OUT_PACKET_RATE	Number of successful packets per second sent through all network interfaces.
GBL_NET_OUT_PACKET	Number of successful packets sent through all network interfaces.
GBL_RUN_QUEUE	Average number of –runnable • processes over all processors.
GBL_STARTED_PROC	Number of processes that started during the interval.

Metric	Description
GBL_SWAP_SPACE_UTIL	The percent of swap space used.
GBL_MEM_UTIL	The percent of physical memory in use.
INTERVAL	Number of seconds in the measurement interval.

Daily and Hourly Processes

At 1:00 a.m., when the SR_OVPA_Daily.pro file is invoked, two events take place:

1. The OVPA type group is populated.
2. The update_device_properties.sql script runs.

The update_device_properties.sql script updates K_OVPA_config and K_Node with system information. The information added to K_Node is used to identify the vendor, system model, and the operating system.

Once an hour, when the SR_OVPA_Hourly.pro file is invoked, the following events take place:

1. New systems are processed.
2. Two tables are updated:
   - Filesystem property table
   - Transaction property table
3. Data is mapped from datapipe rate tables to report pack base tables.

For details about data mapping, see the next chapter.

Processing a New System

New systems are found and added to K_Node before the summaries run. This stored procedure is responsible for adding a new system to K_Node:

ppSR_OVPA_SetIDs

It is invoked by this SQL script:

execute_SR_OVPA_SetIDs.sql

The stored procedure performs two actions:

- Populates K_Node with the new system
- Adds the dsi_key_id value for that system from K_Node to the node_fk column in the datapipe property table

Updating Tables

The filesystem update is update_filesystem_property_table.sql. The file system update populates the K_OVPA_filesystem property table with the filesystem name.

The transaction update is update_Tran_ProTables.sql. On a per-transaction basis, the transaction update populates the SLO_Threshold column in the property table K_Transaction_ByTran, using values from the SLO_Threshold column in the K_TranOVPA_tran property table.

4 Hourly Data Mapping

This chapter covers the following topics:

- SR_OVPA_Hourly.pro file
- Mapping RSR_OVPA_global to SR_SR_Kernel
- Mapping RSR_OVPA_filesystem to SR_SR_Filesystem
- Mapping RSR_OVPA_global to SR_SR_SysVol
- Mapping RSR_OVPA_application to SRApp_application
- Mapping RSR_OVPA_transaction to SRTran_Transaction

If you want to know more about the collections performed by the sub-packages (CPU, Process, Disk, Logical System, Logical Volume, Network Interface), refer to Chapter 4, Data Collection, in the *System Resource Report Pack 4.30 User Guide*.

SR_OVPA_Hourly.pro

The scripts in this file set node IDs and run hourly sums.

```
begin: main wait

{DPIPE_HOME}/bin/ovpi_run_sql  -sqlscript  {DPIPE_HOME}/scripts/
{DBVENDOR}/execute_update_boot_time.sql -logfile {DPIPE_HOME}/tmp/
execute_update_boot_time.out

{DPIPE_HOME}/bin/ovpi_run_sql  -sqlscript  {DPIPE_HOME}/scripts/
{DBVENDOR}/execute_SR_OVPA_SetIDs.sql

{DPIPE_HOME}/bin/ovpi_run_sql  -sqlscript  {DPIPE_HOME}/scripts/
{DBVENDOR}/update_filesystem_property_table.sql

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_global.sum

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_volume.sum

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_filesystem.sum

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_tran.sum

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_application.sum

{DPIPE_HOME}/bin/ovpi_run_sql  -sqlscript  {DPIPE_HOME}/scripts/
{DBVENDOR}/execute_SR_Populate.sql -logfile {DPIPE_HOME}/tmp/
execute_SR_Populate.out

{DPIPE_HOME}/bin/trend_sum -f {DPIPE_HOME}/scripts/SR_SR_SysUp.sum
```

```
{DPIPE_HOME}/bin/ovpi_run_sql  -sqlscript  {DPIPE_HOME}/scripts/
{DBVENDOR}/Update_Tran_ProTables.sql

end:
```

RSR_OVPA_global to SR_SR_Kernel

Purpose: Map metrics from RSR_OVPA_global to the SR_SR_Kernel base table.

File: SR_SR_global.sum source table: RSR_OVPA_global destination table: SR_SR_Kernel by variable: node_fk by variable: ta_period		
Source Metric	**Destination Metric**	**Summary Type**
GBL_MEM_UTIL	TOTmemoryUtil	tot
case when (GBL_MEM_UTIL < 25) then 1 when (GBL_MEM_UTIL < 50) then 2 when (GBL_MEM_UTIL < 75) then 3 when (GBL_MEM_UTIL >= 75) then 4 end	TOTmemoryUtil_grade	tot
GBL_RUN_QUEUE	TOTrunq	tot
case when (GBL_RUN_QUEUE < 2) then 1 when (GBL_RUN_QUEUE < 3) then 2 when (GBL_RUN_QUEUE < 4) then 3 when (GBL_RUN_QUEUE >= 4) then 4 end	TOTrunq_grade	tot
GBL_CPU_TOTAL_UTIL	TOTcpuutil	tot
case when (GBL_CPU_TOTAL_UTIL < 25) then 1 when (GBL_CPU_TOTAL_UTIL < 50) then 2 when (GBL_CPU_TOTAL_UTIL < 75) then 3 when (GBL_CPU_TOTAL_UTIL >= 75) then 4 end	TOTcpuutil_grade	tot
GBL_ALIVE_PROC	TOTavgNumProcs	tot
GBL_MEM_PAGEOUT_RATE	AVGmemPageOutRate	avg
GBL_SWAP_SPACE_UTIL	TOTswapUtil	avg
case when (GBL_SWAP_SPACE_UTIL < 25) then 1 when (GBL_SWAP_SPACE_UTIL < 50) then 2 when (GBL_SWAP_SPACE_UTIL < 75) then 3 when (GBL_SWAP_SPACE_UTIL >= 75) then 4 end	TOTswapUtil_grade	tot

RSR_OVPA_filesystem to SR_SR_Filesystem

Purpose: Map metrics from RSR_OVPA_filesystem to the SR_SR_Filesystem base table.

File: SR_SR_filesystem.sum source table: RSR_OVPA_filesystem destination table: SR_SR_FileSystem by variable: node_fk by variable: FSIndexP by variable: ta_period		
Source Metric	**Destination Metric**	**Summary Type**
FS_MAX_SIZE	TOTfileSystemSize	tot
FS_SPACE_UTIL	TOTfileSystemUtil	tot

RSR_OVPA_global to SR_SR_SysVol

Purpose: Map metrics from RSR_OVPA_global to the SR_SR_SysVol base table.

File: SR_SR_volume.sum source table: RSR_OVPA_global destination table: SR_SR_SysVol by variable: node_fk by variable: ta_period		
Source Metric	**Destination Metric**	**Summary Type**
GBL_NET_IN_PACKET+GBL_NET_OUT_PACKET	TOTvolume	tot

RSR_OVPA_application to SRApp_application

Purpose: Map metrics from RSR_OVPA_application to the SRApp_application base table.

File: SR_SR_application.sum source table: RSR_OVPA_application destination table: SRApp_application by variable: node_fk by variable: prop_app_name by variable: ta_period		
Source Metric	**Destination Metric**	**Summary Type**
APP_CPU_TOTAL_UTIL	AVGCPUUtil	avg
APP_CPU_TOTAL_UTIL	MAXCPUUtil	max
APP_CPU_TOTAL_UTIL	MINCPUUtil	min
APP_CPU_TOTAL_TIME	TOTCPUSeconds	tot
APP_MEM_VIRT	TOTMemSeconds	tot
APP_ALIVE_PROC	TOTProcess	tot

RSR_OVPA_transaction to SRTran_Transaction

Purpose: Map metrics from RSR_OVPA_transaction to the SRTran_Transaction base table.

File: SR_SR_tran.sum source table: RSR_OVPA_transaction destination table: SRTran_Transaction by variable: node_fk by variable: Transaction_Name by variable: ta_period column: Completed=TT_COUNT:tot column: Aborted=TT_ABORT:tot column: ResponseTime=TT_WALL_TIME_PER_TRAN:avg,max,min column: SLOPercent=TT_SLO_PERCENT:avg column: Violation=TT_SLO_COUNT:tot		
Source Metric	**Destination Metric**	**Summary Type**
TT_WALL_TIME_PER_TRAN	AVGResponseTime	avg
TT_WALL_TIME_PER_TRAN	MAXResponseTime	max
TT_WALL_TIME_PER_TRAN	MINResponseTime	min
TT_COUNT	TOTCompleted	tot

File: SR_SR_tran.sum source table: RSR_OVPA_transaction destination table: SRTran_Transaction by variable: node_fk by variable: Transaction_Name by variable: ta_period column: Completed=TT_COUNT:tot column: Aborted=TT_ABORT:tot column: ResponseTime=TT_WALL_TIME_PER_TRAN:avg,max,min column: SLOPercent=TT_SLO_PERCENT:avg column: Violation=TT_SLO_COUNT:tot		
TT_ABORT	TOTAborted	tot
TT_SLO_PERCENT	AVGSLOPercent	avg
TT_SLO_COUNT	TOTViolation	tot

Index

T

U

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click on the bookmark “Comments”.

In case you do not have the email client configured, copy the information below to a web mail client, and send this email to **docfeedback@hp.com**

Product name:

Document title:

Version number:

Feedback:


invent


Java™
COMPATIBLE