

HP OpenView Performance Insight

Interface Reporting ifEntry Datapipe User Guide

Software Version: 2.1

Reporting and Network Solutions 7.0



November 2004

© Copyright 2004 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 2003–2004 Hewlett-Packard Development Company, L.P., all rights reserved.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

OpenView is a U.S. registered trademark of Hewlett-Packard Development Company, L.P.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Windows® and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

Support

Please visit the HP OpenView web site at:

<http://www.managementsoftware.hp.com/>

This site provides contact information and details about the products, services, and support that HP OpenView offers. To reach the Support site, click **Support**. The Support site provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by using the support site to:

- Search for documents of interest
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Join online discussions with other software customers
- Research and register for software training

Most of the support areas require you to register as an HP Passport user and log in. You may also require a support contract.

To find more information about access levels, go to the following URL:

http://support.openview.hp.com/access_level.jsp

To register for an HP Passport ID, go to:

<http://passport.hp.com/hpp2/newuser.do>

contents

- Chapter 1 Overview** 3
 - Version History 3
 - Sources for Additional Information 3

- Chapter 2 Package Installation** 5
 - Guidelines for a Smooth Install 5
 - Installing the Interface Reporting ifEntry Datapipe 6
 - Package Removal 7

- Chapter 3 Datapipe Configuration** 9
 - Configuration Tasks 9
 - Configuring a Polling Rule 10
 - Accessing Forms Bundled with Interface Reporting 11
 - Exporting and Importing Property Data 12
 - Setting the Polling Flag 12
 - Changing the Interface Identifier (Re-Indexing Support) 15
 - Configuring the Datapipe for Multiple Pollers 16

- Chapter 4 Groups and Policies** 19
 - Polling Groups 19
 - Rules-Based Polling Groups 20
 - Polling Policies 20
 - IR_map_p 21

- Chapter 5 List of Polled Objects** 23
 - Table 1: Objects in ifEntry 23
 - Table 2: Objects in MIB-II Extensions 26

- Index** 29

Overview

The Interface Reporting ifEntry Datapipe collects MIB-II ifEntry and ifxEntry data from SNMP manageable devices. In addition to collecting data, the Interface Reporting ifEntry Datapipe provides the following capabilities:

- Re-indexing support (derived from the Interface Discovery Datapipe)
- Directed-instance polling, also known as per-instance polling
- High capacity counters whenever values may exceed 32 bits
- Optional synchronization of polling configuration with Network Node Manager (NNM)

The Interface Reporting ifEntry Datapipe does not require configuration. The configuration options described in Chapter 3 are optional.

Version History

Version	RNS Version - Date	OVPI Version/Enhancements
1.1	RNS 4.0 - June 2003	OVPI 4.5 & 4.6; Sybase support
2.0	RNS 5 - April 2004	OVPI 5.0; Oracle support
2.0	RNS 6.0 - August 2004	No changes
2.1	RNS 7.0 - November 2004	Interface synchronization enhancements (OVPI and NNM) - Sybase Interface synchronization enhancements (OVPI and NNM) - Oracle Minor fixes to formulas that calculate discard, error, and utilization percentages

Sources for Additional Information

The following documents are related to this manual:

- *Interface Reporting ifEntry Datapipe Release Statement 2.1*

- *Interface Discovery Datapipe 2.1 User Guide*
- *Interface Reporting Report Pack 4.6 User Guide*
- *Frame Relay CPE Datapipe 4.0 User Guide*
- *RNS 7.0 Release Notes, November 2004*

Manuals for the core product, OVPI, as well as manuals for the reporting solutions that run on the core product can be downloaded from this web site:

<http://www.hp.com/managementsoftware>

Select **Support** > **Product Manuals** to open the **Product Manuals Search** page. The user guides for OVPI are listed under **Performance Insight**. The user guides for reporting solutions, NNM SPIs, and NNM correlators are listed under **Reporting and Network Solutions**.

The manuals listed under **Reporting and Network Solutions** indicate the month and year of publication. If a manual is revised and reposted, the date of publication will change even if the software version number does not change. Since we post updated user guides on a regular basis, you should search this site for updates before using an outdated manual.

Package Installation

This chapter covers the following topics:

- Guidelines for a smooth installation
- Using Package Manager to install the package
- Package removal

Guidelines for a Smooth Install

The RNS 7.0 CD contains components for NNM and report packs for OVPI. When you select OVPI packages for installation, the install script on the CD extracts every OVPI package from the CD and copies the results to the Packages directory on your system. When the extraction finishes, the install script prompts you to launch the Package Manager install wizard.

If you have already extracted OVPI packages on the RNS 7.0 CD, every package related to OVPI is already on your system and available for installation. Install the Interface Reporting ifEntry Datapipe by starting Package Manager and following the on-screen instructions. Before using Package Manager, review the following guidelines.

Software Prerequisites

The Interface Reporting ifEntry Datapipe 2.1 has the following prerequisites:

- OVPI 5.0
- All available service packs for OVPI 5.0
- Interface Discovery Datapipe 2.1

Service packs are available at the following site:

<http://openview.hp.com/>

You may install the Interface Discovery Datapipe first, or you may install both datapipes at the same time.

Distributed Environments

If your system is distributed, installation is more complex. Make sure that you install the Interface Reporting ifEntry Datapipe on every OVPI server that performs polling. Do not install the Interface Reporting ifEntry Datapipe on the central server unless the central server performs polling.

Upgrading from Version 1.1 or 2.0

There is no upgrade package for the Interface Reporting ifEntry Datapipe. If you are running version 1.1 or 2.0, your first step is to start Package Manager and remove your current version. Then restart Package Manager and install version 2.1.

Installing the Interface Reporting ifEntry Datapipe

Insert the RNS 7.0 CD and follow the on-screen instructions for extracting OVPI solution components. On Windows, the instructions appear in a Main Menu that opens automatically; on UNIX, you must log in as root, mount the CD (if the CD does not mount automatically), navigate to the top level of the CD directory, and then run the `./setup` command.

After the OVPI packages have been extracted and copied to your system, the install script launches OVPI and starts Package Manager. If you just extracted packages from the CD and the Package Manager install wizard is running, start this procedure at [Step 4](#); otherwise begin at [Step 1](#).

- 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/ovpi_timer stop**


Sun: **sh /etc/init.d/ovpi_timer stop**

- 3 From the Management Console, select **Tools > Package Manager**. The Package Manager welcome window opens.
- 4 Click **Next**. The Package Location window opens.
- 5 Click **Install**. Approve the default destination directory or browse to a different directory if necessary.
- 6 Click **Next**. The Report Deployment window opens. Type your OVPI username and password.
- 7 Click **Next**. The Package Selection window opens.
- 8 Click the check boxes for:


Interface_Reporting_ifEntry_Datapipe 2.1

IRifEntry_Sync_Interfaces

- 9 Click **Next**. The Type Discovery window opens. Clear the default.

 If you are also installing the Interface Discovery Datapipe now, refer to the *Interface Discovery Datapipe User Guide* for more information about the Type Discovery option.

- 10 Click **Next**. The Selection Summary window opens.
- 11 Click **Install**. The Installation Progress window opens and the install begins. When the install finishes, a package installation complete message appears.
- 12 Click **Done** to return to the Management Console.

 If you are also installing the Interface Discovery Datapipe, you may want to re-assign devices to the appropriate interface identifier index-by group before you restart OVPI Timer. This step is explained in the *Interface Discovery Datapipe User Guide*.

- 13 If required, re-assign devices to the appropriate interface identifier groups.
- 14 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**

The Interface Reporting ifEntry Datapipe is now installed on your system.

Package Removal

Follow these steps to uninstall the Interface Reporting ifEntry Datapipe.

- 1 Log in to the system. On UNIX systems, log in as root.
- 2 Stop OVPI Timer and wait for processes to terminate.
- 3 Start Package Manager. The Package Manager install wizard opens.
- 4 Follow the on-screen directions for package removal and select the Interface Reporting ifEntry Datapipe for removal. When the uninstall finishes, a package removal complete message appears.
- 5 Click **Done** to return to the Management Console.
- 6 Restart OVPI Timer.

Datapipe Configuration

This chapter covers the following topics:

- Configuration tasks and how to accomplish them
- Configuring a polling rule
- Accessing forms bundled with Interface Reporting 4.5
- Exporting and importing property data
- Setting the poll flag
- Changing the interface identifier
- Configuring the datapipe for multiple pollers

Configuration Tasks

The following table maps each configuration task to one or more methods for accomplishing the task.

Task	Means to Accomplish
Configure a polling rule	Configuration Tool
Set the data filtering mode	Administer Interface Reporting (form)
Set the provisioning mode	Administer Interface Reporting (form)
Set the message level	Administer Interface Reporting (form)
Export and import property data	IR Configuration Tool
Set the polling flag	<ol style="list-style-type: none"> 1 Administer Interface Poll Flag (form) 2 Polling Policy Manager 3 Property import for the IR Report Pack 4 Synchronize polling with NNM 5 Custom SQL script

Task	Means to Accomplish
Disable protocol monitoring	Change Protocol Defaults (form)
Change the interface identifier	<ol style="list-style-type: none"> 1 Interface Discovery Datapipe 2 Polling Policy Manager
Configure a datapipe for multiple pollers	<ol style="list-style-type: none"> 1 OVPI group_manager 2 OVPI collection_manager 3 Polling Policy Manager

Configuring a Polling Rule

Polling is conducted in accordance with one of the following five rules:

(1) Poll All Active Interfaces

Poll all interfaces where the ifAdminStatus is set to Active.

(2) Poll All Interfaces

Poll all interfaces, regardless of ifAdminStatus or protocol. (This is the default rule when the Interface Reporting ifEntry Datapipe is installed.)

(3) Poll All Active Interfaces with Monitored Protocols

Default. Poll all interfaces where the ifAdminStatus is set to Active and the protocol (ifType) is configured for monitoring in the KIR_protocol_config table. You can disable monitoring on protocols that do not require monitoring.



The default is Poll All Active Interfaces with Monitored Protocols.

(4) Poll All Interfaces with Monitored Protocols

Poll all interfaces where the protocol (ifType) is configured for monitoring in the KIR_protocol_config table. You can disable monitoring on protocols that do not require monitoring.

(5) Poll Flagged Interfaces Only

Poll only interfaces that have the polling flag set. This setting allows individual interfaces to be provisioned for collection. Use this setting to control polling for integration with the “unmanaged interfaces” feature of NNM.

Follow these steps to configure a polling rule:

- 1 Log in to the system. On UNIX, log in as root.
- 2 Start the IR Configuration Tool from the command line:

UNIX:

```
$DPIPE_HOME/bin/perl $DPIPE_HOME/scripts/IR_Config.pl
```

Windows:

```
%DPIPE_HOME%\bin\perl %DPIPE_HOME%\scripts\IR_Config.pl
```

The IR Configuration Tool main menu opens.

- 3 Select option 1, Configure Polling, and press **Enter**.

The Configure Polling menu opens.

- 4 Select option 1, Set Polling Rule, and press **Enter**.

The Set Polling Rule menu opens, displaying the current polling rule and the date and time the rule was last changed.

- 5 Select the option for the required polling rule and press **Enter**.

After displaying messages as it rebuilds the database views that implement the polling rules, the Set Polling Rule menu opens again, showing the new polling rule.

- 6 Select **E** and press **Enter** to return to the previous menu.

The Configure Polling menu opens.

- 7 Select **E** and press **Enter** to return to the previous menu.

The IRCT main menu opens.

- 8 Select an option or select **E** and press **Enter** to exit.



If you apply the “Poll Flagged Interfaces Only” rule, you must also set the polling flag for the interfaces you want to poll.

Accessing Forms Bundled with Interface Reporting

Follow these steps to access the forms bundled with Interface Reporting 4.5:

- 1 Start the Management Console.
- 2 Click the **Objects** icon. The Object/Property Management window opens.
- 3 Select a device.

The following forms are listed under **Object Specific Tasks**:

- Change Protocol Defaults



By default, all protocols (ifTypes) are monitored, which means that data will be collected for all protocols. If you are not interested in certain protocols, use the Change Protocol Defaults form to disable collection and reporting.

- Change Interface
- Change Interface Customer
- Change Interface Description
- Administer Interface Poll Flag

The following form is listed under **General Tasks**:

- Administer Interface Reporting

For more information about forms, see Chapter 6, Using Change Forms, in the *Interface Reporting Report Pack 4.5 User Guide*.

Exporting and Importing Property Data

The IR Configuration Tool makes it easy for you to export existing property data from the Interface Reporting Report Pack and import new property data into the Interface Reporting Report Pack. When you use the IR Configuration Tool to export or import property data, the data file will be exported to, and imported from, the default directory for property data owned by the Interface Reporting Report Pack, which is:

```
$DPIPE_HOME/data/PropertyData
```

Follow these steps to export or import property data:

- 1 Log in to the system. On UNIX, log in as root.
- 2 Start the IR Configuration Tool from the command line:

UNIX:

```
$DPIPE_HOME/bin/perl $DPIPE_HOME/scripts/IR_Config.pl
```

Windows:

```
%DPIPE_HOME%\bin\perl %DPIPE_HOME%\scripts\IR_Config.pl
```

The IRCT main menu opens.

- 3 Select option 2, Export or Import Interface Reporting Property Data, and press **Enter**.
The Property Export and Import menu opens.
- 4 Do one of the following:
Select **1. Export Interface Reporting Property Data** to export.
Select **2. Import Interface Reporting Property Data** to import.
- 5 When finished, select **E** and press **Enter** to return to the previous menu.
The IR Configuration Tool main menu opens.
- 6 Select an option, or select **E** and press **Enter** to exit.

Setting the Polling Flag

To set the polling flag, use any of the following methods:

- Administer Interface Poll Flag (a form)
- Polling Policy Manager
- Property import for the Interface Reporting Report Pack

- Synchronize with the “unmanaged interfaces” polling configuration on NNM
 - ▶ This method of setting the polling flag applies to OVPI on Sybase only.
- Create a custom SQL script

Using Polling Policy Manager to Set the Polling Flag

When the Poll Flagged Interfaces Only rule is enabled, the Interface Reporting ifEntry Datapipe collects statistics only from interfaces that have a polling flag set in the main interface property table, K_IFEntry_Disc.poll_flag_ifEntry. Follow these steps to modify the polling flag using Polling Policy Manager:

- 1 Determine which interfaces on which devices will be polled and which interfaces on which devices will not be polled.
- 2 Log in to the system. On UNIX, log in as root.
- 3 Start Polling Policy Manager.
- 4 Select **Edit > Descriptions...** The Edit Descriptions window opens.
- 5 Select the Property Group K_IFEntry_Disc and click **Rule: Edit...** The Create Rule window opens.
- 6 Select or enter a rule; for example, to return higher speed interfaces only, use `interface_speed >= 100000000`, and click **OK**.
- 7 Click **View Resulting Instances**. The instances will appear in the lower pane.
- 8 From the returned list, select all the interfaces that are to be polled and click **Edit Instance Values...** The Edit Instance window opens.
- 9 Select the Column `poll_flag_ifEntry`, enter a value of 1, and click **Apply**; then click **Close** to return to the Edit Descriptions window.
- 10 Click **View Resulting Instances**. The updated instances appear in the lower pane.
- 11 Click **Close**.
- 12 If the Poll Flagged Interfaces Only rule is not already enabled, enable it now.

Using Property Import to Set the Polling Flag

When the Poll Flagged Interfaces Only rule is enabled, the Interface Reporting ifEntry Datapipe collects statistics only from interfaces that have a polling flag set in the main interface property table. Follow these steps to set the polling flag using the Interface Reporting Property Import:

- ▶ If Data Filtering mode is enabled, so that only data for provisioned interfaces is collected, the polling flag is set automatically for each interface included in the property import file.

- 1 Determine which interfaces on which devices will be polled, and which interfaces on which devices will not be polled.

- 2 Log in to the system. On UNIX, log in as root.
- 3 Generate a new Interface Reporting property import file with the interfaces that are to be polled, or, if the interfaces have already been discovered, export the property file and edit it to include only the desired interfaces.
- 4 Use the Administer Interface Reporting form to check whether data filtering mode is on or off. If data filtering mode is not already enabled, enable it now.
- 5 Import the Interface Reporting property data you produced in [Step 3](#).
- 6 If the Poll Flagged Interfaces Only rule is not already enabled, enable it now.

Set the Polling Flag by Synchronizing with NNM



This method of setting the polling flag applies to OVPI on Sybase only.

When the Poll Flagged Interfaces Only rule is enabled, the Interface Reporting ifEntry Datapipe will collect statistics only from interfaces that have a polling flag set in the main interface property table. This flag can be set by synchronizing with the “unmanaged interfaces” polling configuration on NNM. NNM allows the user to define unmanaged interfaces two ways, through the user interface, and by using a special add-on, rules-based tool designed for this purpose.

The IRifEntry Sync Interfaces package is designed to upload the managed and unmanaged interfaces list from NNM, set the polling flag for managed interfaces, and clear the flag for unmanaged interfaces. This package depends on the Device Sync package in the Integration Module. For details about installing both packages, refer to the *NNM/Performance Insight Integration Module 2.0 User Guide*.

When you install the IRifEntry Sync Interfaces package, these events take place:

- A complete list of managed and unmanaged interfaces is downloaded from NNM.
- The interface polling flag is set (based on the NNM configuration).
- OVPI Timer entries are configured for:
 - A daily synchronization of all interfaces that have undergone a change in management status during the previous 24 hours
 - A full re-synchronization once a month

Follow these steps to set the polling flag by synchronizing with NNM:

- 1 If the packages bundled with the Integration Module are not already installed on NNM and OVPI, install them.
- 2 If the IRifEntry Sync Interfaces package is not already installed, install it now. Follow the procedure provided in [Chapter 2, Package Installation](#).
- 3 If the Poll Flagged Interfaces Only rule is not already enabled, enable it now.

Creating a Custom SQL script

If polling configuration is held in a provisioning or configuration management system, develop a custom SQL script to set the poll flag (K_IFEntry_Disc.poll_flag_ifEntry).

Changing the Interface Identifier (Re-Indexing Support)

The Interface Discovery Datapipe provides the following alternate identifiers:

- ifDescr—supported by the index-by-ifDescr polling group
- ifName—supported by the index-by-ifName polling group
- ifAlias—supported by the index-by-ifAlias polling group

To enable re-indexing, select an alternate identifier (just *one* alternate identifier) that is unique for the device and persistent.



Do **not** assign a device to more than one group. Only one alternate identifier per device is allowed.

The Interface Discovery Datapipe includes vendor-specific OVPI Type Discovery files that automatically assign certain types of devices to an alternate interface identifier, so some of the selection work is already done for you. The following table contains a list of device types and pre-assigned identifiers.

Device Type	Identifier Assigned by OVPI Type Discovery
Cisco Catalyst switches	ifName
Cisco routers	ifDescr
HP ProCurve switches (except 800T)	ifName

The index-by-ifIndex group is the default group. This group contains all the devices in the GENMIB2IF_Datapipe group that are not in the index-by-ifAlias, index-by-ifDescr, or index-by-ifName groups.

Follow these steps to change an interface identifier:

- 1 For each device type, choose the interface identifier.
- 2 Assign each device of that type to the appropriate index-by group.
 - a Start Polling Policy Manager.
 - b Select **Edit > Polling Groups...**
 - c Select **All Nodes of the Same Type** in the top pane and the required index-by group in the bottom pane.
 - d Click **Edit**.
 - e Select the nodes to be assigned to this group in the left-hand pane, and click >> to move them into the group. Click **OK** when complete.
- 3 Repeat the process above for other device types and index-by groups as necessary.

If a device is assigned to an index-by group, but the identifiers are not unique, a duplicate identifier message, similar to the following message, will be inserted in the trendlog.

```
bcp_gateway, , WARNING, 1984, 2076, 0, "the object [testhost.test.hp.com_AT3/0] is duplicate for data table rv_ifentry_disc_ifname"
```

Configuring the Datapipe for Multiple Pollers

The Interface Reporting ifEntry Datapipe requires additional configuration for use with multiple pollers. The standard installation configures four groups and four polling policies for a single poller. For use with multiple pollers, additional sets of groups and polling policies must be configured for each additional poller.

The datapipe package includes additional group and collection policy files for two pollers. If you need to configure the datapipe for more than two pollers against a single database, follow these steps to generate additional sets of group files and policies:

- 1 Log in as the OVPI Administrator (trendadm).
- 2 Navigate to the Interface Reporting ifEntry Datapipe package directory:
- 3 Run the command:

```
trend_proc -f create_groups_multi_pollers.pro
```

- 4 Run the command:

```
collection_manager -import -file IRifEntry_Collection_Policies_multi_pollers
```

- 5 Start Polling Policy Manager and select **Edit > Datapipe Installations...**
- 6 Click **Create...** to create new pollers.
- 7 Add the hostname and (optionally) the IP address of each poller and click **OK**.
- 8 When all pollers have been added, click **Close** to exit the Edit Datapipe Installations window.
- 9 Click **All** in the left panel to display all polling policies.
- 10 Delete the original if*_Collection policies (those without a _poller1 or _poller2 at the end of the name).
- 11 Modify the new if*_Collection_poller1 policies by double-clicking the policy.
Change the **Polling Assigned to** setting to be the hostname of poller1.
- 12 Modify the new if*_Collection_poller2 policies by double-clicking the policy.
Change the **Polling Assigned to** setting to be the hostname of poller2.

At this point, the 15-minute policies should look similar to those in the following table.

Policy Name	Table Alias Name for Data to be Collected	Group Name to Collect From	Poll From
ifAlias_Collection_poller1	xV_IRifEntry_ifAlias	ifAlias-poller1	Hostname of poller1
ifDescr_Collection_poller1	xV_IRifEntry_ifDescr	ifDescr-poller1	Hostname of poller1
ifIndex_Collection_poller1	xV_IRifEntry_ifIndex	ifIndex-poller1	Hostname of poller1
ifName_Collection_poller1	xV_IRifEntry_ifName	ifName-poller1	Hostname of poller1
ifAlias_Collection_poller2	xV_IRifEntry_ifAlias	ifAlias-poller2	Hostname of poller2

Policy Name	Table Alias Name for Data to be Collected	Group Name to Collect From	Poll From
ifDescr_Collection_poller2	xV_IRifEntry_ifDescr	ifDescr-poller2	Hostname of poller2
ifIndex_Collection_poller2	xV_IRifEntry_ifIndex	ifIndex-poller2	Hostname of poller2
ifName_Collection_poller2	xV_IRifEntry_ifName	ifName-poller2	Hostname of poller2

- 13 Select **Edit > Polling Groups... > All Nodes in Same View** in the upper pane, then select poller1 in the lower pane. Assign nodes to be collected by poller1 to the poller1 view.
- 14 Select **Edit > Polling Groups... > All Nodes in Same View** in the upper pane, then select poller2 in the lower pane. Assign nodes to be collected by poller2 to the poller2 view.

Groups and Policies

This chapter covers the following topics:

- Polling groups
- Rules-based polling groups
- Polling policies
- The IR_map_p stored procedure

Polling Groups

The Interface Reporting ifEntry Datapipe uses the same OVPI Polling “Type” Groups that the Interface Discovery Datapipe uses. Polling groups make re-indexing support possible. They are created when the Interface Discovery Datapipe is installed.

Polling Group	Description
GENMIB2IF_Datapipe	Top-level group for all nodes that support MIB-II.
index-by-ifAlias	Subset of the GENMIB2IF_Datapipe group; contains all nodes that should use ifAlias as the identifier.
index-by-ifDescr	Subset of the GENMIB2IF_Datapipe group; contains all nodes that should use ifDescr as the identifier.
index-by-ifIndex	Subset of the GENMIB2IF_Datapipe group; contains all the devices that do not belong to the index-by-ifAlias, index-by-ifDescr, or index-by-ifName groups. Unless a device is assigned to a different group, it will default to ifIndex as the interface identifier. Editing this group is not necessary.
index-by-ifName	Subset of the GENMIB2IF_Datapipe group; contains all nodes that should use ifName as the identifier.

The index-by-ifIndex group is the default. This group does not control polling; rather, it shows you which devices will use ifIndex as the interface identifier. Since this group is automatically repopulated by the system once per hour, any device that is re-assigned to a different index-by-

group may not immediately disappear from index-by-ifIndex. However, since the data collection for index-by-ifIndex is based on devices in the GENMIB2IF_Datapipe group, the device will be polled only from its new index-by group.

Rules-Based Polling Groups

In addition to the five basic polling groups, the Interface Reporting ifEntry Datapipe adds four groups that are derived from the basic groups but also contain rules for rules-based polling.

Group Name	Database View	Description
ifAlias-all	KV_IFEntry_by_ifAlias	Derived from the index-by-ifAlias group with polling rules applied.
ifDescr-all	KV_IFEntry_by_ifDescr	Derived from the index-by-ifDescr group with polling rules applied.
ifIndex-all	KV_IFEntry_by_ifIndex	Derived from the devices in the GENMIB2IF_Datapipe group that are not in index-by-ifAlias, index-by-ifDescr, or index-by-ifName, with polling rules applied.
ifName-all	KV_IFEntry_by_ifName	Derived from the index-by-ifName group with polling rules applied.

To see the list of interfaces that will be polled in each group, use SQL to select from the database views in the table above.

Polling Policies

A standard installation of the Interface Reporting ifEntry Datapipe enables a set of four 15-minute polling policies, one for each index-by group. Changing these policies is not necessary, unless you need to enable support for multiple pollers.

Policy Name	Table Alias Name for Data to be Collected	Group Name to Collect From
ifAlias_Collection	xV_IRifEntry_ifAlias	ifAlias-all
ifDescr_Collection	xV_IRifEntry_ifDescr	ifDescr-all
ifIndex_Collection	xV_IRifEntry_ifIndex	ifIndex-all
ifName_Collection	xV_IRifEntry_ifName	ifName-all

The Interface Reporting ifEntry Datapipe adds a flag column called `poll_flag_ifEntry` to the Interface Discovery Datapipe property table `K_IFEntry_Disc`. Provisioning this flag will control directed-instance polling.

IR_map_p

The Interface Reporting ifEntry Datapipe uses a stored procedure called IR_map_p to transfer collected data to the Interface Reporting Report Pack. IR_map_p is called at the end of the raw-to-rate conversion. IR_map_p and the conversion procedure run at the end of each of the four polling policies above. While they are running, these procedures lock tables to ensure that no data is lost. As a result, it is not unusual for one or more instances of the raw-to-rate stored procedure to generate lock errors indicating that an instance is already running. Here is an example of the error message:

```
"Msg 20003, Level 16, State 1:"  
"Server 'MYOVPI_SYBASE', Procedure 'dsi_locks_p', Line 267:"  
"Unable to lock the object 'x_IRifEntryTOR_IRifEntry28_p'."  
"Error: raw2rate procedure failed for table xV_IRifEntry_ifindex, database  
MYOVPI_SYBASE. Exit code=1"
```

IR_map_p also generates standard information messages with the number of rows transferred to the Interface Reporting Report Pack. For example:

```
IR_map_p: Info: 20420 polled rows were mapped to SRIRdevPorts in 18 seconds.
```


List of Polled Objects

The following tables provide the name, identifier, and description of polled objects. Table 1 (below) provides a list of objects, organized by object ID, in the ifEntry MIB table. Table 2 (page 26) provides a list of objects, organized by object ID, in the MIB-II Extensions table, also known as the ifXTable.

Table 1: Objects in ifEntry

MIB Object Name	OID	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	A unique value for each interface, ranging between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.
ifDescr	1.3.6.1.2.1.2.2.1.2	A text string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.
ifType	1.3.6.1.2.1.2.2.1.3	The type of interface, distinguished according to the physical/link protocol(s) immediately below the network layer in the protocol stack.
ifMtu	1.3.6.1.2.1.2.2.1.4	The size of the largest datagram that can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.
ifSpeed	1.3.6.1.2.1.2.2.1.5	An estimate of the interface's current bandwidth in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.

Table 1: Objects in ifEntry

MIB Object Name	OID	Description
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	The interface's address at the protocol layer immediately below the network layer in the protocol stack. For interfaces that do not have such an address (e.g., a serial line), this object should contain an octet string of zero length.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	The desired state of the interface. In the testing state, no operational packets can be passed. States are up(1), down(2), testing(3).
ifOperStatus	1.3.6.1.2.1.2.2.1.8	The current operational state of the interface. In the testing state, no operational packets can be passed. States are up(1), down(2), testing(3).
ifLastChange	1.3.6.1.2.1.2.2.1.9	The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered before the last re-initialization of the local network management subsystem, this object contains a zero value.
IfInOctets	1.3.6.1.2.1.2.2.1.10	The total number of octets received on the interface, including framing characters.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
ifInErrors	1.3.6.1.2.1.2.2.1.14	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifUnknownProtos	1.3.6.1.2.1.2.2.1.15	The number of packets received via the interface that were discarded because of an unknown or unsupported protocol.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	The total number of octets transmitted out of the interface, including framing characters.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.

MIB Object Name	OID	Description
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	The number of outbound packets that could not be transmitted because of errors.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	The length of the output packet queue (in packets).
ifSpecific	1.3.6.1.2.1.2.2.1.22	A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an ethernet, the value of this object refers to a document defining objects specific to ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conforming implementation of ASN.1 and BER must be able to generate and recognize this value.

Table 2: Objects in MIB-II Extensions

MIB Object Name	OID	Description
ifName	1.3.6.1.2.1.31.1.1.1.1	<p>The text name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's console. This might be a text name, such as "le0", or a simple port number, such as 1, depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, each will have the same value of ifName.</p> <p>If there is no local name, or if this object is otherwise not applicable, this object contains a 0-length string.</p>
ifAlias	1.3.6.1.2.1.31.1.1.1.18	<p>This object is an alias name for the interface as specified by a network manager, and provides a nonvolatile "handle" for the interface.</p> <p>On the first instantiation of an interface, the value of ifAlias associated with that interface is the zero-length string. When a value is written into an instance of ifAlias through a network management set operation, the agent must retain the supplied value in the ifAlias instance associated with the same interface for as long as that interface remains instantiated, even if the network management system reinitializes/reboots.</p> <p>An example of the value that a network manager might store in this object for a WAN interface is the (Telco's) circuit number/identifier of the interface.</p> <p>Some agents may support write-access only for interfaces having particular values of ifType. An agent that supports write access to this object is required to keep the value in non-volatile storage, but it may limit the length of new values, depending on how much storage is already occupied by the current values for other interfaces.</p>
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	<p>The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets.</p>

MIB Object Name	OID	Description
ifHCInUcastPkts	1.3.6.1.2.1.31.1.1.1.7	The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were not addressed to a multicast or broadcast address at this sub-layer. This object is a 64-bit version of ifInUcastPkts.
ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8	The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts.
ifHCInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.9	The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were addressed to a broadcast address at this sub-layer. This object is a 64-bit version of ifInBroadcastPkts.
ifHCOctets	1.3.6.1.2.1.31.1.1.1.10	The total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets.
ifHCOUcastPkts	1.3.6.1.2.1.31.1.1.1.11	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts.
ifHCOMulticastPkts	1.3.6.1.2.1.31.1.1.1.12	The total number of packets that higher-level protocols requested be transmitted, and that were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutMulticastPkts.
ifHCOBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13	The total number of packets that higher-level protocols requested be transmitted, and that were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutBroadcastPkts.

A

alternate identifiers, **15**

C

Cisco Catalyst switches, alternate identifier, **15**

Cisco routers, alternate identifier, **15**

configuration

multiple pollers, **16**

polling rules, **10**

D

datapipe

installing, **6**

removing, **7**

directed-instance polling, **20**

distributed environment installation, **6**

G

GENMIB2IF_Datapipe polling group, **19**

H

HP ProCurve switches, alternate identifier, **15**

I

identifiers, alternate, **15**

ifAdminStatus object description, **24**

ifAlias object description, **26**

ifDescr object description, **23**

ifEntry MIB table, objects in, **23**

ifHCInBroadcastPkts object description, **27**

ifHCInMulticastPkts object description, **27**

ifHCInOctets object description, **26**

ifHCInUcastPkts object description, **27**

ifHCOutBroadcastPkts object description, **27**

ifHCOutMulticastPkts object description, **27**

ifHCOutOctets object description, **27**

ifHCOutUcastPkts object description, **27**

ifIndex object description, **23**

ifInDiscards object description, **24**

ifInErrors object description, **24**

ifInNUcastPkts object description, **24**

IfInOctets object description, **24**

ifInUcastPkts object description, **24**

ifLastChange object description, **24**

ifMtu object description, **23**

ifName object description, **26**

ifOperStatus object description, **24**

ifOutDiscards object description, **25**

ifOutErrors object description, **25**

ifOutNUcastPkts object description, **25**

ifOutOctets object description, **24**

ifOutQLen object description, **25**

ifOutUcastPkts object description, **24**

ifPhysAddress object description, **24**

ifSpecific object description, **25**

ifSpeed object description, **23**

ifType object description, **23**

ifUnknownProtos object description, **24**

index-by-ifAlias polling group, **19**

index-by-ifDescr polling group, **19**

index-by-ifIndex polling group, **19**

index-by-ifName polling group, **19**

installation

distributed environment, **6**

instructions, **6**

interface identifier, changing, **15**

Interface Reporting Property Import
setting polling flag, **13**

IR_map_p stored procedure, **21**
IRifEntry Sync Interfaces package, **14**

K

K_IFEntry_Disc property table, **20**

M

manuals, list of, **3**
MIB-II extensions table, objects in, **26**
MIB objects, lists of, **23**
multiple pollers, **16**

N

Network Node Manager (NNM)
 setting polling flag, **14**

O

objects
 in ifEntry MIB table, **23**
 in MIB-II extensions table, **26**
 polled, **23**
OVPI Timer, **7**

P

poll_flag_ifEntry, **20**
polled objects, list of, **23**
pollers, multiple, **16**
polling flag, **12**
 setting with Polling Policy Manager, **13**
 setting with property import, **13**
 synchronizing with polling configuration on
 NNM, **14**
polling groups
 default, **19**
 described, **19**
 rules-based, **20**
polling policies, standard, **20**
Polling Policy Manager
 creating new pollers, **16**
 setting polling flag, **13**
polling rules
 configuring, **10**
 described, **10**
prerequisites, software, **5**
ProCurve switches, alternate identifier, **15**

product documentation, **3**
Product Manuals Search page, **4**
property import, setting polling flag, **13**

R

re-indexing support, **15**
required software, **5**
rules-based polling groups, **20**

S

SQL script, for setting polling flag, **14**
synchronizing with unmanaged interfaces polling
 configuration, **14**

T

Type Discover files, **15**

U

uninstalling the datapipe, **7**
unmanaged interfaces polling configuration, **14**
upgrading the datapipe, **6**

V

vendor-specific OVPI Type Discover files, **15**