Oracle® Communications Network Integrity

CLI Cartridge Guide

Release 7.5
G13611-01
December 2024

ORACLE

Oracle Communications Network Integrity CLI Cartridge Guide, Release 7.5

G13611-01

Contents

3 About the VPLS Reference Cartridge

Preface

This guide explains the functionality and design of the Oracle Communications Network Integrity CLI cartridge.

Audience

This guide is intended for Network Integrity administrators, developers, and integrators.

This guide assumes that you are familiar with the following documents:

- *Network Integrity Developer's Guide*: for basic understanding of cartridges
- *Network Integrity Installation Guide*: for information about deploying and undeploying cartridges

This guide assumes that you are familiar with the following concepts:

- Telnet and Secure Shell (SSH) protocols
- Oracle Communications Design Studio
- Oracle Communications Information Model

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1
Overview

This chapter describes the Oracle Communications Network Integrity Command Line Interface (CLI) cartridge.

CLI Cartridge Overview

The CLI cartridge enables you to build deployable cartridges that connect to devices and retrieve information by using CLI commands over Telnet or Secure Shell (SSH) protocol.

The CLI cartridge provides the following key features:

- Telnet protocol and SSH communication with CLI devices
- Record and playback of CLI communication

The CLI cartridge is an abstract cartridge, meaning that Oracle Communications Design Studio is used to configure and assemble the run time cartridge for devices before deploying it into Network Integrity. See "About the CLI Cartridge" for more information about the CLI cartridge and its components.

Because there are no globally common CLI commands among all CLI devices, you must build a specific extension to the CLI cartridge so it can communicate with a specific device. See *Network Integrity Concepts* for guidelines and best practices for extending cartridges.

The CLI cartridge ZIP file contains a reference implementation cartridge for discovering Cisco devices running the IOS XR operating system and retrieving information about virtual private LAN services (VPLSs) on the Cisco IOS XR devices. See "VPLS Reference Cartridge Overview" for more information.

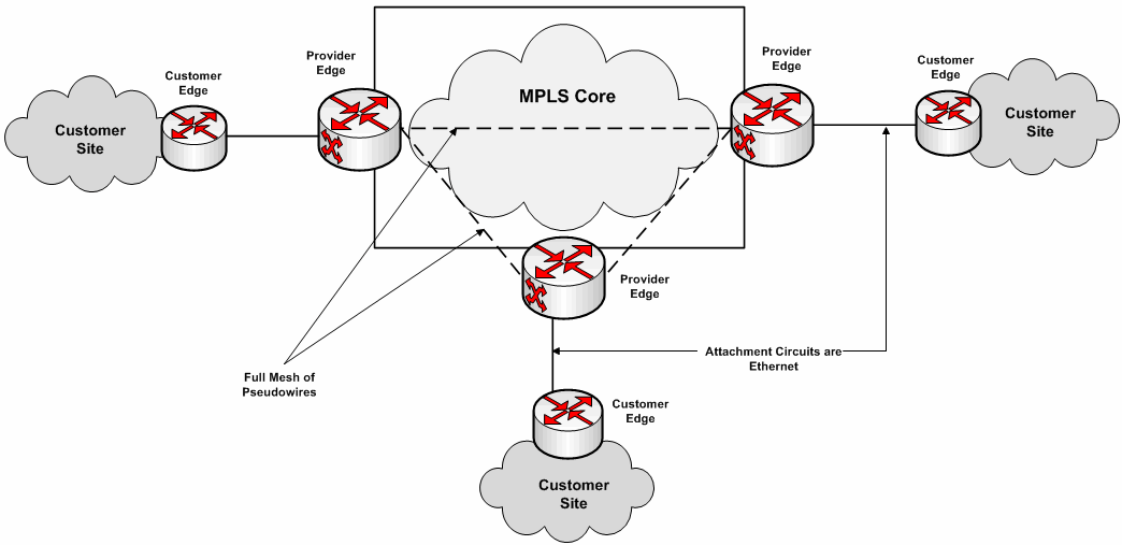
VPLS Reference Cartridge Overview

Layer 2 VPN services are provisioned over an IP network that typically uses multi-protocol label switching (MPLS) to rapidly switch data packets supported by various Layer 2 technologies, including Ethernet, over the network.

Virtual leased line (VLL) and virtual private LAN service (VPLS) are mainly used to provide Layer 2 VPN services. The VPLS reference cartridge supports VPLS Layer 2 VPN services.

A VPLS provides Layer 2 Ethernet connectivity across multiple customer sites in a manner that is transparent to the customer edge (CE) devices. The service provider takes care of transporting the Layer 2 frames and switching them across the provider network from one customer site to another. Service providers typically offer VPLS over a shared MPLS-based IP network. A VPLS is deployed as a full mesh of pseudowires between the provider edge (PE) routers. Figure 1-1 illustrates the VPLS reference model.

Figure 1-1 VPLS Reference Model



A VPLS network includes the following components:

- **Customer edge (CE) router**: The routers connecting individual customer sites to the service provider network.
- **Provider edge (PE) router**: The service provider devices to which the CE routers are directly connected.
- **Core MPLS network**: Interconnects the PEs where traffic is switched based on the MPLS labels.
- **Attachment circuit (AC)**: The physical or virtual circuit attaching a CE to a PE.
- **Pseudowire (PW)**: A term used to indicate an end-to-end path in a service provider network.

Using CLI commands, the VPLS reference cartridge does the following:

- Discovers Cisco devices running the IOS XR operating system
- Retrieves information about VPLS services on the Cisco IOS XR devices
- Models logical device hierarchies that represent the discovered Cisco IOS XR devices
- Models the VPLS service hierarchy

The VPLS reference cartridge supports discovery of both Border Gateway Protocol (BGP) and Label Distribution Protocol (LDP) signaling for a VPLS service, including BGP auto-discovery and manual discovery of devices in VPLS services. The VPLS reference cartridge does not support the discovery of hierarchical VPLS (HVPLS) services.

The VPLS reference cartridge provides no integration with other products but may be extended. This cartridge is designed to discover only Cisco devices running IOS XR.

See "About the VPLS Reference Cartridge" for more information about the VPLS reference cartridge and its components.

About the Cartridge Dependencies

This section provides information about dependencies that the CLI cartridge and VPLS reference cartridge have on other entities.

Run-Time Dependencies

The CLI cartridge and VPLS reference cartridge require that the Address_Handlers cartridge be deployed to Network Integrity.

Design-Time Dependencies

To load the CLI cartridge into Design Studio, the following cartridge projects must be installed:

- Address_Handlers
- NetworkIntegritySDK
- ora_uim_model
- ora_ni_uim_device

The load the VPLS reference cartridge into Design Studio, the following cartridge projects must be installed:

- Address_Handlers
- NetworkIntegritySDK
- ora_uim_model
- ora_ni_uim_device
- Abstract CLI cartridge

Downloading and Opening the Cartridge Files in Design Studio

To review and extend the CLI cartridge and the VPLS reference cartridge, download the VPLS Cartridge ZIP file from the Oracle software delivery website:

https://edelivery.oracle.com/

The VPLS Cartridge ZIP file has the following structure:

- **\Network_Integrity_Cartridge_Projects\Abstract_CLI_Cartridge**
- **\Network_Integrity_Cartridge_Projects\Cisco_IOS_XR_Model**
- **\Network_Integrity_Cartridge_Projects\VPLS_Cartridge**

The **Abstract_CLI_Cartridge** project and the **VPLS_Cartridge** project contain the extendable Design Studio files.

You must open the files in Design Studio before you can review and extend the cartridges.

See "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide* for information about opening files in Design Studio.

Building and Deploying the Cartridge

See "Getting Started with Design Studio for Network Integrity (1)" in *Design Studio Help* for information about building and deploying cartridges.

2

About the CLI Cartridge

This chapter describes the components of the Oracle Communications Network Integrity Command Line Interface (CLI) cartridge.

The CLI cartridge supports a recording mode for recording CLI data. See "About Record and Playback" for more information.

About Actions and Processors

The CLI cartridge contains the following actions:

- Discover Abstract Base CLI
- Discover Abstract CLI

The Discover Abstract CLI action extends the Discover Abstract Base CLI action and inherits all its processors to establish and manage the CLI connection.

The Discover Abstract Base CLI action contains the following processors run in the following order:

1. CLI Property Initializer
2. CLI Connection Manager

Figure 2-1 illustrates the processor workflow of the Discover Abstract Base CLI action.

Figure 2-1 Discover Abstract Base CLI Action Processor Workflow

The Discover Abstract CLI action adds scan parameter groups to the Discover Abstract Base CLI action and passes the property values to the CLIProperties object.

The Discover Abstract CLI action contains the following processors run in the following order:

1. CLI Property Initializer (inherited)
2. CLI Property Customizer
3. CLI Connection Manager (inherited)

Figure 2-2 illustrates the processor workflow of the Discover Abstract CLI action.

Figure 2-2 Discover Abstract CLI Action Processor Workflow

CLI Property Initializer

This processor is part of the Discover Abstract Base CLI action.

This processor initializes all the properties required for connecting to CLI devices and puts all the properties into a CLIProperties object (a Java class).

Table 2-1 lists the values that are initialized by the CLI Property Initializer processor.

Table 2-1 CLI Properties Initialized by the CLI Property Initializer Processor

Parameter	Description
host	The host name or IP address of the CLI device.
port	The port number of the CLI device.
loginName	The user name for logging in to the CLI device. It may be from 1 to 10 characters in length. Input is case-sensitive and can consist of numbers and upper- and lower-case alphabetic letters.
password	The user password or private identifier. It may be up to 10 characters in length.

Table 2-1 (Cont.) CLI Properties Initialized by the CLI Property Initializer Processor

Parameter	Description
prompt	The prompt on the CLI device.
telnetLoginPrompt	The Telnet prompt for the login name on the CLI device.
telentPasswordPrompt	The Telnet prompt for the password on the CLI device.
continueResponsePrompt	The continue response prompt for long responses on the CLI device.
knownHostFileLocation	The location where the public keys of the CLI device are stored if SSH is used. Leave blank to accept all public keys. If a directory is provided, the Network Integrity server must have permission to write to it; otherwise, the CLI cartridge defaults to accepting all public keys.
timeout	The timeout in seconds of the underlying socket connection.
retries	The number of retries that are attempted after a timeout. Obtained from the CLI scan parameter groups.
transportProtocol	The connection protocol to use: Telnet or SSH. The default is SSH.
mode	(Optional) The mode of the Record and Playback feature. Valid values are Normal (no record or playback), Record (recording mode enabled), and Playback (playback mode enabled). Specifying this parameter is optional.
recordFileDir	(Optional) The directory where the cartridge writes and retrieves recorded CLI data files. Specifying this parameter is optional.
deviceType	The type of the device.
softwareVersion	The software version of the device.
service	The type of service to be discovered on the device.

CLI Property Customizer

This processor is part of the Discover Abstract CLI action.

This processor populates the CLIProperties object produced by the CLI Property Initializer processor with the parameter values configured in the Network Integrity UI.

Table 2-1 lists all the parameters that are customizable in the Network Integrity UI. The host property is initialized from the scan address.

CLI Connection Manager

This processor is part of the Discover Abstract Base CLI action.

This processor uses the CLIProperties object produced by the CLI Property Initializer processor or the CLI Property Customizer processor to establish a CLI connection. It provides a CLIConnection object to any extending cartridges and subsequent processors.

To establish a CLI connection, this processor performs the following operations:

1. Creates a connection to the CLI device using the values in the CLIProperties object.
2. Logs in to the CLI device using the credentials in the CLIProperties object.
3. If successful, returns the CLIConnection object.
4. When the scan ends, logs out and disconnects from the device.

   This processor logs any errors and fails the scan if it cannot establish a connection with the CLI device.

About Record and Playback

The CLI cartridge can be configured to record all discovered objects. The Record and Playback feature is not recommended for clustered environments because it relies on files being saved and loaded from the file system.

You enable the Record and Playback feature at run time by setting a managed bean (MBean) configured on a property group on the Discover Abstract CLI action.

When recording mode is enabled, the CLIConnection object writes the response data to a CLI data file in *Local_Dir***/***IP_HostName***/***Filename***.rec**, where:

- *Local_Dir* is a local directory that you can configure in the MBean at runtime. If you do not set a value in the MBean, *Local_Dir* is set to *Domain_Home***/cliData**, where *Domain_Home* is the Network Integrity domain.
- *IP_HostName* is the value of the host property on the CLIProperties object.
- *Filename* the name of a file that contains either the command code passed to the CLI Connection Manager processor or the full command string.

For example: *Local_Dir***/10/156/66/191/***Filename***.rec**

When playback mode is enabled, CLIConnection reads the CLI data file (created in Record mode and stored on the local hard drive) and sends the data back to the discovery cartridge without polling any network devices. The resource adapter does not require a connection to the network device.

See "About Using Record and Playback" for more information about enabling or disabling the Record and Playback feature.

SSH Login Behavior

The following sequence describes the SSH login behavior:

- If the SSH Known Host File Location parameter is not set, the CLI cartridge accepts all public keys from target devices.
- If the SSH Known Host File Location parameter is set:
  - And Network Integrity cannot write to the specified directory, the behavior is to accept all public keys from target devices.
  - And Network Integrity can write to the specified directory, all public keys are stored in the directory for each connection to a device, and all public keys are verified when the cartridge reconnects with a device to ensure that keys match. If the keys do not match, the public key is replaced.

Using the CLI Cartridge

This section describes how to use the CLI cartridge after it is deployed to the server.

Creating a Discovery Scan Action Type for CLI Devices

You can create a discovery scan to discover CLI device information in one or more network systems.

The CLI cartridge has the following scan parameter groups:

- **CLI Parameters**: A group of the parameters that you must configure to connect to devices and retrieve information using CLI commands over Telnet or Secure Shell (SSH) protocol.
- **CLI Device Discovery Parameters**: A group of parameters that you configure for specific device types, specific operating systems on which the device is running, and the services on the device.

To create a CLI discovery scan, follow the instructions explained in "Using Network Integrity" in *Network Integrity Online Help* and do the following during the creation process:

1. On the **General** tab, from the **Scan Action** list, select **Discover Cisco IOS XR VPLS CLI**.

   The **Scan Type** field displays **Discovery**.

2. Under the **Scan Action Parameters** area, do the following:
   - From the **Select Parameter Group** list, select **CLI Parameters** and enter the required information in the following fields:
     - From the **Transport Protocol** list, select the protocol used by the CLI device. Available options are: **SSH** and **Telnet**.
     - If you selected **Telnet** from the **Transport Protocol** list, in the **Telnet Login Prompt** field, enter the Telnet prompt for the login name on the CLI device.
     - If you selected **Telnet** from the **Transport Protocol** list, in the **Telnet Password Prompt** field, enter the Telnet prompt for the password on the CLI device.
     - In the **Prompt** field, specify the prompt for the CLI device.
     - In the **Continue Response Prompt** field, enter the CLI prompt to resume response for long responses on the CLI device.
     - If you selected **SSH** from the **Transport Protocol** list, in the **SSH Known Host File Dir** field, enter the directory location where the public keys of the CLI device are stored. Leave blank to accept all public keys. If a directory location is provided, the Network Integrity server must have permission to write to it; otherwise, the CLI cartridge defaults to accepting all public keys.
     - In the **Port** field, enter the CLI connection port.
     - In the **Login Name** field, enter the user name for logging in to the CLI device.
     - In the **Password** field, enter the user password for logging in to the CLI device.
     - In the **Retries** field, enter the number of retries that the cartridge attempts after a timeout.
     - In the **Timeout (Seconds)** field, enter the timeout length (in seconds) of the underlying socket connection.
   - From the **Select Parameter Group** list, select **CLI Device Discovery Parameters** and enter the required information in the following fields:
     - From the **Device Type** list, select the type of the device. For example, Cisco.
     - From the **Service** list, select the type of service to be discovered on the device. For example, Virtual Private LAN Service.
     - From the **Software Version** list, select the operating system of the device. For example, Cisco IOS XR.
3. On the **Scope** tab, do one of the following:
   - Enter an IP address.
   - Enter a range of IP addresses (for example, **192.0.2.*** or **198.51.100.11-23** or **198.51.100.1/24**).

- Enter a host name.

The CLI cartridge supports IPv4 and IPv6 IP address formats. You can enter multiple IP addresses.

4. On the **Schedule** tab, define a schedule for the scan.

   See the Design Studio Modeling Network Integrity Help for more information about defining a scan schedule.

5. Make any other required configurations.
6. Click **Save and Close**.

About Using Record and Playback

The record feature allows you to record CLI responses from devices for auditing, demonstration, or debugging the cartridge. The playback feature allows you to replay recorded files to simulate interaction with the device.

When record mode is enabled, the raw CLI responses are written to a CLI data file stored on the server.

When playback mode is enabled, the CLI connection reads the CLI data file (created in Record mode and stored on the server) and sends the data back to the discovery cartridge. The resource adapter does not require a connection to the network device.

> **Note:**
>
> Record and Playback are not recommended for clustered environments because it relies on files being saved and loaded from the file system.

A property group on the Discover Abstract CLI action controls the Record and Playback feature. MBeans allows you to adjust the record and playback functionality in the runtime system without the need to restart systems or servers.

The Record and Playback feature **mode** property has these valid values:

- Normal: The recording mode and playback mode are disabled.
- Record: The recording mode is enabled.
- Playback: The playback mode is enabled.

Viewing and Configuring the Current Record and Playback Mode

You use the MBean Browser in Oracle Enterprise Manager is used to view the **mode** property of the Record and Playback feature. See "Network Integrity System Administration Overview" in *Network Integrity System Administrator's Guide* for more information.

To view the current Record and Playback feature mode:

1. In the MBean Browser, navigate to the **oracle.communications.integrity.ActionProperty.ActionProperties** MBean.
2. Run the listPropertyGroups operation.

   This operation lists the configurable property groups. The Returned Value table displays the current mode.

To configure the Record and Playback feature mode:

1. In the MBean Browser, navigate to the **oracle.communications.integrity.ActionProperty.ActionProperties** MBean.
2. Select the *Action_Name***:CLI Property Initializer:RecordPlayback** property group, where *Action_Name* is the name of the extending action.
3. Run the listProperties action, using the full property group name in the argument.
4. Copy the *Action_Name***:CLI Property Initializer:RecordPlayback:mode** string from the Return Value table.
5. Open the setProperty operation and paste the string into the **Property** field.
6. In the Value field, set the value of the Record and Playback feature to either **Normal**, **Record**, or **Playback**.
7. Click the **Invoke** button.

To set the location where the recorded data is saved, open the **recordFileDir** action property and specify the desired directory. The directory must exist on the server and must be accessible by the Oracle WebLogic Server user.

Design Studio Construction

This section provides information about the composition of the CLI cartridge from the Design Studio perspective.

Actions

The following tables outline the Design Studio construction of the CLI cartridge actions and associated components:

- Table 2-2
- Table 2-3
- Table 2-4
- Table 2-5

> **Note:**
>
> Parameter values are case-sensitive and must be entered in capital letters when commands are run from a command line interface.

Table 2-2 CLI Cartridge Actions

Action Name	Result Category	Address Handler	Scan Parameter Group	Processors
Discover Abstract Base CLI	Device	IPAddressHandler	N/A	• CLI Property Initializer • CLI Connection Manager
Discover Abstract CLI	Device	IPAddressHandler	CLI Parameters CLI Device Discovery Parameters	• CLI Property Initializer • CLI Property Customizer • CLI Connection Manager

Table 2-3 CLI Cartridge Scan Parameter Group

Parameter Name	Parameter Type	Description	UI Label
loginName	Text box	The user name for logging in to the CLI device.	Login Name
password	Secret text	The user password or private identifier.	Password
telnetLoginPrompt	Text box	The Telnet prompt for the login name on the CLI device.	Telnet Login Prompt
telnetPasswordPrompt	Text box	The Telnet prompt for the password on the CLI device.	Telnet Password Prompt
prompt	Text box	The prompt on the CLI device.	Prompt
continueResponsePrompt	Text box	The continue response prompt for long responses on the CLI device.	Continue Response Prompt
port	Text box	The port number of the CLI device. The default is 22.	Port
timeout	Text box	The timeout in seconds of the underlying socket connection. The default is 60.	Timeout
retries	Text box	The number of retries that are attempted after a timeout. The default is 2.	Retries
knownHostFileDir	Text box	The location where the public keys of the CLI device are stored if SSH is used.	SSH Known Host File Dir
transportProtocol	Drop Down	The connection protocol to use: Telnet or SSH. The default is SSH.	Transport Protocol

Table 2-4 CLI Cartridge Device Discovery Scan Parameter Group

Parameter Name	Parameter Type	Description	UI Label
deviceType	Drop Down	The device vendor type.	Device Type
softwareVersion	Drop Down	Software version of the device.	Software Version
service	Drop Down	Type of service to be discovered on the device.	Service

Table 2-5 CLI Cartridge Processors

Processor Name	Variable
CLI Property Initializer	Input: N/A Output: cliProperties
CLI Property Customizer	Input: N/A Output: cliProperties
CLI Connection Manager	Input: cliProperties Output: cliConnection

Design Studio Extension

This section contains examples and explanations of how to extend certain aspects of the CLI cartridge by using Oracle Communications Design Studio. See "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide* for more information.

Extending the Discover Abstract CLI Action

This example explains how to extend the Discover Abstract CLI action to collect device-specific information using CLI commands and model the collected information into the Oracle Communications Information Model.

1. Open Design Studio in the Design perspective.
2. Create a Network Integrity cartridge project; for example, CLIDeviceExtension.
3. Make the CLIDeviceExtension cartridge project dependent on the CLI cartridge project.
4. Create a new discovery action; for example, CLI Extension.
5. Add the Discover Abstract CLI action as a processor in the CLI Extension action.
6. Add any new scan parameter groups in the CLI Extension action.
7. Create a new discovery processor; for example, Custom CLI Device Collector, and specify **cliConnection** and **cliProperties** as input parameters to this processor.
8. In the Custom CLI Device Collector processor implementation, add code to run any new CLI commands using the connection object, as shown in the following example:

```
CLIConnection connection = request.getCliConnection();
String resp = connection.runCommand("dir");
```

9. Create a new discovery processor to model the scan results (for example, Custom CLI Device Modeler) and then do the following:
    a. Specify the output from the Custom CLI Device Collector processor as input to the Custom CLI Device Modeler processor.
    b. In the Custom CLI Device Modeler processor implementation, add code to model the data into the required scan result entities.
10. Create a new discovery processor to persist the scan results (for example, Custom CLI Device Persister) and then do the following:
    a. Specify the output from the Custom CLI Device Modeler processor as input to the Custom CLI Device Persister processor.
    b. In the Custom CLI Device Persister processor implementation, add code to persist the entities into the database.
11. Build, deploy, and test your cartridge.

The new custom processors are run in the order shown in Figure 2-3.

Figure 2-3 New Custom Processors Workflow

3

About the VPLS Reference Cartridge

This chapter describes the functionality and design of the Oracle Communications Network Integrity Virtual Private LAN Service (VPLS) reference cartridge and how to use and build the cartridge.

About Actions and Processors

The VPLS reference cartridge contains the Discover Cisco IOS XR VPLS CLI action.

The Discover Cisco IOS XR VPLS CLI action does the following:

- Discovers Cisco IOS XR devices in a VPLS service
- Retrieves information about VPLS services on the Cisco IOS XR devices
- Models logical device hierarchies that represent the discovered Cisco IOS XR device
- Models the VPLS service hierarchy

This discovery action inherits all the processors from the Discover Abstract CLI actions. See "About the CLI Cartridge" for more information about the inherited processors.

The Discover Cisco IOS XR VPLS CLI action contains the following processors run in the following order:

1. CLI Property Initializer (inherited)
2. CLI Property Customizer (inherited)
3. CLI Connection Manager (inherited)
4. Cisco IOS XR VPLS CLI Device Collector
5. Cisco IOS XR VPLS CLI Device Modeler
6. Cisco IOS XR VPLS CLI Device Persister

Figure 3-1 illustrates the processor workflow of the Discover Cisco IOS XR VPLS CLI action.

Figure 3-1 Discover Cisco IOS XR VPLS CLI Action Processor Workflow

Cisco IOS XR VPLS CLI Device Collector

This processor retrieves the device data and VPLS network data using CLI commands and makes the data available for modeling. This processor runs commands specific to Cisco IOS XR device.

Cisco IOS XR VPLS CLI Device Modeler

This processor models the data collected by the Cisco IOS XR VPLS CLI Device Collector processor into logical device hierarchy and VPLS services.

Cisco IOS XR VPLS CLI Device Persister

This processor persists the logical device tree and the VPLS services data to the Network Integrity database.

About Collected Data

This section describes the data that is collected for modeling, listing and explaining each command response.

CLI Commands

This section provides information about the CLI commands and the sample responses that the processors may receive from a device.

show version

This command displays the device type, software version, and host name.

Example 3-1 shows the sample response for the show version CLI command.

Example 3-1 Sample Response for show version Command

```
Tue Feb 24 01:27:55.783 EST

Cisco IOS XR Software, Version 4.2.4[Default]
Copyright (c) 2012 by Cisco Systems, Inc.

ROM: System Bootstrap, Version 12.00(20100127:230559) [skumarss-33s 1.24]
RELEASE SOFTWARE
Copyright (c) 1994-2010 by cisco Systems, Inc.
rotgsr-2 uptime is 6 weeks, 5 days, 2 hours, 42 minutes
System image file is "disk0:c12k-os-mbi-4.2.4/mbiprp-rp.vm"

cisco 12410/PRP (7457) processor with 3670016K bytes of memory.
7457 processor at 1265Mhz, Revision 1.2
GSR 12410 200 Gbps

1 Cisco 12000 Series Performance Route Processor
1 Cisco 12000 4 Port Gigabit Ethernet Controller (4 GigabitEthernet)
3 Management Ethernet
1 PLIM_QOS
4 GigabitEthernet/IEEE 802.3 interface(s)
1019k bytes of non-volatile configuration memory.
2000352k bytes of disk0: (Sector size 512 bytes).
46692k bytes of disk1: (Sector size 512 bytes).
65536k bytes of Flash internal SIMM (Sector size 256k).
...........
```

show interfaces

This command lists all the available interfaces.

Example 3-2 shows the sample response for the show interfaces CLI command.

Example 3-2 Sample Response for show interfaces Command

```
Wed Feb 25 01:14:42.200 EST
Loopback0 is up, line protocol is up
Interface state transitions: 1
Hardware is Loopback interface(s)
Internet address is 12.12.12.12/32
MTU 1500 bytes, BW 0 Kbit
reliability Unknown, txload Unknown, rxload Unknown
Encapsulation Loopback,  loopback not set,
Last input Unknown, output Unknown
Last clearing of "show interface" counters Unknown
Input/output data rate is disabled.

MgmtEth0/4/CPU0/1 is up, line protocol is up
Interface state transitions: 1
Hardware is Management Ethernet, address is 0011.bcf1.8965 (bia 0011.bcf1.8965)
Description: Management Interface DOT NOT CONFIGURE
Internet address is 198.51.100.254/24
MTU 1514 bytes, BW 100000 Kbit (Max: 100000 Kbit)
```

```
reliability 254/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 100Mb/s, TX, link type is autonegotiation
output flow control is off, input flow control is off
loopback not set,
ARP type ARPA, ARP timeout 04:00:00
Last input 00:00:03, output 00:00:03
Last clearing of "show interface" counters never
5 minute input rate 0 bits/sec, 1 packets/sec
5 minute output rate 3000 bits/sec, 2 packets/sec
2296633 packets input, 178794962 bytes, 3 total input drops
0 drops for unrecognized upper-level protocol
Received 1570634 broadcast packets, 711099 multicast packets
153 runts, 0 giants, 0 throttles, 0 parity
418 input errors, 265 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
978323 packets output, 68803752 bytes, 0 total output drops
Output 237581 broadcast packets, 8 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
1 carrier transitions
...........
```

show l2vpn bridge-domain detail

This command retrieves information about all the VPLS instances running on the device. Following are some of the important attributes that can be extracted from the command response:

- Bridge Group
- Bridge Domain
- Neighbors (PEs only; does not include transit Providers)
- Pseudo-wire IDs connecting to different neighbors
- BGP Auto-discovery
- VFI ID
- VPN ID
- VPLS ID
- Signaling Protocol
- MPLS Label
- Remote MPLS Label
- Route Distinguisher
- Import Route Target
- Export Route Target
- VE ID
- Advertised VE ID
- VE Range
- AS Number
- L2VPN Router ID

Example 3-3 shows the sample response for the show l2vpn bridge-domain detail CLI command.

Example 3-3 Sample Response for show l2vpn bridge-domain detail Command

```
Mon Mar 16 06:55:49.990 EDT
Legend: pp = Partially Programmed.
Bridge group: niVpls, bridge-domain: niDomain12k, id: 3, state: up, ShgId: 0, MSTi: 0
Coupled state: disabled
MAC learning: enabled
MAC withdraw: enabled
MAC withdraw for Access PW: enabled
MAC withdraw sent on bridge port down: disabled
Flooding:
Broadcast & Multicast: enabled
Unknown unicast: enabled
MAC aging time: 300 s, Type: inactivity
MAC limit: 4000, Action: none, Notification: syslog
MAC limit reached: no
MAC port down flush: enabled
MAC Secure: disabled, Logging: disabled
Split Horizon Group: none
Dynamic ARP Inspection: disabled, Logging: disabled
IP Source Guard: disabled, Logging: disabled
DHCPv4 snooping: disabled
IGMP Snooping profile: none
Bridge MTU: 1500
MIB cvplsConfigIndex: 4
Filter MAC addresses:
Create time: 08/02/2015 23:08:51 (5w0d ago)
No status change since creation
ACs: 1 (0 up), VFIs: 1, PWs: 1 (0 up), PBBs: 0 (0 up)
List of ACs:
    AC: GigabitEthernet0/5/0/0.5, state is unresolved
  MAC learning: enabled
  Flooding:
      Broadcast & Multicast: enabled
      Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 4000, Action: none, Notification: syslog
  MAC limit reached: no
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 snooping: disabled
  IGMP Snooping profile: none
  Storm Control: disabled
  Static MAC addresses:
List of Access PWs:
List of VFIs:
  VFI 1 (up)
    PW: neighbor 198.51.100.254, PW ID 1, state is down ( provisioned ) (Transport LSP
Down)
PW class not set, XC ID 0xff000009
   Encapsulation MPLS, protocol LDP
   Source address 12.12.12.12
   PW type Ethernet, control word disabled, interworking none
   PW backup disable delay 0 sec
   Sequencing not set
   PW Status TLV in use
MPLS        Local     Remote
------------------------------
Label       16018     unknown
```

```
Group ID    0x3       0x0
Interface   1         unknown
MTU         1500      unknown
Control word disabled unknown
PW type     Ethernet  unknown
VCCV CV type 0x2      0x0
(LSP ping verification) (none)
VCCV CC type 0x6  0x0 (none)
(Router Label) (TTL Expiry)
------------------------------
MIB cpwVcIndex: 4278190089
  Create time: 08/02/2015 23:08:51 (5w0d ago)
  Last time status changed: 08/02/2015 23:09:07 (5w0d ago)
  MAC withdraw message: send 0 receive 0
  Static MAC addresses:
  DHCPv4 snooping: disabled
  IGMP Snooping profile: none
VFI Statistics:
  drops: illegal VLAN 0, illegal length 0
```

About Cartridge Modeling

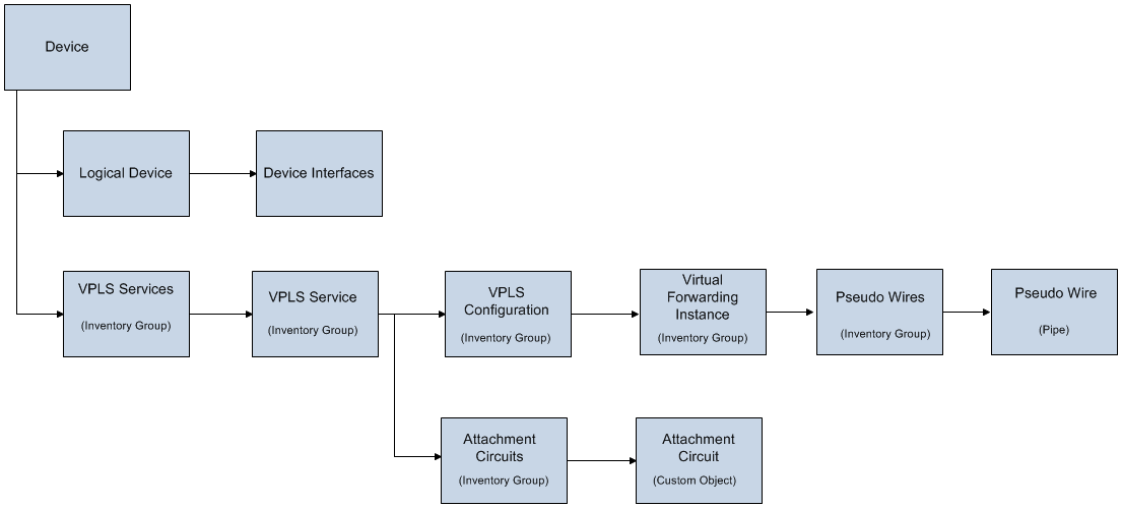
The VPLS reference cartridge models collected data according to the Oracle Communications Information Model. Collected data is modeled into the following entities:

- LogicalDevice
- DeviceInterface
- InventoryGroup
- Pipe

See *Oracle Communications Information Model Reference* for more information. *Oracle Communications Information Model Reference* is located in the Network Integrity Developer Documentation on the Oracle software delivery website.

Figure 3-2 shows how the VPLS reference cartridge models collected data according to the Oracle Communications Information Model.

Figure 3-2 Collected Data Modeling



Field Mapping

The following tables explain the field mappings for each Information Model object.

Table 3-1 Logical Device Mappings

Information Model Attribute	Information Model Support	Type
Description	Static	String
Device Identifier	Static	String
ID	Static	String
Name	Static	String
Network Location Entity Code	Static	String
Management IP Address	Dynamic	String
Native EMS Admin Service State	Dynamic	String
Native EMS Name	Dynamic	String
Native EMS Service State	Dynamic	String
System Object ID	Dynamic	String

Table 3-2 Device Interface Mappings

Information Model Attribute	Information Model Support	Type
Customer Interface Number	Static	String
Description	Static	String
ID	Static	String
Interface Number	Static	String
Name	Static	String
Nominal Speed	Static	String
Vendor Interface Number	Static	String
IfAlias	Dynamic	String
IfName	Dynamic	String
ifPromiscuousMode	Dynamic	String
Current MTU	Dynamic	String
Supported MTU	Dynamic	String
Native EMS Admin Service State	Dynamic	String

Table 3-2 (Cont.) Device Interface Mappings

Information Model Attribute	Information Model Support	Type
Native EMS Connector Present	Dynamic	String
Native EMS Name	Dynamic	String
Native EMS Service State	Dynamic	String
Physical Address	Dynamic	String

Table 3-3 Inventory Group (VPLS Service) Mappings

Information Model Attribute	Information Model Support	Type
Name	Static	String
Description	Static	String

Table 3-4 Inventory Group (VPLS Configuration) Mappings

Information Model Attribute	Information Model Support	Type
Name	Static	String
Description	Static	String
BridgeGroup	Dynamic	String
BridgeDomain	Dynamic	String
VpnId	Dynamic	String
BridgeMTU	Dynamic	String
PeDiscovery	Dynamic	String
BroadcastAndMulticast	Dynamic	String
UnknownUnicast	Dynamic	String
SplitHorizonGroup	Dynamic	String
SignalingProtocol	Dynamic	String
VeRange	Dynamic	String
VeId	Dynamic	String
AdvertisedVEId	Dynamic	String
VplsId	Dynamic	String
AsNumber	Dynamic	String
L2VPNRouterId	Dynamic	String
RouteDistinguisher	Dynamic	String
ImportRouteTarget	Dynamic	String
ExportRouteTarget	Dynamic	String

Table 3-5 Inventory Group (Virtual Forwarding Instance) Mappings

Information Model Attribute	Information Model Support	Field Type
Name	Static	String

Table 3-5 (Cont.) Inventory Group (Virtual Forwarding Instance) Mappings

Information Model Attribute	Information Model Support	Field Type
Description	Static	String
Status	Dynamic	String

Table 3-6 Inventory Group (Attachment Circuit) Mappings

Information Model Attribute	Information Model Support	Field Type
Name	Static	String
Description	Static	String
ID	Static	String
Status	Dynamic	String

Table 3-7 Pipe Mappings

Information Model Attribute	Information Model Support	Field Type
Name	Static	String
Description	Static	String
ID	Static	String
Medium	Static	String
Transmission Signal Type	Static	String
Gap Pipe	Static	Boolean
Versioned	Static	Boolean
PwType	Dynamic	String
PwStatus	Dynamic	String
PwClass	Dynamic	String
LoopbackInterface	Dynamic	String
Protocol	Dynamic	String
Encapsulation	Dynamic	String
MplsLabel	Dynamic	String
RemoteMPLSLabel	Dynamic	String
AEnd	Dynamic	String
ZEnd	Dynamic	String

Using the VPLS Reference Cartridge

This section describes how to use the VPLS reference cartridge.

Creating a VPLS Discovery Scan

The VPLS reference cartridge does not introduce any new scan parameter groups. Because the VPLS reference cartridge extends the CLI cartridge, the configurable scan parameter groups are those from the CLI cartridge.

VPLS Service Discovery

To discover a VPLS service on a Cisco IOS XR device, create a VPLS discovery scan in the same way that you create a discovery scan for CLI devices. See "Using the CLI Cartridge" for the procedure. When setting the values in the **General** tab in the Scan Action Parameters area, do the following:

- From the **Select Parameter Group** list, select **CLI Device Discovery Parameters** and enter the required information in the following fields:
  - From the **Device Type** list, select the type of the device. For example, Cisco Device.
  - From the **Service** list, select the type of service to be discovered on the device. For example, Virtual Private LAN Service.
  - From the **Software Version** list, select the operating system of the device. For example, Cisco IOS XR.

Discovered Results

Discovered results have a result group for each device.

Figure 3-3 shows sample data for VPLS configuration discovered by the VPLS reference cartridge.

Figure 3-3 Sample Discovered Data for VPLS Configuration

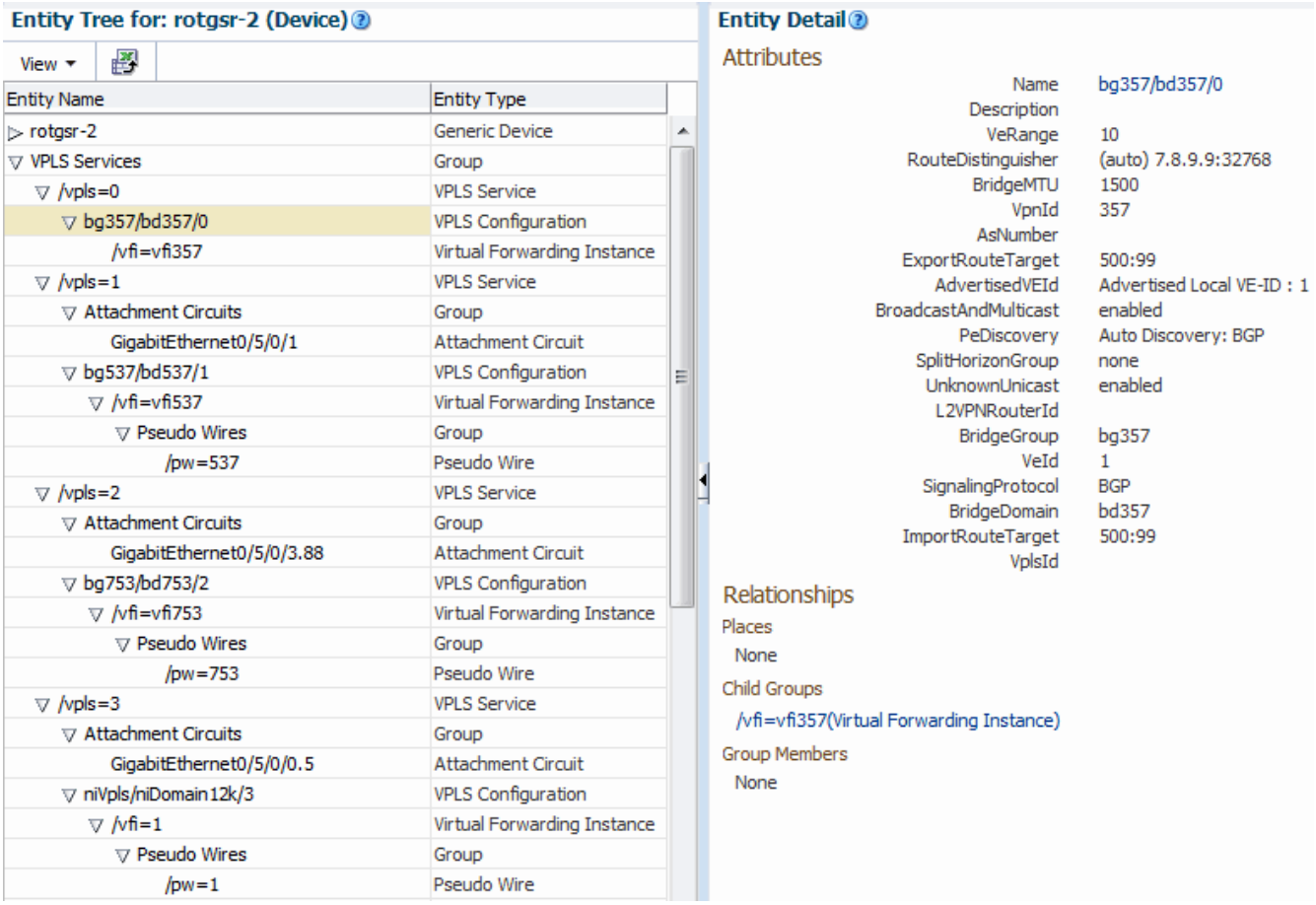


Figure 3-4 shows sample data for an attachment circuit discovered by the VPLS reference cartridge.

Figure 3-4 Sample Discovered Data for an Attachment Circuit

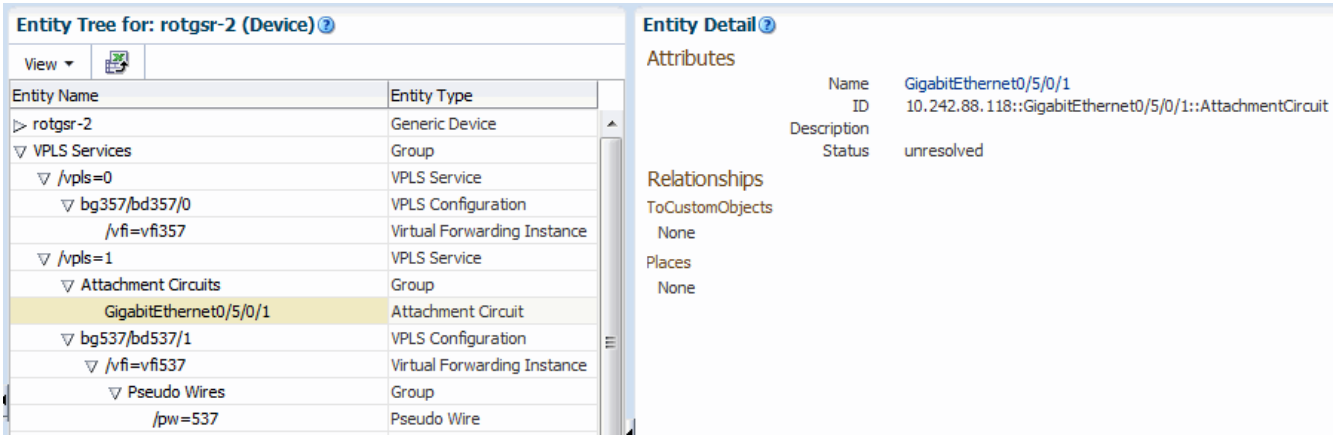


Figure 3-5 shows sample data for a virtual forward instance discovered by the VPLS reference cartridge.

Figure 3-5 Sample Discovered Data for a Virtual Forwarding Instance

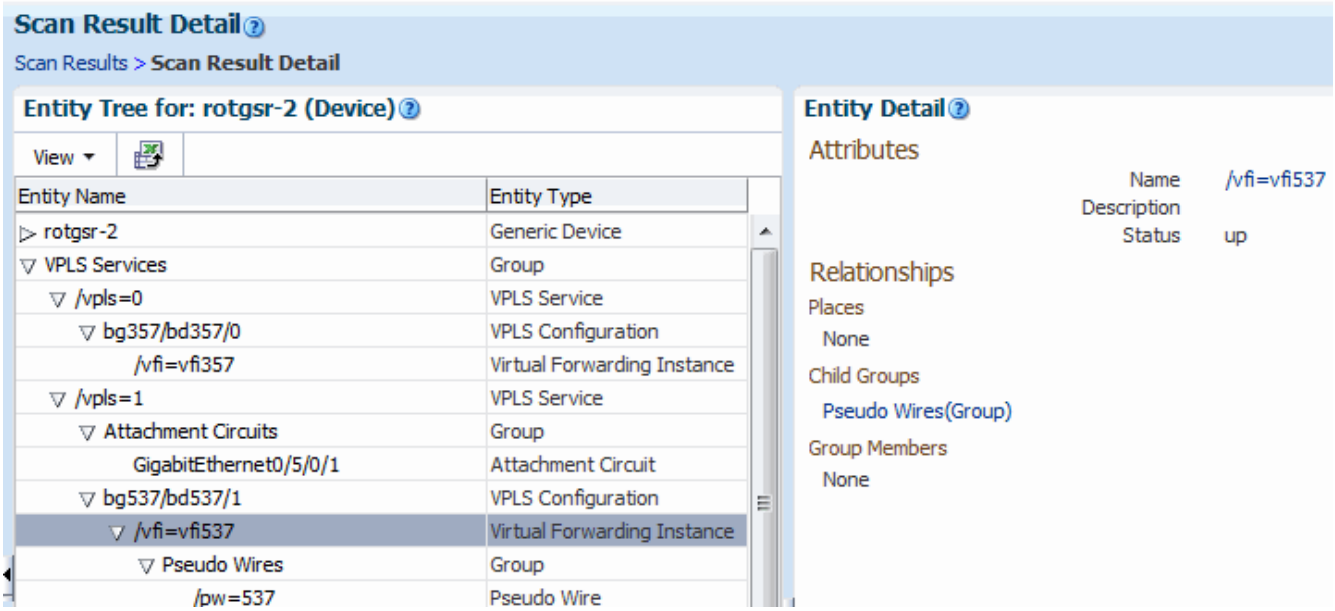
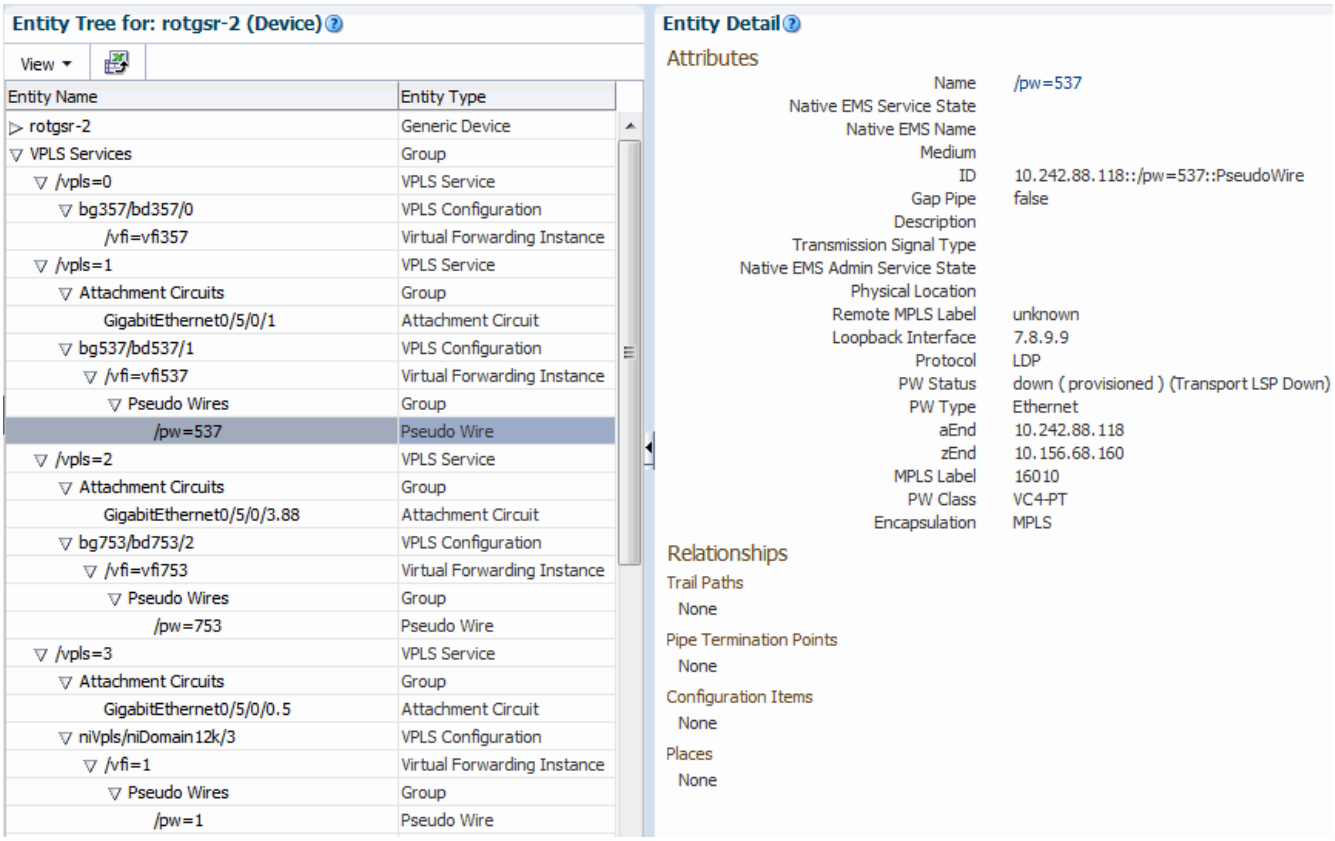


Figure 3-6 shows sample data for pseudo wire discovered by the VPLS reference cartridge.

Figure 3-6 Sample Discovered Data for a Pseudo Wire



Design Studio Construction

This section provides information about the composition of the VPLS reference cartridge from the Oracle Communications Design Studio perspective.

Model Collections

Table 3-8 shows the model collection used in the VPLS reference cartridge.

Table 3-8 VPLS Model Collection

Specification	OCIM Model Entity Type
Generic Device (ora_ni_uim_device)	Logical Device
Generic Interface (ora_ni_uim_device)	Device Interface
VPLS Service	Inventory Group
VPLS Configuration	Inventory Group
Virtual Forwarding Instance	Inventory Group
Attachment Circuit	Custom Object
Pseudo Wire	Pipe

Actions

The following tables outline the Design Studio construction of the VPLS reference actions and associated components:

- Table 3-9
- Table 3-10

Table 3-9 VPLS Cartridge Actions

Action Name	Result Category	Address Handler	Scan Parameter Groups	Processors
Discover Cisco IOS XR VPLS CLI	Device	IPAddressHandler	N/A	• CLI Property Initializer • CLI Property Customizer • CLI Connection Manager • Cisco IOS XR VPLS CLI Device Collector • Cisco IOS XR VPLS CLI Device Modeler • Cisco IOS XR VPLS CLI Device Persister

Table 3-10 VPLS Cartridge Processors

Processor Name	Variable
CLI Property Initializer	See "Table 2-5" for more information.
CLI Property Customizer	See "Table 2-5" for more information.
CLI Connection Manager	See "Table 2-5" for more information.

Table 3-10 (Cont.) VPLS Cartridge Processors

Processor Name	Variable
Cisco IOS XR VPLS CLI Device Collector	Input: • cliConnection • cliProperties Output: entityList
Cisco IOS XR VPLS CLI Device Modeler	Input: entityList Output: • logicalTree • vplsServices
Cisco IOS XR VPLS CLI Device Persister	Input: • logicalTree • vplsServices Output: N/A

Design Studio Extension

This section provides information about Design Studio extensions to the VPLS reference cartridge.

The VPLS reference cartridge contains code used to run CLI commands specific to Cisco IOS XR devices, parse command responses, and model the VPLS service as an unsealed cartridge. You can change any part of the code to customize this cartridge to fit your environment, or you can use the code as an example on which to model your own custom VPLS cartridge. For more information about extensibility, see "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide* and "About Network Integrity" in *Network Integrity Concepts*.