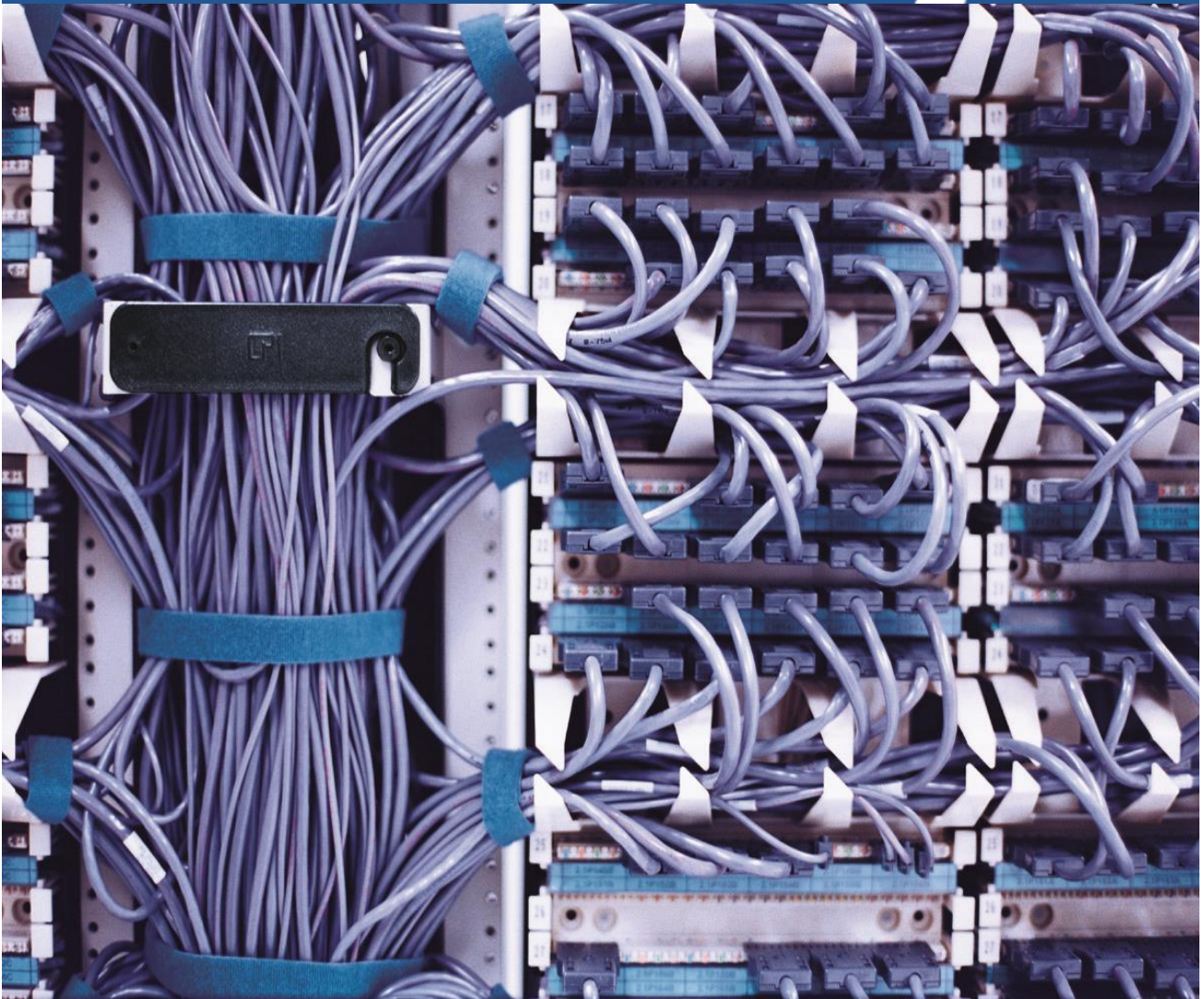


# HPSA - VPN SVP 7.0

## Service Discovery Guide



## Legal Notices

### Warranty

*Hewlett-Packard makes no warranty of any kind with regard to this manual, 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 2001-2015 Hewlett-Packard Development Company, L.P., all rights reserved.

No part of this document may be copied, reproduced, or translated to 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

Jboss is a registered trademark of Red Hat, Inc.

Linux is a U.S. registered trademark of Linus Torvalds

Oracle® and Java™ are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of the Open Group.

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

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Printed in the DK

# Contents

<b>1</b>	<b>Introduction</b>	
1-1	Intended Audience .....	5
1-2	Background .....	5
1-3	Install Location Descriptors .....	5
<b>2</b>	<b>What does Service Discovery do?</b>	
2-1	What does Service Discovery support? .....	7
2-1-1	What does Service Discovery not support? .....	8
2-2	What are the requirements? .....	8
2-2-1	Hardware Requirements .....	8
2-2-2	Software Requirements .....	8
<b>3</b>	<b>The Service Discovery Process</b>	
3-1	Overview of the Process .....	10
3-1-1	Pre-configuration and Equipment Load .....	10
3-1-2	Load Phase .....	11
3-1-3	Compare .....	11
3-1-4	Reconcile .....	11
<b>4</b>	<b>Configuring VPN SVP before start</b>	
4-1	Required Configuration Steps before start .....	13
4-1-1	VPN SVP Configuration .....	13
4-1-2	Equipment population .....	13
4-1-3	Manual equipment configuration steps .....	14
<b>5</b>	<b>Customizing Service Discovery</b>	
5-1	Modifying Patterns .....	17
5-1-1	Understanding the existing patterns .....	17
5-1-2	Matching interface descriptions .....	18
5-1-3	Pattern for matching VRF description .....	19
5-1-4	Customizing the patterns .....	20
5-2	Other configurable properties .....	21
5-3	Advanced Customization .....	22
<b>6</b>	<b>Running the Service Discovery</b>	
6-1	Approaches .....	23
6-1-1	Moving Data from Test Server to Production Server .....	23
6-2	Command line options .....	24
<b>7</b>	<b>Report Generation</b>	
<b>8</b>	<b>Parse and Analysis Details</b>	
8-1	Rules for pairing route-target values into Routing Communities .....	26
<b>9</b>	<b>Resolving Service Ownership</b>	
9-1	Required operator interaction .....	28
9-1-1	Cause of Ambiguity in Ownership .....	28
9-1-2	Automatic Resolving of Ambiguities .....	30
9-1-3	Manual Resolving of Ambiguities .....	30
9-1-3-1	Understanding the Service Discovery View and Icons .....	31
9-1-3-2	Resolving Ambiguous Customer Ownership of Sites .....	32
9-1-3-3	Resolving Ambiguous Customer Ownership of VPNs .....	33
9-1-3-4	Resolving Ownership of a Set of Services in one Operation .....	34
9-1-3-5	Merging customers .....	35
<b>10</b>	<b>Operator Validation of Data</b>	
10-1	Understanding the XML service file .....	37
10-1-1	Understanding the XML attributes .....	37
10-1-2	Understanding the Service Identifiers .....	38
10-1-3	Service Data Hierarchy .....	38
10-1-4	Service Object Relationships .....	40

10-1-5 Identifier.....	40
10-1-6 Traffic Classifier.....	41
10-1-7 Policy Mapping.....	41
10-1-8 QoS Profile.....	41
10-1-9 VRF.....	42
10-1-10 CE Router.....	42
10-1-11 Interface.....	43
10-1-12 IP Net.....	44
10-1-13 Customer.....	44
10-1-14 L3VPN.....	44
10-1-15 Site.....	45
10-1-16 RC.....	45
10-1-17 RC Membership.....	45
10-1-18 L3AccessFlow.....	46
10-1-19 L3FlowPoint.....	47
10-1-20 FlowPoint.....	47
10-1-21 VPNFPMembership.....	48
10-1-22 L3 VPN Membership.....	48
<b>11 Updating Information in the CRM Repository</b>	
<b>12 Log Files</b>	
12-1 Location, naming convention and format of Log files.....	50
12-2 Logging Level.....	50
12-3 What is being logged?.....	51
12-4 Error and Warning Messages in Log files.....	51
12-4-1 Unreferenced VRF.....	51
12-4-2 No matching IP Address pool found.....	51
12-4-3 Unknown Router Protocol.....	52
12-4-4 Committed VRF differs from discovered.....	52
12-4-5 VPN Services Mark with Error.....	52
12-4-6 Can not delete Split Files.....	52
12-4-7 Service Ownership has not been resolved.....	53
12-4-8 Error preventing Commit of Data.....	53
12-5 Cleaning up of Log files.....	53
<b>Appendix A: Service XML File Format</b>	
DTD for XML File Format.....	54

# 1 Introduction

The Service Discovery for the VPN SVP provides a tool to discover L3VPN services in network equipment configuration files and populating the services into the VPN SVP service repository. After the services have been committed they can be managed in network using the VPN SVP software.

The Service Discovery component also provides a tool for populating the equipment resources from the network equipment configuration file.

## 1-1 Intended Audience

This guide provides information needed by the system integrators and operator running the service discovery. The system integrator's task is to install and configure the Service Discovery components. The operator's task is to run and interact with the Service Discovery process.

Working knowledge of HPSA and the VPN SVP is required. Also knowledge of the technologies used by these products, such as XML and Oracle is required.

## 1-2 Background

When introducing new provisioning software for managing VPN services, many providers already have a number of existing services running. These services may have been activated using manual activation, home-grown scripts or other existing provisioning tools such as Cisco ISC or VPN-SC. The existing services are an important revenue base which must be taken into consideration when planning for a new provisioning tool. It is therefore often a strict requirement from providers that the VPN SVP must be able to manage these existing services.

## 1-3 Install Location Descriptors

The following names are used to define install locations throughout this guide.

**Table 1-1** Install Location Descriptors

Descriptor	What the Descriptor Represents
<code>\$ACTIVATOR_OPT</code>	The base installation location of Service Activator. The UNIX location is <code>/opt/OV/ServiceActivator</code> The Windows location is <code>&lt;install drive&gt;:\HP\OpenView\ServiceActivator</code>
<code>\$ACTIVATOR_VAR</code>	The install location of specific Service Activator files. The UNIX location is <code>/var/opt/OV/ServiceActivator</code> The Windows location is <code>\$ACTIVATOR_OPT\var</code>
<code>\$ACTIVATOR_BIN</code>	The install location of specific Service Activator files. The UNIX location is <code>\$ACTIVATOR_OPT/bin</code> The Windows location is <code>\$ACTIVATOR_OPT\bin</code>
<code>\$JBOSS_HOME</code>	The install location for JBoss. The UNIX location is <code>/opt/HP/jboss</code> The Windows location is <code>&lt;install drive&gt;:\HP\jboss</code>

**Table 1-1** Install Location Descriptors

<b>Descriptor</b>	<b>What the Descriptor Represents</b>
<i>\$JBOSS_DEPLOY</i>	The install location of the Service Activator J2EE components. The UNIX location is <code>\$JBOSS_HOME/standalone/deployments</code> The Windows location is <code>\$JBOSS_HOME\standalone\deployments</code>
<i>\$SOLUTION</i>	The install location of the VPN SVP solution. The location is <code>\$ACTIVATOR_OPT\solutions\SAVPN</code>

---

## 2 What does Service Discovery do?

This section describes what features are supported by the Service Discovery, the hardware and software requirements to running the Service Discovery and what the Service Discovery software consists of.

### 2-1 What does Service Discovery support?

The following list of features are supported by the Service Discovery software

- Support for discovery of services in Cisco and Juniper equipment based on the configuration files. Cisco IOS version 12.x and Junos 9.4Rx is supported. Support for other vendors such as Huawei can be provided through a SI delivery project or possibly through the VPN SVP product group.
- The Service Discovery software populates new discovered services into the CRM and VPN SVP repository.
- The software discovers Layer 3 VPN services. This includes discovery of
  - Interfaces with L3VPN services,
  - VRF tables,
  - routing communities,
  - quality of service and rate limits,
  - routing protocol and static routes,
  - site-of-origin attributes for multihome-site and
  - IP addresses for the PE-CE attachment links.
- The software compares the discovered services with the existing services in the VPN SVP repository. Only new services and customers are added, existing services are not modified or replaced.
- The software provides mechanism for an operator to approve the updates of services that are committed to the VPN SVP repository.
- It is possible for an operator to correct and provide additional information before the discovered services are committed to repository, e.g. provide other customer or VPN names than those suggested by the Service Discovery software.
- The Service Discovery software generates log messages for services which are not compliant with VPN SVP conventions and that can not be loaded into the service repository. This includes messages regarding services which are not correctly configured, route target values which could not be matched with a routing community or VRF objects on the routers which are not in use.
- The Service Discovery software provides an equipment load tool, which populates PE-router and their interfaces into the VPN SVP repository. The tool can process many network element configurations in a single session.
- The services discovered in the network and which have been populated into the VPN SVP repository can be managed by the VPN SVP software. The following operations are supported
  - **Deletion of site**; the interface will be cleaned for VRF association and ip address. If the service is associated to a sub-interface, sub-interface will be removed. If the QoS is compliant with the VPN SVP convention and it is a last site using this QoS profile, the QoS will also be removed from the equipment.
  - **Add site to discovered VPN**
  - **Modify site to leave discovered VPN**
  - **Modify QoS and rate limit**; it is only possible to associate a new QoS and rate-limit compliant with VPN SVP convention
  - Modify Site to Join VPN

- Suspend and resume discovered services
- The service discovery can be performed without connectivity to the network. The service discovery only requires access to the files containing the configuration
- The Service Discovery software provides possibilities for the system integrator to configure the software according to a specific provider's convention:
  - the algorithm for determining VPN and customer names from description field on interface and VRF objects can be replaced.
  - it is possible to replace the algorithm for determining routing community pairs.

## 2-1-1 What does Service Discovery not support?

The following is not supported by the Service Discovery software:

- Layer 2 VPWS and VPLS services are not supported.
- For safety reasons the Service Discovery software will not delete services in the VPN SVP repository which has not been found in the network during a service discovery session. Neither will the software modify an existing service in the repository.
- Sites with QoS configuration which does not comply with the VPN SVP conventions (modular QoS) may be uploaded, but management of the site may not be fully supported, e.g. it may be possible to delete the site, but not modify the bandwidth or the QoS profile. It is a stretch goal if it should be possible to add VPN SVP QoS to an uploaded service.
- As for the VPN SVP there is only support for one administrative VPN Routing Community
- Upload of CE routers and access network is not supported. However for each discovered services there will be created a (simple) CE router object in the VPN SVP
- For services which do not comply with the VPN SVP QoS provisioning convention, the service-policy maps, class-maps and ACL will be left untouched on the router.
- Upload of multicast site is not supported.
- Clean up of "dead" objects in database is not supported (e.g. RC objects existing in the repository and which are not used by any services in the network are not detected and deleted)
- Upload of VRRP protocol is not supported.
- All the configuration files should be uploaded together. Incremental upload of the configuration files is not supported.
- Service Discovery may associate RIP protocol to sites with static routes.
- Service Discovery will identify both the initial attachment and protection attachment as initial attachment in case of BGP protocol.

## 2-2 What are the requirements?

Before the Service Discovery process can be executed there are requirements to the hardware and the software which must be in place.

### 2-2-1 Hardware Requirements

The service discovery does not need to take place on the production equipment but can be done off-line on any system where the VPN SVP is installed together with the Service Discovery components.

This allows system integrators to prepare the service discovery on a remote system.

Consult the HPSA and VPN SVP documentation for further details on hardware requirements.

### 2-2-2 Software Requirements

Consult the HPSA and VPN SVP documentation for further details on the software requirements.





# 3 The Service Discovery Process

This section gives an overview of the steps in the data load process.

Before a service discovery is run for the first time, a pre-configuration and an equipment load step is required. The pre-configuration consists of the configuration of the VPN SVP.

The equipment load populates routers and their interfaces.

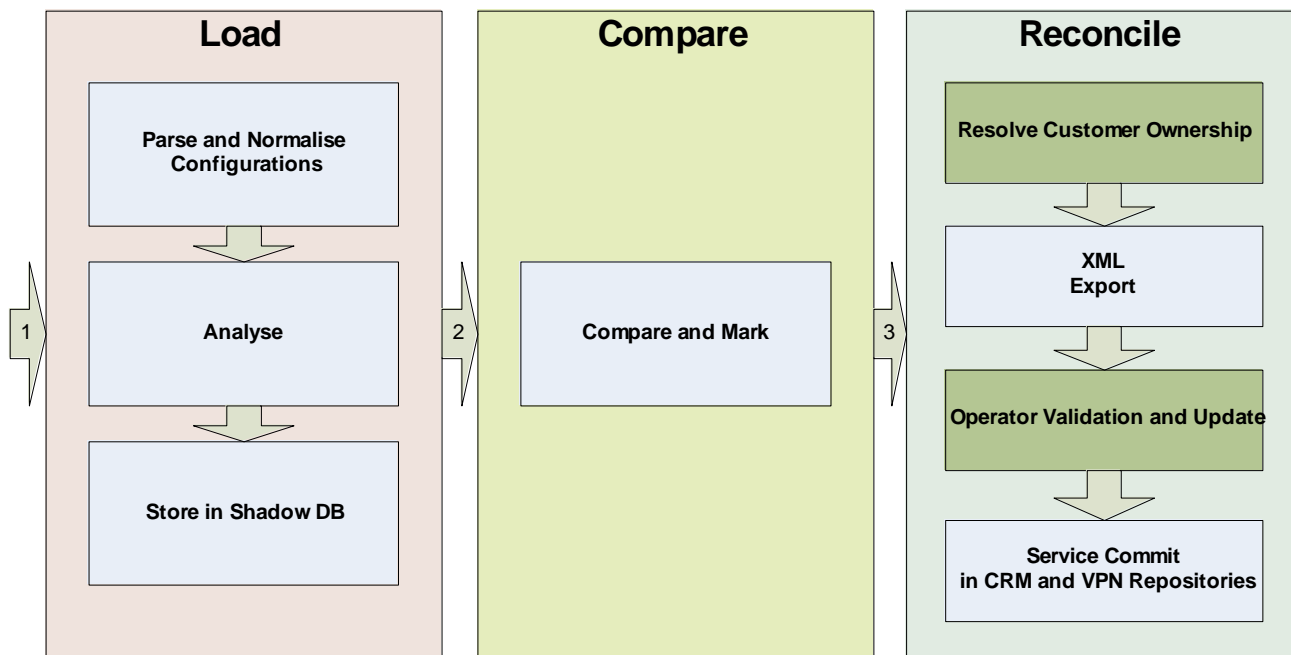
The actual discovery process is described in three phases:

- Load
- Compare
- Reconcile

Each phase consists of a number of steps. The phases are explained in the following sections.

## 3-1 Overview of the Process

**Figure 3-1:** The phases and steps in the service discovery process. Boxes in light blue show steps where no operator interactions is required. Boxes in olive green show steps where interaction may be required.



### 3-1-1 Pre-configuration and Equipment Load

Before the service upload can take place, a number of configurations must be done. This includes VPN SVP product configuration and equipment load.

As part of the standard configuration of the VPN SVP, the ASN and the admin VPN must be configured. Also the IP address pools' parent net must be configured. During the Service Discovery IPNet object will be created if they not already exist in the VPN SVP repository. However it is required that parent with a mask including the IPNet address exist in order to the software to know under which parent net the address must be created.

Comparison between the discovered services in the network and the existing services in the VPN SVP repository requires that all network elements and their termination points (interfaces) have been loaded before service discovery is started.

The equipment data can be populated using the same equipment configuration files as used for service discovery. The equipment load is performed with the loader tool provided with the Service Discovery software. This tools can process and all configuration files in a single session.

## 3-1-2 Load Phase

The purpose of the “Load Phase” is parsing information from an external source and bringing the data onto a form where it can be compared with existing data in VPN SVP repository. Depending on the source of data, different loaders can be used. Currently Cisco and Juniper Service loaders are provided, but other approaches can be envisioned during a system integration project. HP has on project basis delivered service discovery projects where the Cisco VPN-SC XML export files has been used to populate the services.

The load phase works in two steps. First step is parsing the configuration files and normalize the data for determining FlowPoint, AccessFlows, VRFs and QoS for each interface associated with a L3VPN service. Second step is analyzing the results to compute the routing communities. Following the L3 VPN Services and their Customers are calculated and the sites are associated to these.

Once data from the input has been parsed and analyzed, the data is populated into tables in the database. These tables are not the VPN SVP tables, but in so called shadows tables with the same layout as the existing VPN SVP tables. The service information are now on a form where it is ready to be compare with the existing services in the VPN SVP repository.

## 3-1-3 Compare

During the compare phase the loaded data is compared with the existing data in the VPN SVP repository. The compare functionality can compare the discovered services with the VPN SVP repository are determine if the discovery services are new or if they already exists.

The result of the comparison is to mark the data in the shadow tables, if the services are subject for creation or if the exists.

## 3-1-4 Reconcile

The purpose of the reconciliation process is to get the VPN SVP repository updated with the service data loaded from the external source. The reconciliation process consists of four steps.

1. Resolve Customer Ownership
2. Service XML file Export
3. Operator confirmation
4. Repository Update

After the compare phase has completed, the operator may inspect the services in the inventory viewer. The GUI allows the operator to assign the correct owner of a service where the software could not determine one. It is also possible to merge two customers into one.

The service information is presented in a XML file to operator. Elements which may be changed by the operator are marked with appropriate attributes.

The operator confirmation step allows the operator to accept, modify or reject the updates planned to be executed.

For customers and services which names can not be identified from the data source, the operator may replace the auto generated names will more meaningful names (e.g. it may not be possible from an interface configuration to determine a meaningful name. The loader will in this case provide an auto generated name for the customer).

The outcome of the operator confirmation is XML file in the same format as the provided XML file.

It is left as a task for system integrators to provide tools and GUIs to display and process the Service Upload XML and generate the result XML file. Existing tools like Microsoft Excel can be used for processing the XML file.

The repository update step is done in two sub-steps. First step is importing the Service XML file after the operator confirmation. During the process new service-identifier and generate and the date conformance is verified by storing in temporarily oracle segment while validating that the data values have not been made inconsistent. Only if no error were detected, the data is committed.

The validation is to ensure that the operator has not made any inconsistencies of the data, e.g. the VPNs are owned by customer that doesn't exist or sites are member of VPNs that doesn't exists.

The "operator confirmation" step is optional and can be skipped during an initial service load or proof of concept scenario, if necessary. However, the services in the file can only be committed if ambiguities of the ownership of site and VPN have been resolved.

In the VPN SVP v7.0 the service upload functionality for updating the CRM tables is located on the HPSA server. As a consequence it is required to have access from the HPSA server to the CRM database. Later versions may support load of the CRM service using the Service XML file as input.

---

## 4 Configuring VPN SVP before start

### 4-1 Required Configuration Steps before start

Before service discovery process can be run, there is a set of configurations that must be completed first. Besides installation of the VPN SVP on top of HPSA, the VPN SVP must also be configured and the equipment and interfaces must be loaded.

The configuration of the VPN SVP is explained in the VPN SVP administrator's guide.

#### 4-1-1 VPN SVP Configuration

The following four set of parameters are required to be configured before the service discovery process can be run.

- 1. The ASN and the admin VPN must be configured:** During parsing and analysis of the information from the equipment configuration file, the Autonomous System Number is used to analyze Route Target values and BGP route distribution configuration. The administrative RT values are used during the analysis of the VPNs. The administrative Routing Community is not modeled as a customer VPN and the detected administrative RT values are therefore treated specially when found in the configuration files.
- 2. The IP address pools must be configured:** The VPN SVP is managing the IP address allocation and keeping track of the IP addresses which are in use in the network for the PE-CE attachment links. During the load phase, the service discovery process will determine which IP address pool the service belongs to and create the corresponding IP-NET object if needed. If an IP-Net object already exists in the VPN SVP repository, this object will just be marked as reserved during the commit phase.  
If an IP address pool does not match the IP address of the detected service in the network, the Service Discovery will not upload this particular service and an error is logged.
- 3. Rate limits used in the configuration files must be configured:** If QoS and ratelimit of a service in the network is configured compliant with the VPN SVP convention, the used ratelimit must defined in the VPN SVP repository before service discovery is started. Otherwise the QoS for this service will be marked as "Partial Compliant" or "Non Compliant" and only limited clean up of the QoS configuration for this service will be supported.
- 4. Interface Types:** The equipment upload process will only upload interfaces listed in "Interface Type" under CRModel →Parameter → Interface Type. If during Service Discovery, services are detected at termination points (interfaces) which have not been populated into the in the VPN SVP repository, the services will not be updated and an error is logged.

Consult the VPN SVP administrator's guide for detailed information on how to configure these parameters.

#### 4-1-2 Equipment population

Before service discovery is started, all routers and their interfaces must be populated into the VPN SVP repository. The service discovery software provides the "equipmentupload" tool which allows population of multiple routers and their interfaces based on a set of configuration files stored in a directory. Equipmentupload is only supported for Cisco and Juniper.

All the configuration files should be placed in configuration directory under a subfolder named against the vendor. For example: All the Cisco and Juniper configuration files should be placed in: <configuration directory>/cisco and <configuration directory>/juniper respectively. The routers will be named after the filename of the configuration file.

The equipmentupload tool can be run multiple times if required. The tool will only create a router object if it does not already exist in the VPN SVP repository. If the router is already in the repository, the objects and its attributes are left unchanged. In same way, only interfaces that do not exist on a given router in the inventory will be created. Interfaces which already exist are left unchanged. If a card with a set of interfaces has been removed from a router and the router is uploaded, the

removed interfaces will **not** be deleted in the repository. For safety reasons, it is left as a manual operation for the operator to delete the interfaces through inventory viewer.

It is still possible to create a new router object through the inventory viewer and perform an interface discovery when the equipment is accessible.

The command-line invocation for uploading the equipments using the Service Discovery tool is as below:

```
equipmentupload.sh -confdir <configuration directory> [-sessionid <id>] [-username <oracle username>] [-password <oracle password>] [-url <DB connection URL>] [-verbose]
```

Equipment load command line options:

Option syntax	Purpose
-confdir <configuration directory>	Specifies the directory which contains cisco/juniper sub-folder. All the cisco and juniper equipment configuration files should be placed in cisco and juniper sub-folder respectively. The filenames must comply to the following convention: <routername>.<extension>. Where <routername> must be unique and will become the name of the router object in the repository. The filenames must have the extension ".cfg", ".txt", ".cnf", ".config", ".shr", ".shrun" or ".conf". The extension list can be modified. Refer to section 5-2 <b>Other configurable properties</b> for more details.
[-sessionid <id>]	Session name for the equipment upload session. If sessionid option is not specified, then current timestamp will be used as session id by the equipment upload process. The log file for an equipment upload session will be of format "equipment-load<sessionid>.xml"
[-username <oracle username>]	Oracle user name to be used for the HPSA database connection. If the dbAccess.cfg is configured, it picks up the oracle username from this file.
[-password <oracle password>]	Oracle password to be used for the HPSA database connection. If the dbAccess.cfg is configured, it picks up the oracle password from this file.
[-url <URL>]	URL of HPSA database of format jdbc:oracle:thin:@<DB hostname>:<DB port number>:<DB SID> If the url is not specified, the value is retrieved from the mwfm.xml file
[-verbose]	Displays all the log messages

## 4-1-3 Manual equipment configuration steps

After the equipment has been loaded with the equipment upload tools, you must update a number of attribute on each router element. This is information which can not be retrieved from the configuration file and which is important for the VPN SVP to correctly manage the equipment.

You can find the routers which were created during the equipment upload in the inventory tree under the equipment view. In the equipment view, the routers are located under Region → Unknown → Discovered PE Routers. See an example in the

**Figure 4-1.**

The Equipment view in the inventory viewer shows the loaded router located in the "Unknown" Region under the "Discovered PE Routers" branch. Before service discovery is started the router must be updated with information about its location in the network topology and the type and version of the router. The Region is associated the containing Network, and will change if the containing NetworkID is changed.

Figure 4-1 Router uploaded by EquipmentUpload under Unknown region in Equipment tree.

The screenshot shows the HP Network Assistant interface. On the left, the 'Inventory' tree is expanded to 'SAVPN/Equipment' > 'Unknown' > 'Discovered PE routers', where 'ar1-bi0133' is selected. The main pane displays the 'View PERouter' details for this router.

Name	Value	Description
NetworkId	ISP Network	Network the NE belongs to
NetworkElementId *	100	Primary key
Name *	ar1-bi0133	Meaningful name of the device
Description	Uploaded Equipment	User information
Region	Provider	The region the PE belongs to
Location *	Unknown	Location of the device
Loopback IP	15.76.223.3	Primary IP address of the device
Management IP	15.76.223.3	IP address for management of the device
Management Protocol	telnet	
PWPolicyEnabled	No	True if this NE use a password policy to authenticate
PWPolicy		Name of the password policy
UsernameEnabled	No	True if username is used to authenticate management connection
Username		Username for management connection
Password	*****	Password for management connection
EnablePassword	*****	Password to enable device configuration
Vendor	Cisco	Vendor of device
OSVersion	Cisco-12.2 (32)	OS version of device
ElementType	C340024TSA	Type of device
BGPDDiscovery	No	NE supports VPLS BGP discovery (rfc 4761)
Tier		Tier levels for routers
SerialNumber	A65435sxx	Serial number of the device (inventory information)

The following set of attributes of may be required to be updated before the service discovery process can be started.

- **Loopback IP:** The address is used as the end point when setting up L2 VPN martini-wires.
- **Management IP:** This is the IP address HPSA will use when connecting to the equipment.
- **NetworkId:** Specifies the network and the region where the router is located. Once the attributes has be updated and the change committed, the router object will no longer be displayed under “Discovered PE Routers”, but instead you can locate it under the selected network.
- **Location:** The location in the region where the router is located
- **State:** By default set to down. It must be set to “Up”, before activation is allowed.
- **Elementtype:** The attributes OSversion, Vendor, Elementtype and Role are used to identify the activation dialogues used during activation. Select the appropriate value.
- **Backup:** Specifies if the router must be included in the scheduled backup.

- **PWPolicyEnabled** and **PWPolicy**: If password policy is enabled and defined.
- **Username**: Username for connecting to the router.
- **Password**: Authentication password
- **EnablePassword**: Enable password
- **ManagementProtocol**: By default it is telnet, but it could also be ssh.
- **OSversion**: The attributes OSversion, Vendor, Elementtype and Role are used to identify the activation dialogues used during activation. Select the appropriate value.
- **LifeCycleState**: By default it is set to "Planned". The state must be set to "Ready" or "Accessible" before activation can be initiated. If set to "Accessible", the router must be uploaded through the inventory GUI before it becomes ready.



---

# 5 Customizing Service Discovery

## 5-1 Modifying Patterns

### 5-1-1 Understanding the existing patterns

This section assumes that the reader has basic understanding of the regular expressions and groups. Parsing of the router configurations is done using the java regular expressions patterns and these patterns are described in the property files `serviceupload.properties`, `cisco_serviceupload.properties` and `juniper_serviceupload.properties` files placed in the directory `$SOLUTION/etc/config/service_upload`.

`serviceupload.properties` file described the patterns that are common to both Cisco and Juniper. While `cisco_serviceupload.properties` and `juniper_serviceupload.properties` describe vendor specific patterns. The following tables explain the patterns used for extracting service details from interface description and vrf description found in the configurations files. The patterns configured and the groups used for capturing the values are explained with the help of sample texts.

The patterns used for extracting service details can be broadly classified into two, based on the expected description string. If the string is in accordance with the VPN SVP conventions then one class of patterns is used. Otherwise a different class is used. The second class by default expects description to be in accordance with Cisco ISC convention. This set of patterns must be updated according to the convention used for the existing services in the provider's network.

If the description string matches the pattern shown in **Table 5-1**, the string is assumed to be following VPN SVP convention and details are extracted using the VPN SVP patterns, otherwise the Cisco ISC patterns are used.

**Table 5-1** VPN SVP String

---

**Pattern that decides if the vrf or interface description is VPN SVP compliant or not**

---

```
com.hp.ov.activator.vpn.serviceupload.VPN_DESCRIPTION= (\\*\\* (OV)? (HP)?SA VPN  
\\*\\*) (.*)
```

---

## 5-1-2 Matching interface descriptions

The following tables describes the different patterns used for matching the interface

**Table 5-2** VPN SVP interface description pattern

<b>Patten which extracts service details from VPN SVP compliant services interface description</b>
com.hp.ov.activator.vpn.serviceupload.VPN_IF_PATTERN = Added i/f, Customer\\(id\\):\\s(.*)\\((.*)\\).\\sSiteAttachment:\\s(.*)\\sSite name\\(id\\):\\s(.*)\\((.*)\\),\\sDate:(.*)
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_CUSTOMERNAME = 1
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_CUSTOMERID = 2
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_SERVICEID = 3
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_SITENAME = 4
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_SITEID = 5
com.hp.ov.activator.vpn.serviceupload.VPN_IF_DESC_DATE = 6

The patterns and group numbers are arranged adjacently in the properties file.

**Table 5-3** provides an example of the values captured by each group using the above patterns. The parsed string is:

*\*\* HPSA VPN \*\* Added i/f, Customer(id): A-Corp(1), SiteAttachment: 1022, Site name(id): ACorp-Budapest(1021), Date: 2007.09.19 20:39:15*

**Table 5-3** Sample values extracted by the VPN SVP interface description pattern

<b>Group name(number)</b>	<b>Description</b>	<b>Value</b>
VPN_IF_DESC_CUSTOMERNAM E ( 1)	Name of the customers who owns the service	A-Corp
VPN_IF_DESC_CUSTOMERID ( 2)	Id of the customer	1
VPN_IF_DESC_SERVICEID ( 3)	Service id for the attachment	1022
VPN_IF_DESC_SITENAME ( 4)	Name of the site service	ACorp-Budapest
VPN_IF_DESC_SITEID ( 5)	Id of the site service	1021
VPN_IF_DESC_DATE ( 6)	Date when the service is last modified or created	2007.09.19 20:39:15

**Table 5-4** lists the patterns used for extracting the service details from the interface description of a service configured by Cisco ISC.

**Table 5-4** Cisco ISC interface description pattern

Pattern which extracts the service details from Cisco ISC interface description
com.hp.ov.activator.vpn.serviceupload.ISC_IF_PATTERN =(.+), BW: (.*), CID: (.*)
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_CUSTOMERNAME = 1
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_SITENAME = 3
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_SERVICEID = 3

Values captured by each group for the above pattern is explained in **Table 5-5** . The reference interface description used for the values explained is

*BLUE-CHIP Corp, BW: 1984 Kbps, CID: MSAA000020967*

**Table 5-5** Sample Values Extracted by the Cisco ISC interface description pattern

Group name(number)	Description	Value
ISC_IF_DESC_CUSTOMERNAME (1)	Customer Name	BLUE-CHIP Corp
ISC_IF_DESC_SITENAME (3)	Name of the site service	MSAA000020967
ISC_IF_DESC_SERVICEID (3)	Service id	MSAA000020967

## 5-1-3 Pattern for matching VRF description

This section lists the patterns for matching VRF description strings.

**Table 5-6** VPN SVP VRF description pattern

Patten which extracts service details from VPN SVP compliant services VRF description
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_PATTERN= Added vrf, Customer id:\s(((\d)*\s*))\s*VPN name\((id\):((\s*(\S+)\((\S+)\)),*)\)\sDate:\s((\S+) (\S+))
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_DESC_CUSTOMERID_LIST = 1
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_DESC_VPN_LIST = 4
com.hp.ov.activator.vpn.serviceupload.VPN_LIST_PATTERN =((\S+)\((\S+)\)),*
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_DESC_LIST_VPNNAME = 1
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_DESC_LIST_VPNID = 2
com.hp.ov.activator.vpn.serviceupload.CUSTOMER_LIST_PATTERN =((\d+),*
com.hp.ov.activator.vpn.serviceupload.VPN_VRF_DESC_LIST_CUSTOMERID = 1

A sample VRF description for the service provisioned by VPN SVP is like this.

*\*\* HPSA VPN \*\* Added vrf, Customer id: 2, 1, VPN name(id): ACorp-VPN(1033), BCorp-VPN(1000), Date: 2007.09.19 20:52:14*

This vrf description gives list of VPNs and the list of customers that shares that particular vrf  
Note that sub patterns are used to extract the service details from list of entries

**Table 5-7** Sample Values Extracted by the VPN SVP vrf description pattern

Group name(number)	Description	value
VPN_VRF_DESC_CUSTOMERID_LIST	Customer ids listed under vrf description	2, 1,
VPN_VRF_DESC_VPN_LIST	VPN name and id list	ACorp-VPN(1033), BCorp-VPN(1000)
VPN_VRF_DESC_LIST_VPNNAME	One VPN name from list	ACorp-VPN and BCorp-VPN
VPN_VRF_DESC_LIST_CUSTOMERID	One VPN id from list	1033 and 1000

Pattern for extracting VRF name for Cisco, and extracting Route Distinguisher for Juniper is shown in **Table 5-8**

**Table 5-8** vrf name pattern

Pattern that extracts the vrf name vrf line for Cisco and RD for Juniper
com.hp.ov.activator.vpn.serviceupload.cisco.VRF = (^ip vrf\s(\S+)\$)
com.hp.ov.activator.vpn.serviceupload.cisco.VRF_NAME = 2
com.hp.ov.activator.vpn.serviceupload.juniper.VRF_ROUTE_DISTINGUISHER = (\s*route-distinguisher (.*)\$)
com.hp.ov.activator.vpn.serviceupload.juniper.VRF_ROUTE_DISTINGUISHER_GROUP=2

See the <vendor>\_serviceupload.properties files to find the other vendor specific configurable patterns.

## 5-1-4 Customizing the patterns

This section describes how to modify an existing pattern and group numbers if the configuration contains any text which is not as expected by the existing patterns. Consider the Cisco ISC interface description pattern explained in **Table 5-4** as an example. Consider that the interface description found in the Cisco ISC configuration file is as follows

*Service Name: BLUE-CHIP Corp, Rate Limit: 1984 Kbps, Service ID: MSAA000020967, CID: 1000*

To capture the service details from the above text, a new regular expression pattern has to be formed and the group numbers has to be updated accordingly. An example pattern and group for the above given sample text is shown in **Table 5-9**

**Table 5-9** Cisco ISC customized interface description pattern

<b>Pattern which extracts the service details from Cisco ISC interface description for the sample custom interface description</b>
com.hp.ov.activator.vpn.serviceupload.ISC_IF_PATTERN = Service Name: (*), Rate Limit: (*), Service ID: (*),\sCID: (*)
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_CUSTOMERNAME = 4
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_SITENAME = 1
com.hp.ov.activator.vpn.serviceupload.ISC_IF_DESC_SERVICEID = 3

There are different ways of forming a regular expression and above given is just an example. Similarly other patterns also can be updated in case the configuration file contains text of a different format than the one expected by the existing patterns. Refer the Properties file for more details about the existing patterns. Most of the properties names used are self explanatory.

## 5-2 Other configurable properties

Some of the properties of the service upload can be controlled by changing the configuration parameters in the file `$SOLUTION/etc/config/service_upload/upload.setenv`

**Table 5-10** Configurable properties in upload.setenv

<b>Property name</b>	<b>Description</b>	<b>Default value</b>
EXTENSIONS_LIST	List of valid extensions for the configuration files To upload all the files use '*' For example EXTENSIONS_LIST=* or EXTENSIONS_LIST=config,* will upload all the files in the directory . EXTENSIONS_LIST=config.txt will upload only the files with extensions .cfg and .txt	cfg,conf,shrun,txt,cnf,shr,config
LOGLEVEL	Log level for service upload	Log level is obtained from the value set in mwfm.xml for the vpn_log_manager module. The log level can also be set in upload.setenv to ERROR, WARNING, INFORMATIVE, DEBUG or DEBUG2, which takes precedence

## 5-3 Advanced Customization

Configuration of VPNs and QoS can be done in many different ways. In instances where the service provider is using conventions different from the VPN SVP model, it may be necessary to modify the code used in the VPN SVP discovery software. Often this will be part of the delivery project where the VPN SVP is customized for the specific solution. This may include changes to the inventory model, workflows and services. In these instances, service discovery must also be updated accordingly. Please contact the HPSA product group to get access to the source code for the service discovery.

Examples where code modification is required:

- **QoS convention:** If the service provider requires usage of another QoS model, the VPN SVP repository model may need to be modified. Even when the QoS model stays unchanged, it may be useful to update the parsing and analysis code for QoS to recognize important parameters from the deployed service. The examples can be rate-limits, classes of service. Analysis of QoS is done in AnalyseQoSProfiles method in the Analyse class.
- **Pairing of RT-values into Route Communities:** If the service provider have a special algorithm, which can be used to determine which RT-values belong together in a routing community, it can be helpful to implement this in the discovery of VPNs. See section 8-1 on how the route-target values are paired to routing communities. This is handled in findRCs and findRCType methods in the GetIdentifiers class.
- **Support for multiple Administrative VPN or Enterprise VPNs:** The VPN SVP will only recognize and model a single Administrative VPN. The service provider may also use special VPNs for Voice or internet access. These VPNs may not be of interest to model as traditional VPNs, but may be model as a sub-service for example. The special treatment of administrative VPNs in done in setManagedSite method in the Analyse class.
- **Parsing of route-maps, access-lists and prefix-lists:** The service discovery software does not recognize and parse route-maps, access-lists and prefix-lists. Currently they are considered as a function of the service provisioned service and not modeled in the repository. However router-maps configured for a VRF may contain important information needed to discovery the correct VPN topology.

---

# 6 Running the Service Discovery

## 6-1 Approaches

The service discovery tool is primarily a tool to be used when the VPN SVP is being deployed at a provider that has existing services which must be loaded into the VPN SVP.

Before starting the actual service discovery the VPN SVP must be configured. It is also recommended to run a few trial runs with some of the configuration files to understand if additional configuration or modified the pattern matching expression is required. A load trial may reveal that some investigation is required to understand the route target values for an enterprise VPN. It may also be that some services in the network can not be loaded as they are, and it must be considered if the services in the network should be modified before a service discovery is started or if extensions to the VPN SVP must be implemented.

In general service discovery can be run in two different ways:

- A bulk load of all existing services
- Upload of one or more routers after the VPN SVP has started managing the VPN services

The bulk load is the recommended usage of the service discovery tool. During the analysis phase route target values are paired into routing communities. The best result of the VPN topology is achieved when all VRF tables are loaded, meaning all router configurations are loaded together.

The load of a set of files after the services have already been committed can be useful to detect new services. However, it must be noted that existing services (VPN and Sites) in the VPN SVP repository will not be updated in the repository if there is a difference. Only new services will be added. This approach is useful if services have been added manually to the router. It should be noted that manually configuration of VPN services is not recommended and manually configuration can be the cause of error for any provisioning system.

The service discovery process must be run to an end before a loading a different set of files is started. It is for example not recommended to first take one set of files, load, compare and export them, before doing the same steps with a second set. Attempt to commit both set of export files may fail or leads to duplicate customers and services in the database.

It is always possible to start the service discovery process from the beginning, if something has failed. All data is temporary until the commit step has completed successfully.

The service discovery process can be done with no connectivity to the network. This allows that a delivery team can start on the service discovery process once the configuration has been received, before the solution is being deployed at the operator.

### 6-1-1 Moving Data from Test Server to Production Server

If the service discovery process is run on a system which is not the production server, the data must be moved from this system to the production server. The movements must be planned carefully to avoid inconsistencies in the production server inventory.

The simplest and safest approach is to populate the test server with the complete inventory to be loaded on the production server. Then, once the service discovery is complete, you will export the inventory from preparation server and load them onto the production server.

Alternatively, if all services are contained in the service.xml file generated during the export step, you may also use this file to populate the inventory on the production server. However this requires that you have confirmed that the file is consistent, i.e. you can commit it on your test server. It also requires that the all the equipment and termination points have the same ids on the production server as on the test server. This also includes QoSProfiles, IP pools, ASN, AdminVPN.

## 6-2 Command line options

The command-line invocation for uploading the equipments using the Service Discovery tool is as below:

```
serviceupload.sh -confdir <configuration directory> -load -compare -export [<xml file>] -commit <xml_file> [-sessionid <id>] [-noDBCleanup] [-username <oracle username>] [-password <oracle password>] [-url <DB connection URL>] [-all]
```

It is possible to run several steps in one session, e.g. `serviceupload.sh -load -confdir <dir> -compare`. However it is recommended to do a single step at the time and inspect the error and warning messages. Error steps must be resolved and understood before the next steps is executed.

Option syntax	Purpose
[-confdir <configuration directory>]	<p>Specifies the directory which contains cisco/juniper sub-folder. All the cisco and juniper equipment configuration files should be placed in cisco and juniper sub-folder respectively</p> <p>This option must be used with the <code>-load</code> option.</p> <p>The filenames must comply to the following convention: <code>&lt;routername&gt;.&lt;extension&gt;</code>. Where <code>&lt;routername&gt;</code> must be unique and will become the name of the router object in the repository. The filenames must have the extension <code>“.cfg”, “.txt”, “.cnf”, “.config”, “.shr”, “.shrun”</code> or <code>“.conf”</code>. The extension list can be modified. See 5-2 Other configurable properties for details.</p>
[-load]	<p>Runs the load phase in the service discovery. The equipment configuration files are parsed, the data is normalized, and the services are analysed before being stored in the shadow tables in the database. The configuration files are located in the directory specified with <code>-confdir</code>. This <code>-load</code> option should be used along with the <code>-confdir</code> option.</p>
[-compare]	<p>Runs the compare phase in the service discovery. During the phase, the discovered service information gets compared with the existing VPN SVP data.</p>
[-export <xml file>]	<p>Runs the export step in the reconciliation phase. The discovered service information is exported into a XML file for operator validation. This xml file is generated under <code>\$(SOLUTION)/var/service_upload</code> folder.</p>
[-commit <xml file>]	<p>Runs the commit step in the reconciliation phase. The xml file containing the service information is read and the VPN SVP and CRM repositories are updated.</p>
[-sessionid <id>]	<p>Specifies the session name for the equipment upload session. If <code>sessionid</code> option is not specified, the current timestamp will be used as session id by the equipment upload process. The name of the log file for an equipment upload session will be on the format <code>“equipment-load&lt;sessionid&gt;.xml”</code></p>
[-username <oracle username>]	<p>Oracle user name to be used for the HPSA database connection. If the <code>dbAccess.cfg</code> is configured, it picks up the oracle user name from this file.</p>
[-password <oracle password>]	<p>Oracle password to be used for the HPSA database connection. If the <code>dbAccess.cfg</code> is configured, it picks up the oracle password from this file.</p>
[-url <URL>]	<p>URL of HPSA database of format <code>jdbc:oracle:thin:@&lt;DB hostname&gt;:&lt;DB port number&gt;:&lt;DB SID&gt;</code></p> <p>If the url is not specified, the value is retrieved from the <code>mwfm.xml</code> file</p>
[-all]	<p>Runs load, compare, export and commit phases one after another assuming reconciliation is not required,</p> <p>This option must be used only for demo purposes</p>



## 7 Report Generation

The service discovery report generation tool generates a report for the services that is uploaded from the configuration file. This utility should be used after data has been uploaded in the shadow table and all the Services are marked appropriately. Marking of services is done during the “compare phase” of serviceupload.

Generation of report is done by analyzing the data that is uploaded in the shadow table and the logs that are generated by doing a service upload of the configuration file. All the data base queries that are used to generate report are stored in the file

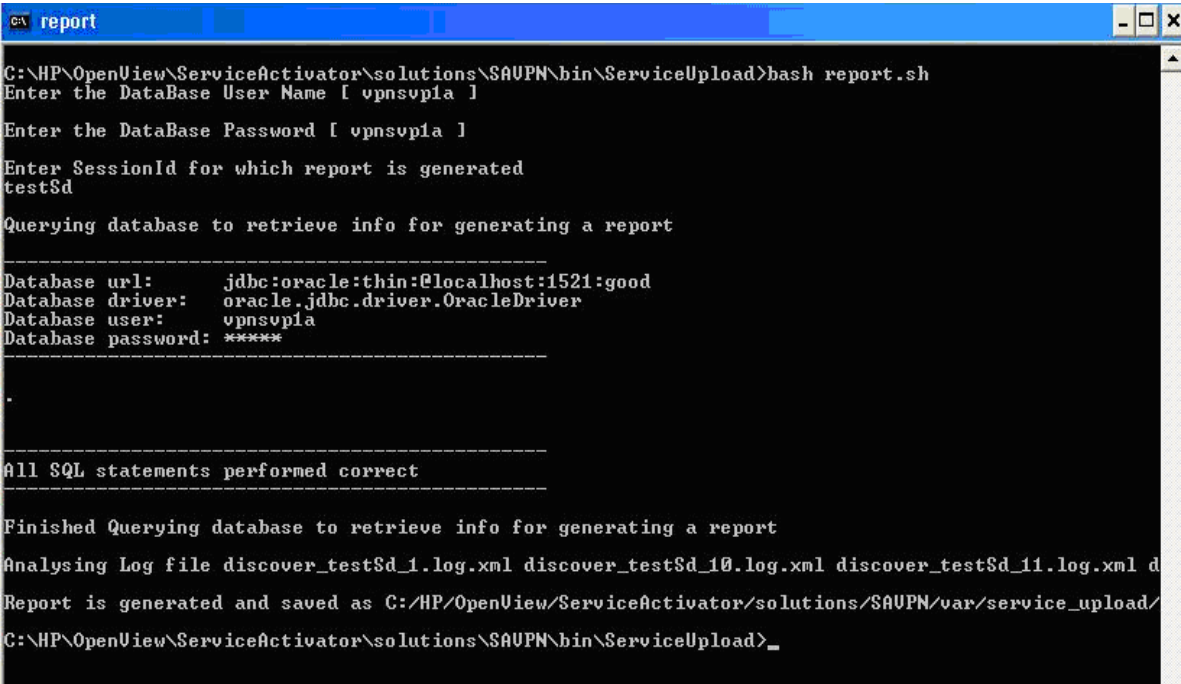
`$$SOLUTION/etc/config/service_upload/report/ServiceDiscovery.sql` and all the patterns that are used to analyse logs are stored in the file

`$$SOLUTION/config/service_upload/report/analyseSdLogs`. User can expand the report generation by adding more queries and pattern to these files.

The command-line invocation for generating the report is done by running `report.sh` placed under `$$SOLUTION/bin/ServiceUpload`. On executing, it will ask for the username, password and the sessionId to interact with the database and logs. By default `report.sh` looks for all the logs in HPSA log directory `$$ACTIVATOR_OPT/var/log/<hostname>`

Reports are generated and stored as a csv format: “`sdReport_<sessionID>.csv`” in `$$SOLUTION/var/service_upload/report`.

**Figure 7-1** Shows the generation of the report.



```
report
C:\HP\OpenView\ServiceActivator\solutions\SAUPN\bin\ServiceUpload>bash report.sh
Enter the DataBase User Name [ vpnsupla ]
Enter the DataBase Password [ vpnsupla ]
Enter SessionId for which report is generated
testSd
Querying database to retrieve info for generating a report
-----
Database url:      jdbc:oracle:thin:@localhost:1521:good
Database driver:  oracle.jdbc.driver.OracleDriver
Database user:    vpnsupla
Database password: *****
-----
All SQL statements performed correct
-----
Finished Querying database to retrieve info for generating a report
Analysing Log file discover_testSd_1.log.xml discover_testSd_10.log.xml discover_testSd_11.log.xml d
Report is generated and saved as C:/HP/OpenView/ServiceActivator/solutions/SAUPN/var/service_upload/
C:\HP\OpenView\ServiceActivator\solutions\SAUPN\bin\ServiceUpload>_
```

---

## 8 Parse and Analysis Details

This section explains how route-target values are paired into routing communities.

### 8-1 Rules for pairing route-target values into Routing Communities

The following rules in the listed order are applied when pairing route-target export and import values found in a VRF into routing communities.

If the VRF contains route-target export and route-target import values are found with same value, a mesh RC is assumed to be found. E.g route-target import 100:77 and route-target export 100:77 is paired into a mesh RC.

If the VRF contains an export and import values with one in difference, a hub and spoke RC is assume to be found. That is if  $\langle \text{ASN} \rangle : \langle \text{value} \rangle = \langle \text{ASN} \rangle : \langle \text{value} + 1 \rangle$ , where  $\langle \text{ASN} \rangle$  is typically the autonomous system number, but may also any other value like 100, 201, 6500, etc. E.g route-target export 100:26 and route-target import 100:25 are paired. If export value is smaller than the import, then the site is assumed to be a hub.

If the service provider has the convention for hubs that the rt-export is bigger than the rt-import value, then this is opposite from VPN SVP and it is required to change the convention in the VPN SVP

If a single RT export value and a single RT value have not been paired in the VRF, then the software assumes it to be a RC. In the example in the table below, route-target export 100:4 and route-target import 100:98 is paired into a hub-and-spoke RC after the route-target export 100:26 and route-target import 100:25 are paired.

**Table 8-1** Example of VRF table for Cisco

---

<b>VRF Configuration</b>
Ip vrf ExampleVPN
route-target export 100:26
route-target export 100:4
route-target import 100:25
route-target import 100:98

---

**Table 8-2** Example of VRF Configuration for Juniper

---

**VRF Configuration**

---

```
vrf_1000 {
  description "*** HPSA VPN ** Added vrf, Customer id: 1, VPN name(id): AllProtocolsVPN(1000),
  Date: 2009.07.17 12:49:52";
  instance-type vrf;
  interface fe-0/2/3.3001;
  interface ae0.2001;
  interface ae0.2002;
  interface ae1.0;
  route-distinguisher 12345:10020;
  vrf-import vrf_1000-import;
  vrf-export vrf_1000-export;
  routing-options {
    .....
    .....
  }
}
```

---

VRF route export policy:

```
policy-statement vrf_1000-export {
  term a {
    from protocol [ rip direct ospf static bgp ];
    then {
      community add 12345:10000;
    }
    accept;
  }
  term b {
    then reject;
  }
}
```

---

VRF route import policy:

```
policy-statement vrf_1000-import {
  term a {
    from {
      protocol bgp;
      community 12345:10000;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
```

---

---

## 9 Resolving Service Ownership

During the reconciliation phase, you must validate the data before it is committed to the VPN SVP and CRM repositories. The reconciliation phase is a phase that must be done with care. Once the data has been committed to the repositories there is no undo functionality, so it is essential that the data is correct. During the reconciliation phase

- You must approve all discovered service and customer data which should be committed
- You are required to resolve customer ambiguous ownership of services
- You may update the data with additional information, e.g. meaningful customer and site names

You have two steps where data can be updated during the Reconciliation step. See the olive green boxes in [Figure 3-1](#). In the “Resolved Customer Ownership”, you can through the inventory GUI update data. This is addressed in this chapter. After the “XML Export” step, you can validate all services in the service.xml file. This is addressed in the next chapter.

### 9-1 Required operator interaction

In network equipment configuration files it is possible to discover interfaces and the type of services configured. Information about the customer owing the service is rarely stored in the configuration file and for this reason it may not be possible to determine who the owner of a service is and if the same customer is the owner of several VPN services in the network.

If the service discovery software can not determine the customer owning the service, the operator is required to provide this information. The software may not be able to identify that the owner of two VPNs may be the same. This is only possibly if some customer information is stored in the description fields or in the used names for the VRF object. The operator may therefore also need to merge customers into one customer.

#### 9-1-1 Cause of Ambiguity in Ownership

When the Service Discovery software is discovering VPN services, it is parsing the configuration information for the termination point (interface) and the VRF associated with the interface.

In the simple example below configuration from a Cisco router is displayed

**Table 9-1** Cisco Configuration Sample for a L3VPN interface

---

```
interface Ethernet1/0.2001
  encapsulation dot1Q 2001
  ip vrf forwarding vrf_PE-CE_Default_1176
  ip address 172.17.0.25 255.255.255.252
  no snmp trap link-status
  no cdp enable
  service-policy input I3_simple_0.0.0.0.100_in_1M
  service-policy output I3_simple_0.0.0.0.100_out_1M
```

---

For each interface with a VRF associated the service discovery software will create a site.

**Table 9-2** Cisco Configuration Sample for a VRF

```
ip vrf vrf_PE-CE_Default_1176
rd 12345:11770
route-target export 12345:11760
route-target import 12345:11760
```

The VRF associated with interface is analyzed and the used route target values are matched into routing communities. If no description fields containing the customer, VPN or site information have been provided for the interface or the VRF, this information will be auto generated.

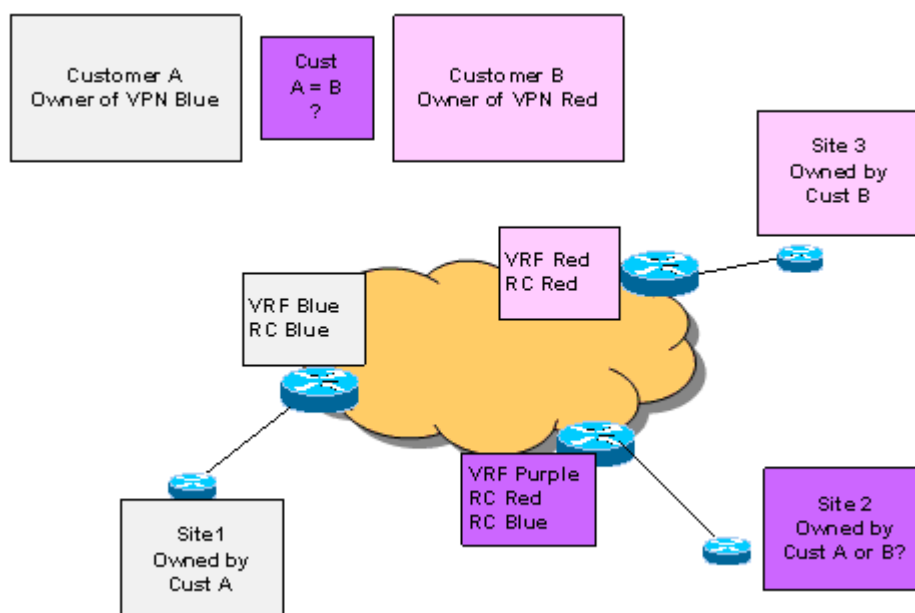
For the configuration sample provided in **Table 9-2** no customer information about the VPN can be deduced. The auto generate customer id and name and VPN service id and name is then based on the router-target values. The values generated from the samples are listed in the tables below

**Table 9-3** Autogenerated values for Customer and VPN objects

Attribute Name	Attribute Value
Customer Identifier	<RTimp>##<RTexp> e.g.:12345:11760##12345:11760
Customer Name	<RTimp>##<RTexp> e.g.:12345:11760##12345:11760
VPN Service Id	<RTimp>##<RTexp> e.g.:12345:11760##12345:11760
VPN Name	<RTimp>##<RTexp> e.g.:12345:11760##12345:11760
Site Name	<hostname>,<InterfaceName> e.g.: C3600_1:Ethernet1/0.2001
Site Service ID	<next_serviceid[<next_serviceid>] e.g: <next_serviceid[1977]>

If descriptions field in the interface configuration or in the VRF configuration contains customer or service names and id, these may be used by the service discovery software and provide more complete information.

**Figure 9-1** Figure shows the service ownership which must be resolved manually. The Service Discovery software may not be able to determine automatically if Site2 is owned by customer A or customer B. The software can not determine if Customer A and Customer B is in fact the same customer.



If a site is associated a VRF containing Routing Communities values from different VPNs, the software can not determine which VPN-owner is also the owner of the site. For safety reasons the software does not assume that all RCs (VPNs) in a VRF belong to the same customer. The service provider may for example provide an enterprise service, like VoIP, where sites must be joined into

the VoiP-VPN in order to receive the service. If all sites associated with the same RCs were assumed to belong to a single customer, the results of the analysis could then be just a single customers (the provider itself) owning all sites.

In **Figure 9-2** site 2 is a member of both red and blue VPN. Instead of assuming that the site is owned by customer A or B, a temporary customer “A;B” is created and display as the owner of Site2. During the reconciliation phase, you must resolve this ambiguity and state which of the customers the owner is. The software will not allow any commitment of data before the ambiguity has been resolved.

**Table 9-4** Auto generated values for the temporary Customer owning the resolved site

Attribute Name	Attribute Value
Customer Name	customer:<VPN-A RTimp>##< VPN-A RTextp>< VPN-B RTimp>##< VPN-B RTextp>; e.g. customer:12345:11760##12345:11760;12345:11760##12345:11760
Customer ID	12345:11760##12345:11760;12345:11760##12345:11760
Site Name	<hostname>,<InterfaceName> e.g.: C3600_1:Ethernet1/0.2001

## 9-1-2 Automatic Resolving of Ambiguities

During the compare existing customer and services are identified and marked as existing in the repository. Base on the existing customer and service information it is possible to resolve some of the ambiguities.

If we in **Figure 9-2** have found that Customer A and Customer B are existing and is the same customer, then the software will during the compare phase also conclude that site 2 is owned by this customer and the ownership of the site does not need to resolved. Further, the customer A and customer B will be merged into a new customer in the shadow tables with the correct customer id and name as found in the VNP SVP repository. All services identified as being owned by this customer will be displayed under this customer in the inventory view for the service discovery.

## 9-1-3 Manual Resolving of Ambiguities

The first step in the Reconcile phase is resolving customer ownerships of services (see **Figure 3-1**). After successful completion of compare phase of service discovery, the operator has the option to resolve the services where the ownership is ambiguous and to merge two customers into one. The inventory viewer can help you with this.


The Service Discovery view in the Inventory Viewer allows:

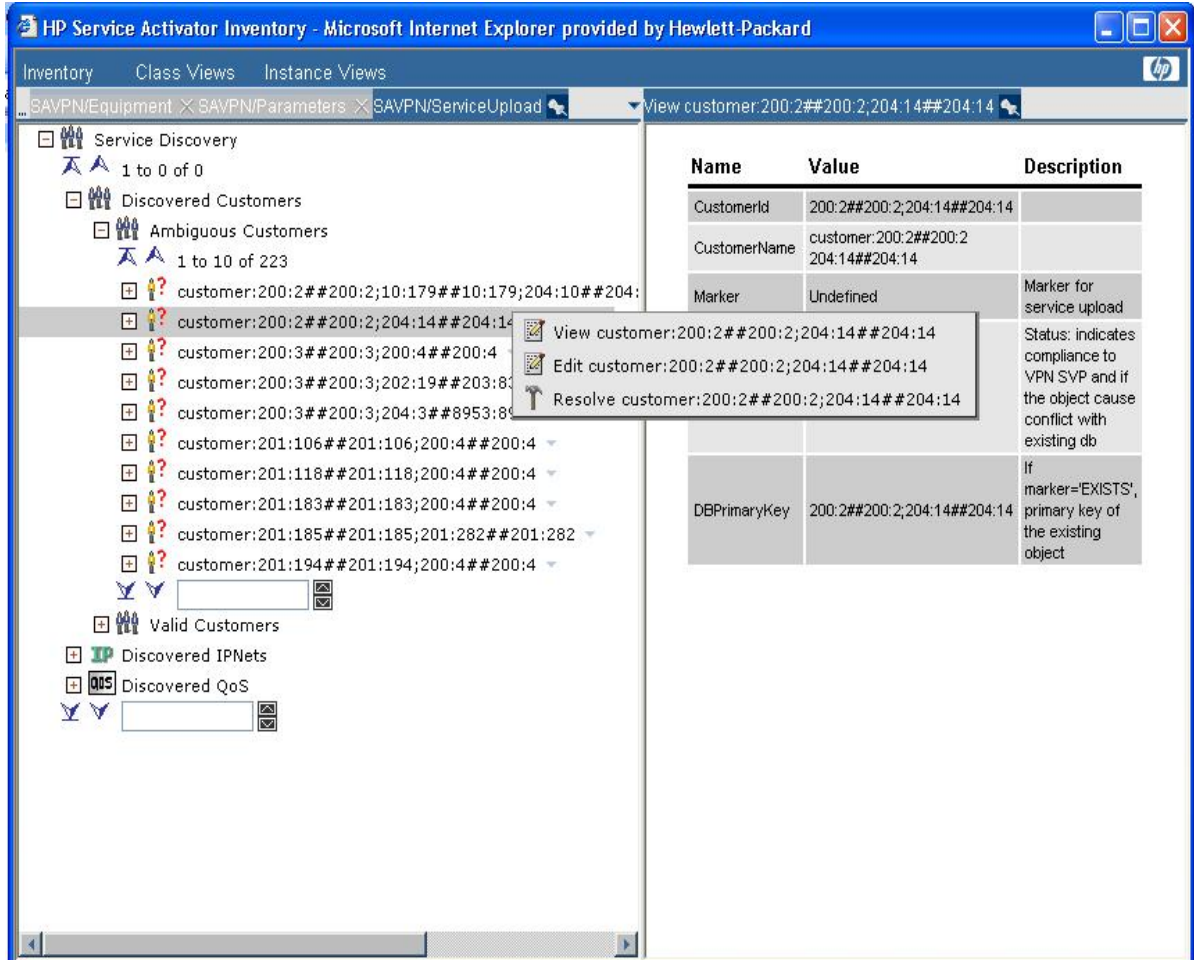
- Resolving the ownership of a service. This can both be a site and a VPN service
- Merge two customers into one
- Updating the customer, site and VPN names of an object
- Specify the CE IP address for the PE-CE link where this could not be determined

Resolving ownership of all services is required before any data can be committed into the VNP SVP repository.

## 9-1-3-1 Understanding the Service Discovery View and Icons

The discovered services are displayed in the HPSA inventory viewer under the “ServiceUpload” tab (see **Figure 9-2**). The “ServiceUpload” tab will be displayed only for users with admin privileges.

**Figure 9-2:** The figure shows an example of the ServiceUpload view. In this view the operator can resolve ownerships of sites with ambiguous owners listed under the “Ambiguous Customers” branch by pressing the icon .



Name	Value	Description
CustomerId	200:2##200:2;204:14##204:14	
CustomerName	customer:200:2##200:2 204:14##204:14	
Marker	Undefined	Marker for service upload
		Status: indicates compliance to VPN SVP and if the object cause conflict with existing db
DBPrimaryKey	200:2##200:2;204:14##204:14	If marker='EXISTS', primary key of the existing object

The service discovery view presents the discovered customers, the IP-NET addresses used by services and the QoS objects used by the discovery services.

The “Discovered Customers” contains two branches: Ambiguous Customers and Valid customers. The ambiguous customers are not actually customers, but place holders for services where the customer ownership can not be determined.

Resolved IPNet contains the IP network addresses for the PE-CE subnets. IP nets where the CE IP Address could not be determined are listed under the “Unresolved IPNets”

The following table provides the semantic meaning on the icons used in the Service Discovery view in the Inventory Viewer

**Table 9-5** Semantic meaning of icons used in the service discovery view

Icon	Type	Description
	Object Icon	The customer which already exists in the VPN SVP repository
	Object Icon	A new customer discovered in the network
	Object Icon	A temporary customer which is used as a placeholder for services where the correct customer ownership could not be determined.
	Object Icon	A new customer discovered in the network and which is a candidate to be merged with another customer
	Object Icon	A VPN which already exists in the VPN SVP repository
	Object Icon	A new VPN discovered in the network
	Object Icon	A VPN service with an error. This service will not be committed.
	Object Icon	A site which already exists in the VPN SVP repository
	Object Icon	A new site discovered in the network
	Object Icon	A Site service with an error. This service will not be committed. The used l3siteconnection may have an error.
	Object Icon	A VRF with an error. The VRF will not be committed. The VRF may be defined with multiple routers with different RT values.
	Object Icon	A RC with an error. The VRF will not be committed. The RC object may be associated a VRF which has already been committed.
	Object Icon	An error was encountered during service discovery for this connection. The service associated this connection will not be committed. Possible reasons for the error can be: undefined IP-addresses, unknown termination point on router, referencing VRF has error.
	Action Icon	Action to resolve the ownership of a service or a sets of services
	Action Icon	Action to merge two customers into one.
	Action icon	Action to edit object, e.g. update customer name, site name, VPN name, CE IP address

### 9-1-3-2 Resolving Ambiguous Customer Ownership of Sites

Before any services can be committed into the VPN SVP repository it is required that all the service with ambiguous ownership has been resolved. This is most simply done through the inventory GUI, but it can also be done through modifications in the service.xml file. If the ownership of the services has not been resolved, the final commit step will reject the service.xml file.

The customers listed under “Ambiguous Customers” are not valid customer, but temporary customers used as place holder of the ambiguous services. The ownership of the services listed under these customers must be resolved to one in a list of customers. The operator needs to resolve the ambiguity.

The operator can resolve the ambiguity for a single site at the time or for all sites listed on the ambiguous customer (see section 9-1-3-3 Resolving Ambiguous Customer Ownership of VPNs) by selecting the correct customer for a site; this moves the site to selected customer.

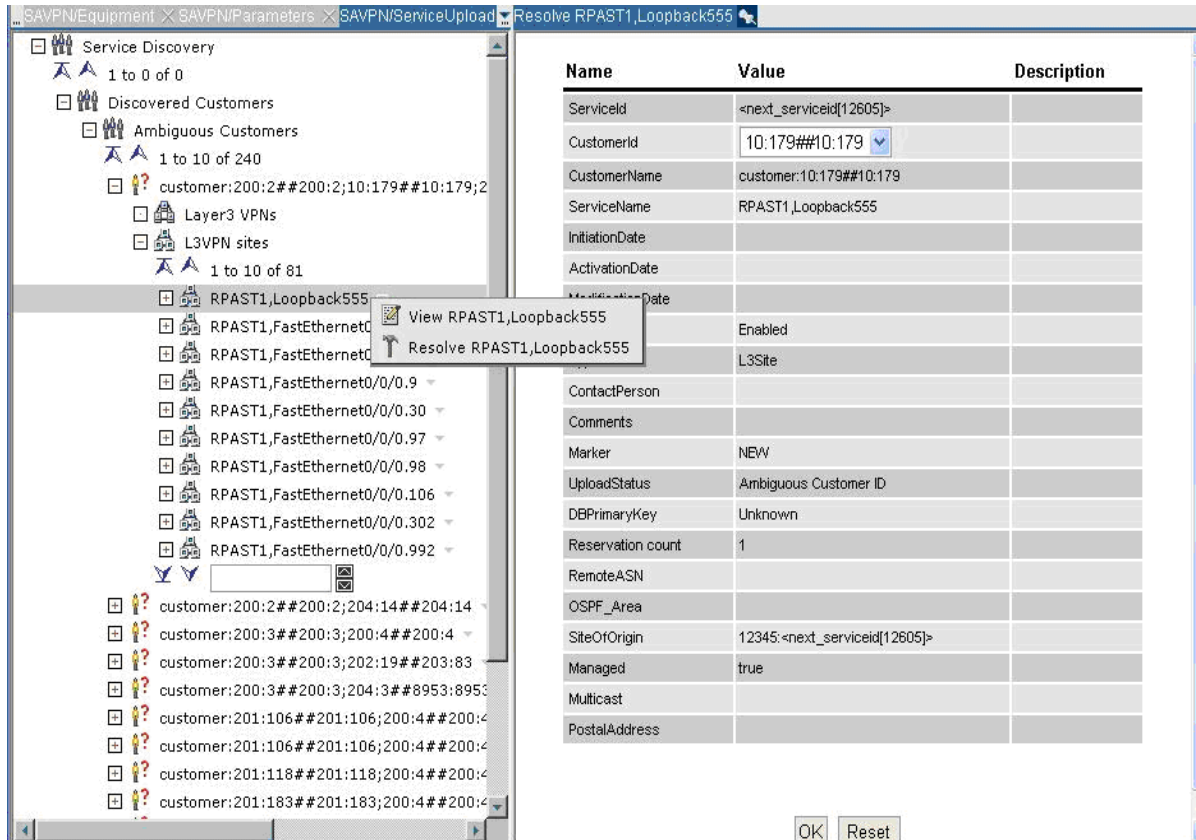
To resolve the ambiguity of a site, you should in the inventory viewer:

- Select the *ServiceUpload* menu and expand the *Service Discovery* → *Discovered Customers* → *Ambiguous Customers* branch.
- Locate your customer and expand its branch.



- Right Click on the *Ambiguous L3VPN site* and select *Resolve* option. It will display *Resolve Site* page on the right hand side.
- Possible customers to whom this ambiguous site may belong to are displayed in the “CustomerId” drop down. Select the correct customer from the “*Customer ID*” drop down to select the valid customer to whom the selected site can be moved to and click on submit.

**Figure 9-3** In the below example, clicking submit moves the selected service “<next\_serviceid[12605]>” to the selected customer “10:179##10:179”.



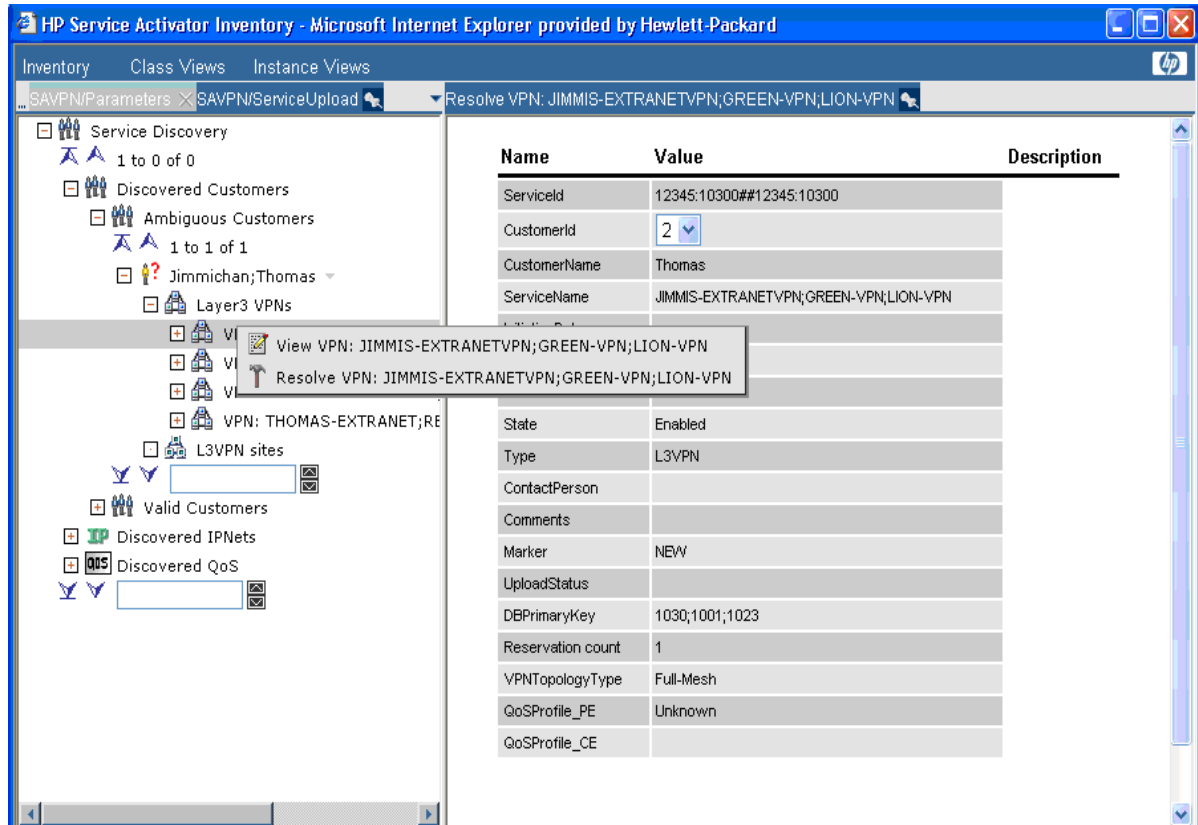
### 9-1-3-3 Resolving Ambiguous Customer Ownership of VPNs

The operator can resolve the ambiguity of a VPN by selecting the correct customer for a VPN; this moves the VPN service to selected customer.

To resolve the ambiguity of a VPN, you should in the inventory viewer:

- Select the *ServiceUpload* menu and expand the *Service Discovery* → *Discovered Customers* → *Ambiguous Customers* branch.
- Locate your customer and expand its branch.
- Right click on the *Layer3 VPN* that need to be resolved.
  - Select *Resolve* option. It will display *Resolve VPN* page on the right hand side.
  - Possible customers to whom this ambiguous VPN may belong to are displayed in the “CustomerId” drop down. Select the correct customer from the “*Customer ID*” drop down to select the valid customer to whom the selected VPN can be moved to and click on submit.

**Figure 9-4** In the below example, clicking submit moves the selected VPN “JIMMIS-EXTRANETVPN;GREEN-VPN;LION-VPN” to the selected customer “1”.



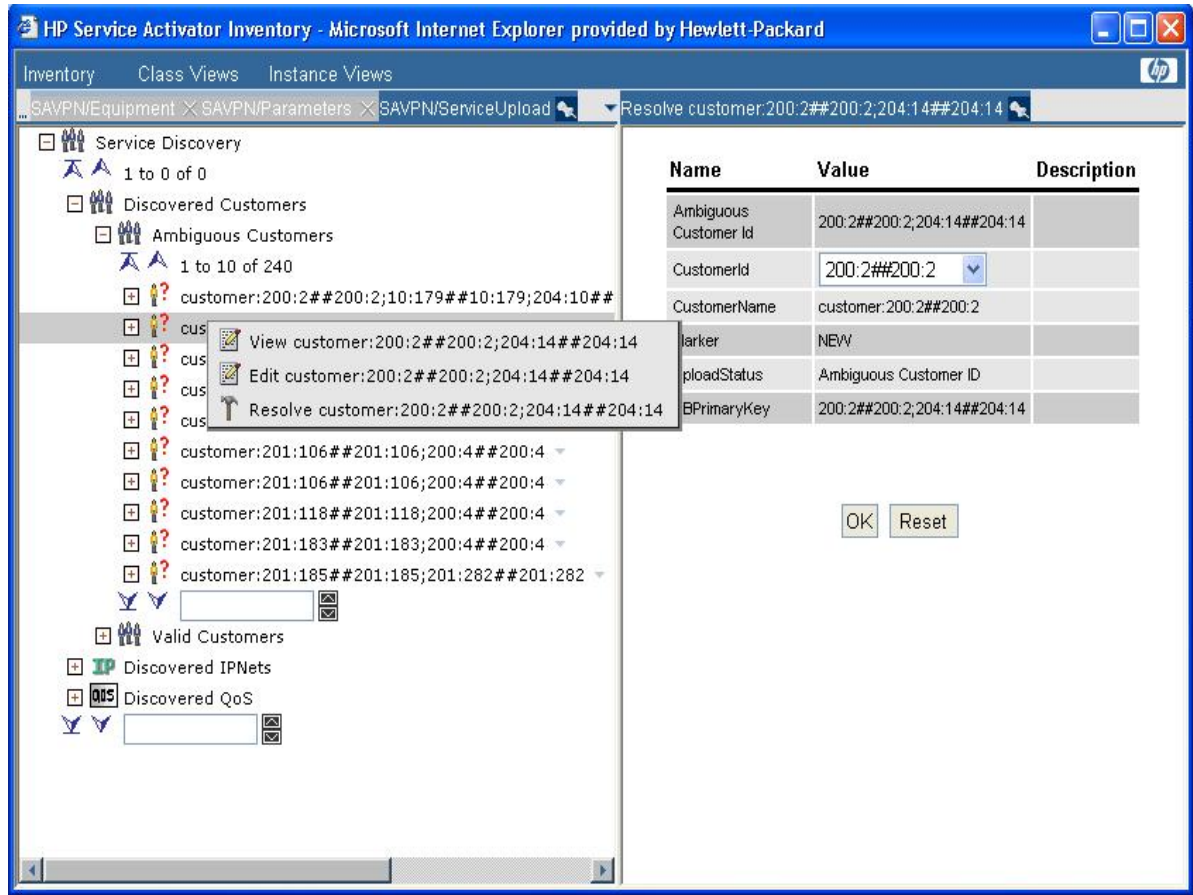
#### 9-1-3-4 Resolving Ownership of a Set of Services in one Operation

The other option is to resolve the ambiguity at the customer level. This option moves all the ambiguous sites listed under the customer to the selected valid customer.

To resolve the ambiguity of sites at customer level, you should:

- Select the *ServiceUpload* menu and expand the *Service Discovery* → *Discovered Customers* → *Ambiguous Customers* branch.
- Select the Ambiguous Customer that needs to be resolved.
- Right Click on the *Customer* and select *Resolve* option. It will display *Resolve Customer* page on the right hand side.
- Select the correct customer from the “Customer ID” drop down to select the valid customer to whom the services can be moved to and click on submit.

**Figure 9-5** In the below example, clicking submit moves all the services listed under the ambiguous customer ID “200:2##200:2;204:14##204:14” to the selected valid customer “200:2##200:2”.



The resolved services resolved are updated in the HPSA inventory. All the ambiguous customers listed in the “ServiceUpload” tab need to be resolved before proceeding with export. Services in the exported XML file can only be committed if all the ambiguities are resolved.

### 9-1-3-5 Merging customers

The operator can also merge services of one valid customer into another valid customer if required. In **Figure 9-2** customer A and Customer B may be the same customer and customer A and B must in this case be merged to prevent having two instances of the same customer in the VPN SVP repository.

If a customer object icon is two yellow customers (👤👤) instead of a single customer (👤), this indicates that the service discovery software has identified this customer as a likely candidate for merge. This information is based on discovery of a site which is member of two or more VPNs with two different owners. In **Figure 9-2** customer A and B will be listed with (👤👤) because Site 2 is member of both VPN Blue and Red. The operator must determine, if these customers are in fact two customer (an extra-net configuration) or a single customer (the customer owns more than one VPN).

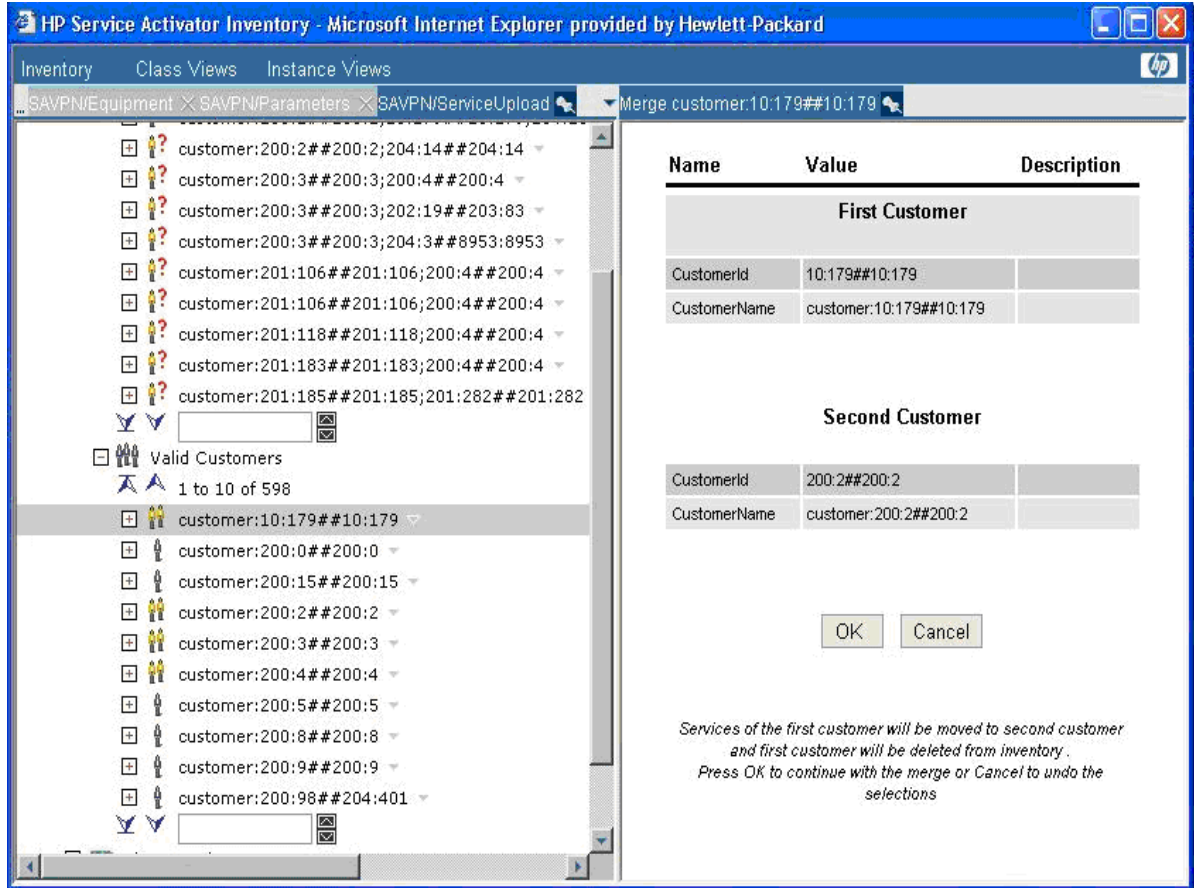
This can be done from the inventory GUI. To merge the services of one valid customer to another valid customer, you should:

- Select the *ServiceUpload* menu and expand the *Service Discovery* → *Discovered Customers* → *Valid Customers* branch.
- Select the customer whose services need to be moved to another valid customer.
- Right Click and select the *Merge Customer* option to display the “Merge Customer” page. Instruction to “Select another customer to continue...” will be displayed on this page.
- Select another valid customer to whom the services should be moved to by clicking the Merge Customer corresponding to that customer. Both the customers will be now displayed on the

“Merge Customer” page with message “Services of the first customer will be moved to second customer and first customer will be deleted from inventory. Press OK to continue with the merge or Cancel to undo the selections”.

- Click on “OK” to merge the services from first customer to the second customer.

**Figure 9-6** In the below example, clicking on “Ok” merges all the services in customer “650” to the customer “503” and the customer “650” is deleted from the inventory.



---

# 10 Operator Validation of Data

After the ownership of the services have been resolved, you must export the service data. The correctness of the data must be validated before it is committed. The data can also be updated with information which was not retrieved from the network. Data which is not desired to commit can also be removed here.

## 10-1 Understanding the XML service file

The service XML file contains all the service objects found during the discovery. Each object is marked with information if it is a new objects, existing or if the object has an error.

The service XML is used to update the services before they are committed to the VPN SVP repository. This can be information such as customer, site and VPN names, but it can also be more advance updates where services are deleted or added.

It is the Service Operators responsibility to verify that the data is correct, before the services are committed. On a production system it is highly recommended you make a backup of the repository before the commit step is executed.

The service objects are organized in a containment hierarchy which is described in [Figure 10-1](#). The relationship between the objects is shown in [Figure 10-2](#). The attributes for each element are described at the end of this chapter.

### 10-1-1 Understanding the XML attributes

**Table 10-1** Service XML Attributes

Attribute Name	Description
Marker	Can be "NEW", "EXISTS", "ERROR" and "IGNORE"
DBPrimaryKey	Is "Undefined" if it is a new object. If the object exists, the value is the primary key for the object in the VPN SVP repository.
UploadStatus	Is an informative description for this object. E.g for a L3VPN hub and spoke service, the message can be "no spokes found". This means that the topology is not complete and the VPN is not in use.

In the example below, a new customer has been found with a new VPN. During commit the customerid will be replaced with a correct customer id generated by the VPN SVP. The same will happen for the service id.

```

<Customer Marker="NEW" DBPrimaryKey="Undefined" >
  <CustomerId>6500:10836##6500:10836</CustomerId>
  <CustomerName>ACME INC</CustomerName>
  <VPNs>
    <L3VPN Marker="NEW" DBPrimaryKey="null" >
      <ServiceId>6500:10836##6500:10836</ServiceId>
      <CustomerId>6500:10836##6500:10836</CustomerId>

```

In the example below, an existing customer with an existing VPN is found. The customer id in the VPN SVP is "22". The existing service id for the VPN is "1275"

```

<Customer Marker="EXISTS" DBPrimaryKey="22" >
  <CustomerId>22</CustomerId>
  <CustomerName>customer:6503:10000##6503:10000</CustomerName>
  <VPNs>
    <L3VPN Marker="EXISTS" UploadStatus="no spokes found"
DBPrimaryKey="1275" >
      <ServiceId>6503:10000##6503:10001</ServiceId>
      <CustomerId>22</CustomerId>

```

## 10-1-2 Understanding the Service Identifiers

On successful export of the service XML file, the operator has to audit the XML file for correctness of data. The operator has to do manual audit of lines with customer ID, service ID, VPN ID and RC name.

For customer ID validation, search in the export XML file for all occurrences of "CustomerId". The value of "CustomerId" attribute should not have ambiguous customer ID of the pattern "<customer ID1>;<customer ID2>". Those ambiguous customers must be resolved before commit.

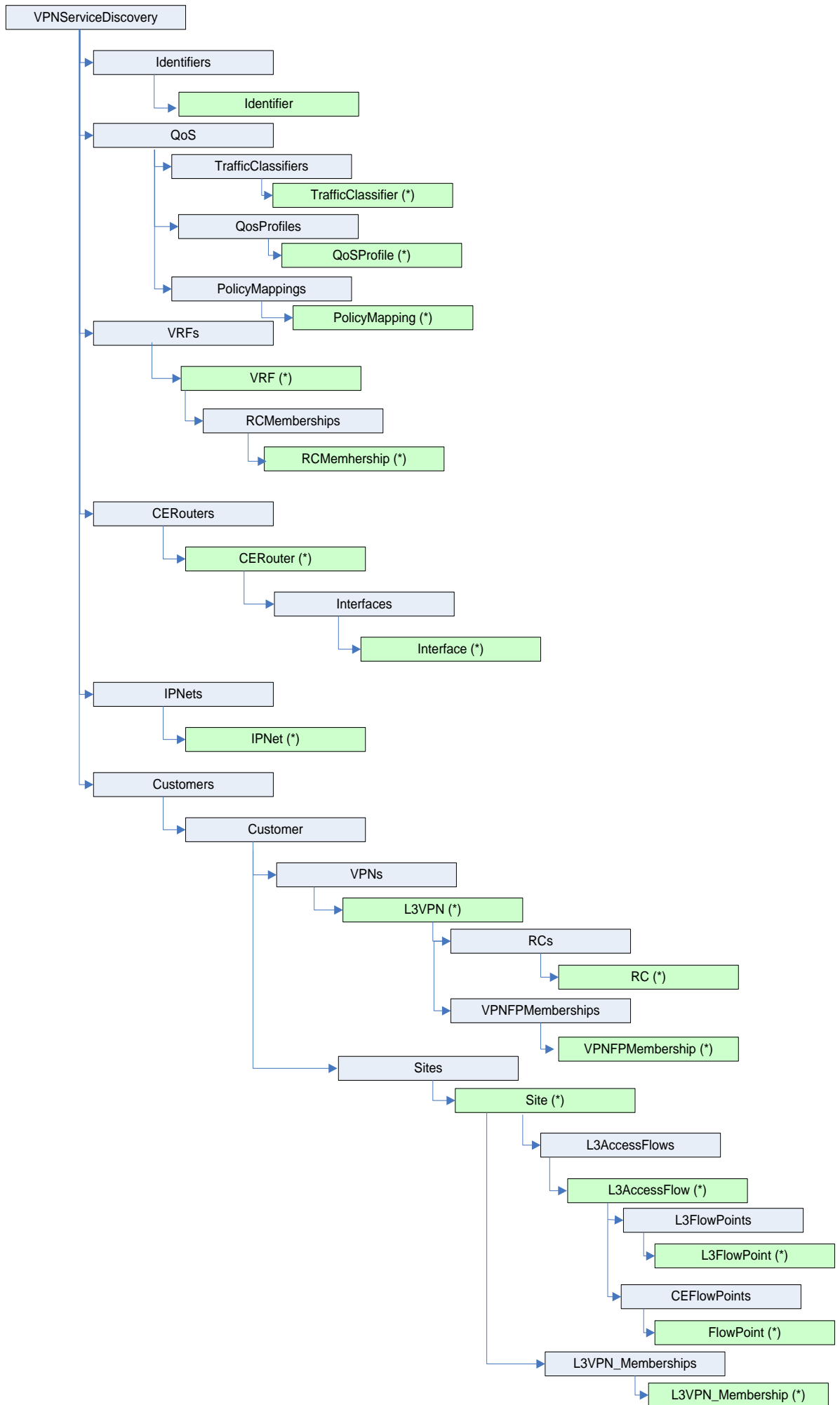
For service ID validation, search in the export XML file for all occurrences of "ServiceId". The value of "ServiceId" attribute should not have ambiguous service ID of the pattern "<service ID1>;<service ID2>". Those ambiguous services should be resolved before commit. Service IDs with the pattern "&lt;next\_serviceid[<num>]&gt;" are valid and should not be edited.

For VPN ID validation, search in the export XML file for all occurrences of "VPNId". The value of "VPNId" attribute should not have ambiguous VPN ID of the pattern "<VPN ID1>;<VPN ID2>". Those ambiguous VPNs should be resolved before commit.

The customer names are auto generated (customer:<customer ID>) by the service discovery tool when it is not possible to determine meaningful name from the interface configuration. To replace the auto generated names with more meaningful names, operator can use system integrator provided tools and GUIs to display and process the Service Upload XML and generate the result XML file. Existing tools like Microsoft Excel can be used for processing the XML file. The result XML file should be in the same format as the service export XML file.

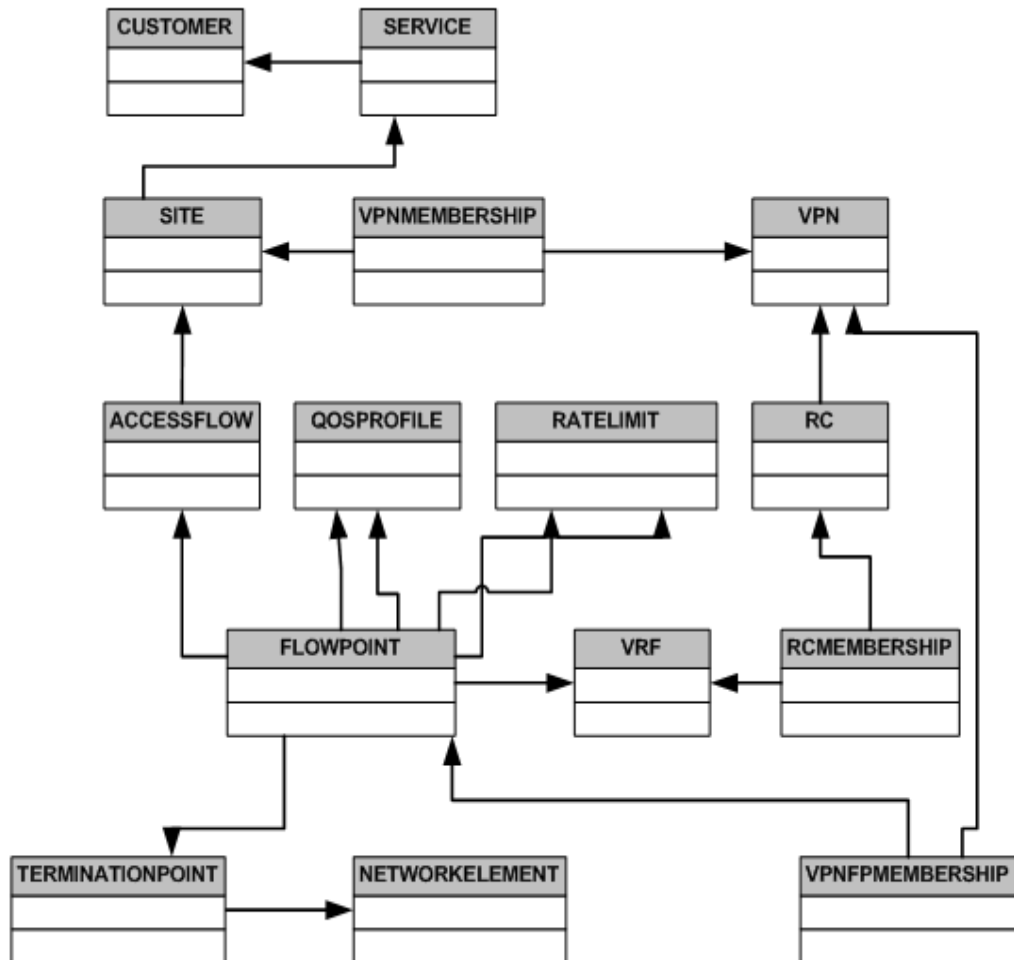
## 10-1-3 Service Data Hierarchy

**Figure 10-1** : The figure provides a graphical representation of the Service XML file. The XML objects are organized in a containment hierarchy similar to the Service view in the inventory viewer. Elements in green are object in the database. Element in blue are elements tags used to organize the data structure.



## 10-1-4 Service Object Relationships

**Figure 10-2** the figure shows the relationship between the objects loaded with the service discovery software.



## 10-1-5 Identifier

**Table 10-2** Identifier stores the maximum service and customer identifiers in the shadow tables used for service upload.

Attribute Name	Description
Max_ServiceId	Maximum service identifier found on the shadow tables
Max_CustomerId	Maximum customer identifier found on the shadow tables



## 10-1-6 Traffic Classifier

**Table 10-3** TrafficClassifier defines the CoS, layer and degree of compliance of the traffic class.

Attribute Name	Description
Name	Name of the traffic classifier
CustomerId	The id of customer-owner
DSCPs	List of DSCP bits delimited by comma
Filter	List of addresses filters delimited by comma.
CoSs	List of IEEE 802.1 bits delimited by comma
Layer	Layer of Traffic class (2 or 3)
Compliant	States the degree of compliance

## 10-1-7 Policy Mapping

**Table 10-4** PolicyMapping defines the combination of QoS and Traffic Class that form the various policy mappings.

Attribute Name	Description
TClassName	The name of Traffic Class
ProfileName	The name of QoS profile the mapping belongs to
Exp	MPLS EXP value for remarking selected traffic class
Dscp	DSCP value for marking CE traffic classes
Percentage	The part of bandwidth in the site's link
Position	The position in provider class table
PLP	Loss priority bit.
QueueName	Name of Queue/Forwarding class.
CoSName	Name of Class of Service

## 10-1-8 QoS Profile

**Table 10-5** QoSProfile defines the QoS, layer and degree of compliance of the profile.

Attribute Name	Description
QoSProfileName	The name of Profile
CustomerId	The id of customer-owner
Prefix	The prefix of the profile
Description	The description of profile
Profilename_in	Name of the service policy input
Profilename_out	Name of the service policy output
Layer	Layer of Traffic class 2 or 3
PEQoSProfileName	PE QoS profile name in case of CE based QoS
Compliant	States the degree of compliance

## 10-1-9 VRF

**Table 10-6** VRF definition

Attribute Name	Description
VRFName	VRF Name
RD	Route distinguisher
NE_ID	Networkelement identifier.

## 10-1-10 CE Router

**Table 10-7** CERouter defines the CE router configuration.

Attribute Name	Description
NetworkElementID	Network Element ID
Name	Network Element Name
IP	Loopback IP address of Network Element
management_IP	Management IP address of Network Element
Description	Network Element Description
Location	Location of Network Element
State	Network Element State
NetworkID	Network id
ElementType	Type of Element
Vendor	Equipment manufacturer
Username	User Name
UsernameEnabled	Username authentication enabled on router
Password	Router password
EnablePassword	Enable Password
ManagementInterface	Protocol used for connecting to Management port
OSVersion	Version of OS
Role	Role for router (e.g. PE, CE or P)
LifeCycleState	Life Cycle State
ROCommunity	SNMP Read-Only Community String
RWCommunity	SNMP Read-Write Community String
SerialNumber	Equipment serial number
SchPolicyName	Equipment scheduling policy for backup
Backup	Backup the configuration of this equipment?
Managed	Is the router Managed by the ISP (True/False)
Present	Is the router present (True/False)
CE_LoopbackPool	CE Loopback IP address

## 10-1-11 Interface

**Table 10-8** Interface defines the router interface details.

<b>Attribute Name</b>	<b>Description</b>
TerminationPointID	Termination point ID
Name	Name of the termination point
NE_ID	Parent Network Element ID
EC_ID	Element Component ID
State	State of Termination Point
Type	Type of Interface
ParentIf	Pointer to Parent interface if subinterface
IPAddr	IP address assigned to Interface
SubType	Details usage of the interface
Encapsulation	Encapsulation type
ifIndex	Unique number that identifies each interface for SNMP identification of that interface
ActivationState	Activation State of Interface
UsageState	Usage state of interface. Available, uplink or reserved
VlanId	VLAN Id used for the interface
DLCI	Data link connection identifier
Timeslots	Timeslots (0..31)
NumberOFSlots	Number of timeslots
Bandwidth	Bandwidth of interface
LmiType	CISCO-ANSI-CCITT type LMI
IntfType	FR DTE/DCE/NNI interface
BundleKey	Identification of bundle
BundleId	Link between PE/CE side of bundle

## 10-1-12 IP Net

**Table 10-9** IPNet defines the IP Net details.

Attribute Name	Description
IPNetAddr	IP Address Of The Net Link
PE1_IPAddr	IP Address Of The First PE Router Interface
CE1_IPAddr	IP Address Of The First CE Router Interface
PE2_IPAddr	IP Address Of The Second PE Router Interface
CE2_IPAddr	IP Address Of The Second CE Router Interface
Netmask	Net mask For The IP Net
Hostmask	Net mask For The Host Net
PoolName	Name of IP address pool
IPNetAddrStr	IP address of the net link (IPNetAddr) padded with zeroes (e.g. 172.000.000.008). Used for sorting of IPNets.

## 10-1-13 Customer

**Table 10-10** Customer defines the Customer details.

Attribute Name	Description
CustomerId	Uniqueld assigne to the Customer
CustomerName	Name of the Customer

## 10-1-14 L3VPN

**Table 10-11** L3VPN defines the details of Layer 3 VPN.

Attribute Name	Description
ServiceId	Service Identifier
VPNTopologyType	Topology Type Full-Mesh or Hub-and-Spoke
QoSProfile_PE	The default QoS profile on the PE
QoSProfile_CE	The default QoS profile on the CE
CustomerId	Customer identifier
ServiceName	User assigned name
InitiationDate	Service initiation date
ActivationDate	Service activation date
ModificationDate	Service modification date
State	State of service
Type	Type of service
ContactPerson	Customer's contact person
Comments	Comment
ParentId	Id of the parent VPN (used for multicast VPN to reference normal vpn)
Multicast	Multicast status of the VPN

## 10-1-15 Site

**Table 10-12** Site defines the Site details.

Attribute Name	Description
ServiceId	Service Identifier
CustomerId	Customer identifier
ServiceName	User assigned name
InitiationDate	Service initiation date
ActivationDate	Service activation date
ModificationDate	Service modification date
State	State of service
Type	Type of service
ContactPerson	Customer's contact person
Comments	Comment
RemoteASN	Remote Autonomous System Number for eBGP link
OSPF_Area	Remote OSPF area id for OSPF link
SiteOfOrigin	Site of Origin identifier for multi-home service
Managed	Is the site managed?
Multicast	Multicast status of the site
PostalAddress	Address of Site

## 10-1-16 RC

**Table 10-13** RC defines route target values for the routing community defining the VPN. For each VPN three RC objects always exists: hub, spoke and mesh.

Attribute Name	Description
RCName	Identifier of the routing community
L3VPNId	Id of the VPN the RC belongs to
RTExport	Export route target
RTImport	Import route target
Type	Connectivity type: hub, spoke, mesh or multicast

## 10-1-17 RC Membership

**Table 10-14** RCMembership associates the VRF and its Routing Communities used by it.

Attribute Name	Description
VRFName	Name of VRF object
NE_ID	Networkelement identifier
RCName	Identifier of the routing community

## 10-1-18 L3AccessFlow

**Table 10-15** . L3AccessFlow defines the L3 Access Flow details. For a multi-homed site with is not fully protected only one site attachment object will exist, but two L3FlowPoints are present.

<b>Attribute Name</b>	<b>Description</b>
ServiceId	Service Identifier
CustomerId	Customer identifier
ServiceName	User assigned name
InitiationDate	Service initiation date
ActivationDate	Service activation date
ModificationDate	Service modification date
State	State of service
Type	Type of service
ContactPerson	Customer's contact person
Comments	Comment
__count	Reservation Count
__uniqueid	Service Instance Id
SiteId	Service Identifier for the site
VlanId	VLAN Id used for the AccessFlow
PE_Status	Configuration Status of the PE Router
CE_Status	Configuration Status of the CE Router
AccessNW_Status	Configuration Status of the Access Network
IPNet	IP address of the net link
Netmask	Netmask for The net link
Domain_id	OSPF domain ID in IP address format.
MDTData	The MDT data field for the multicast
LoopAddr	The address for multicast loopback interface
RP	Rendezvous point status if the multicast is enabled
CE_based_QoS	Is CE based QoS enabled?

## 10-1-19 L3FlowPoint

**Table 10-16** L3FlowPoint defines the Layer 3 PE FlowPoint details.

Attribute Name	Description
TerminationPointID	ID of the Termination point where Flow point is associated
AttachmentId	Service identifier for the attachment
QoSProfile_in	QoS profile for ingress traffic
QoSProfile_out	QoS profile for egress traffic
RateLimit_in	RateLimit for ingress traffic
RateLimit_out	RateLimit for egress traffic
Protocol	Routing protocol running on the attachment
Maximum_Prefix	Maximum number of prefix limit
StaticRoutes	Static routes for customer site
OSPF_id	ID of OSPF process. Can not be more than 65535.
Rip_id	ID of process on Huawei router. Can not be more than 65535. Used for RIP sites only.
VRFName	Name of VRF used by this attachment
PE_InterfaceIP	IP address of the PE interface
CE_InterfaceIP	IP address of the CE interface
mCAR	Bandwidth value for the multicast traffic
mCoS	Class of service(IPP) for the multicast traffic
LoopbackId	The loopback Id for multicast loopback interface
SOO_Configured	Status of the site of origin configuration
Master	Master Ip address if VRRP is enabled for site. ServiceUpload does not support VRRP protocol.
Priority	Priority to be used for VRRP.
Vrrp_Group_ID	VRRP group identifier.

## 10-1-20 FlowPoint

**Table 10-17** Flowpoint defines the Layer 3 PE and CE FlowPoint details.

Attribute Name	Description
TerminationPointID	ID of the Termination point where Flow point is associated
AttachmentId	Service identifier for the attachment
QoSProfile_in	QoS profile for ingress traffic
QoSProfile_out	QoS profile for egress traffic
RateLimit_in	RateLimit for ingress traffic
RateLimit_out	RateLimit for egress traffic

## 10-1-21 VPNFPMembership

**Table 10-18** VPNFPMembership defines the association of a flowpoint with the VPN. For each VPN the flowpoint is a member of, a VPNFPMembership object exists

Attribute Name	Description
VPNId	Service id of VPN
FlowPointId	TerminationPoint Id of Flowpoint

## 10-1-22 L3 VPN Membership

**Table 10-19** L3VPNMembership defines the association of site with the VPN. For each VPN the site is a member of, a L3 VPN Membership object exists.

Attribute Name	Description
VPNId	Service id of VPN
SiteId	Service id of Site
SiteName	Name of Site
VPNName	Name of VPN
JoinDate	Date of joining VPN
CustomerName	Name of VPN owner



# 11 Updating Information in the CRM Repository

The service discovery does not upload any customer attributes besides customer ID and name. If additional customer information must be stored in the CRM repository, the operator must provide this information (e.g. in a excel sheet). The information can either be entered through the CRM GUI or the CRM-customer table in the database can be updated using sql and other dataload tools.

**Figure 11-1** The CRM GUI allows manual update of the customer information. Only Customer name and id are stored in the VPN SVP repository and uploaded during service discovery.

The screenshot shows the HP CRM GUI for updating customer information. On the left is a sidebar with the HP logo and navigation links: 'Provisioning portal operator is admin', 'Customers' (with sub-links 'New customer', 'List', 'Search'), and 'Settings' (with a note 'Skip activation is default'). The main area is titled 'Update Customer' and contains a table with the following fields:

Field Name	Value	Description
Customer id	2	Unique customer identifier
Company name	Citibank	Name of the company
Company address	<input type="text"/>	Company address, both street and number
City	<input type="text"/>	The city in which the company is located
Zip code	<input type="text"/>	Postal code
Contact person First name	<input type="text"/>	First name of the contact person
Contact person Surname	<input type="text"/>	Surname of the contact person
Phone number	<input type="text"/>	Phone number of the contact person
E-mail address	<input type="text"/>	E-mail address of the contact person

A blue arrow button is located at the bottom right of the form.

---

# 12 Log Files

The logfiles contains information about the progress of the service discovery process and warning and error messages for issues encountered during the process.

Before committing data to the VPN SVP it is important that you have inspected and understood the error messages in the log files.

## 12-1 Location, naming convention and format of Log files

Log files from the service discovery process are located in the standard HPSA log directory under `$ACTIVATOR_VAR/log/<hostname>`.

The logfiles complies to the XML format of the standard log files. The file naming convention is `discovery_<session_id>.xml` where the session id has been specified when launching the service discovery program. If no session-id was specified, the time-stamp is used instead.

The log files can be inspected through the HPSA GUI under the EQUIPMENT-LOAD and DISCOVER tabs

## 12-2 Logging Level

The level of log information can be controlled by defined the `LOGLEVEL` property. The `loglevel` is defined in the `upload.setenv` file located in `$ACTIVATOR_OPT/solutions/SAVPN/etc/config/service_upload`

Valid values for the `LOGLEVEL` property is the same as for the HPSA product: `ERROR`, `WARNING`, `INFORMATIVE`, `DEBUG` and `DEBUG2`

If it is desired to control the `LOGLEVEL` value through the `mwmf.xml` file, you can remove the line:

```
LOGLEVEL=INFORMATIVE
```

In the `upload.setenv` file

## 12-3 What is being logged?

For each phase start time, end time and duration is being logged. Information about each sub step is listed, such as which router configuration is being passed, which operation in the compare phase is being executed.

Also information about the number of objects located and stored is logged.

The two sample entries below shows the start and end time for the compare phase.

BLRTHIRTC4FW11	12-10-2007 11:59:11	LOGGER_THREAD	Starting 12-10-2007 11:59:11 at : 12-10-2007 11:59:11	Driver	Upload
----------------	------------------------	---------------	--	--------	--------

BLRTHIRTC4FW11	12-10-2007 11:59:31	LOGGER_THREAD	Total execution time for COMPARE phase = 0h:0m:20s:355ms	Driver	Upload
----------------	---------------------	---------------	---	--------	--------

An sample of the number of object with errors:

BLRTHIRTC4FW11	12-10-2007 11:59:31	LOGGER_THREAD	Number of site connections with errors: 15	compare	Upload
----------------	---------------------	---------------	---	---------	--------

## 12-4 Error and Warning Messages in Log files

This section describes the error and warning messages in the log files. It is important to understand the course of the error messages before the services are committed into the VPN SVP repository.

### 12-4-1 Unreferenced VRF

VRF object on a router which are not associated an interface are listed as a warning during the Analyse phase. To prevent generation of future VRFName do not collide with the existing VRFs, the unused VRF will be stored in the VPN SVP repository.

The Operator can use the list of warning to manual clean up the routers of unused VRF objects.

BLRTHIRTC4FW11	12-10-2007 11:51:40	LOGGER_THREAD	Vrf: V15388:VoiceFOX on host: C10K is not referenced. The vrf will be uploaded to avoid future name clash.	Analyse	Upload
----------------	------------------------	---------------	--	---------	--------

### 12-4-2 No matching IP Address pool found

The error below indicates that a address pool matching the IP address was not found. The services where an address was not found will be marked with ERROR and will not be loaded.

BLRTHIRTC4FW11	12-10-2007 11:52:41	LOGGER_THREAD	Could not determine IP Addr pool for IPNet: 172.16.13.1. IP Address pools containing the IPnet must be configured before upload start	Analyse	Upload
----------------	------------------------	---------------	---	---------	--------

In VPN service interface where no ip addressed is defined (e.g. unnumbered IP) creates the following message:

BLRTHIRTC4FW11	06-12-2007 13:51:41	LOGGER_THREAD	Could not determine IP Addr pool for IPNet: Undefined. IP Address pools containing the IPnet must be configured before upload start	Analyse	Upload
----------------	------------------------	---------------	---	---------	--------

## 12-4-3 Unknown Router Protocol

Routing protocol for the service configured on router C300-1 and interface Serial2/0:0.48 could not be determined. The service can be committed to the VPN SVP

BLRTHIRTC4FW11	06-12-2007 10:05:47	LOGGER_THREAD	C3600-1,Serial2/0:0.48: Protocol could not be discovered, setting the protocol to Unknown	Analyse	Upload
----------------	------------------------	---------------	---	---------	--------

## 12-4-4 Committed VRF differs from discovered

The VRF committed into the VPN SVP repository is different from the VRF discovered in the network. The discovery VRF and the services using it will be marked with ERROR and will not be committed.

This is done to prevent storing the wrong VRF information.

BLRTHIRTC4FW11	06-12-2007 23:57:41	LOGGER_THREAD	Found VRF in configuration file with same name as VRF in inventory, but which has a different RD values! The found VRF's attribute 'Uploadstatus' is marked with 'ERROR'	Compare	Upload
BLRTHIRTC4FW11	06-12-2007 23:57:42	LOGGER_THREAD	The list of VRFs is : V1923:accs_linkpay1	Compare	Upload

## 12-4-5 VPN Services Mark with Error

At the end of the compare all interfaces with services which have errors are listed.

BLRTHIRTC4FW11	06-12-2007 23:57:48	LOGGER_THREAD	Following interfaces with VPN services are marked with ERROR. The services will not be committed. You may check if equipment upload of all service interfaces has been done.	Compare	Upload
----------------	------------------------	---------------	--	---------	--------

## 12-4-6 Can not delete Split Files

Before services in the service xml file can be committed into the VPN SVP repository, the service xml files is split into a set of files; one file of each type of object which must be committed. These temporary files are stored in \$SERVICE\_ACTIVATOR\_VAR/service\_upload/temp directory. Before the split operation start the directory is emptied for all files to ensure existing files to not conflict with the current session. If these files can not be deleted the message below is generated and the commit phase is terminated.

BLRTHIRTC4FW11	06-12-2007 13:32:06	LOGGER_THREAD	Error initializing split. Cleanup of C:/hp/OpenView/ServiceActivator/var///service_upload/temp/ failed	Commit	Upload
----------------	------------------------	---------------	--	--------	--------

## 12-4-7 Service Ownership has not been resolved

The ambiguous ownership of some services has not been resolved. The commit phase is terminated with a message specifying which lines must be addressed. The problem may be solved through the inventory view for the service discovery. The line states object were located where the customerid id is on the form: "<customer ID1>;<customer ID2>" or the service id is on the form: ""<service ID1>;<service ID2>".

BLRTHIRTC4FW11	06-12-2007 13:28:15	LOGGER_THREAD	<p>ERROR: MANUAL AUDIT OF EXPORT FILE NOT DONE CORRECTLY Please correct the following objects and run commit again</p> <p>ERROR: Line No. 320605: CustomerId must be resolved. ERROR: Line No. 325425: CustomerId must be resolved. ERROR: Line No. 325434: CustomerId must be resolved. ERROR: Line No. 325454: CustomerId must be resolved.</p>	Commit	Upload
----------------	------------------------	---------------	---	--------	--------

## 12-4-8 Error preventing Commit of Data

If inconsistencies exist in the service xml file, no data from the session will be committed. The service xml file has not been generated correctly by the operator. The error message states which object could not be committed and the encountered constraint. In the log below, the VPN service referenced by the VPNMembership does not exist.

BLRTHIRTC4FW11	06-12-2007 23:00:34	LOGGER_THREAD	<p>LOGGER_THREAD</p> <p>Error storing the bean - com.hp.ov.activator.vpn.inventory.L3VPNMembership VPNId = 6000:10481##6000:10481, SiteId = 28889, SiteName = BANK_I92121, BW: 64 Kbps, CID: MSHA000031734@dIci276, VPNName = msha000051425, JoinDate = , CustomerName = CITY_BANK, BW: java.sql.SQLException: ORA-02291: integrity constraint (VPN4_4.SYS_C0027750) violated - parent key not found Referenced value is not found.</p> <p>com.hp.ov.activator.vpn.inventory.L3VPNMembership is constrained by column: 'VPNID' in table: 'VPNMEMBERSHIP'</p> <p>Commit</p>		Upload
BLRTHIRTC4FW11	06-12-2007 23:00:34	LOGGER_THREAD	<p>LOGGER_THREAD</p> <p>ERRORS FOUND, ROLLING BACK.</p> <p>Commit</p>		Upload

## 12-5 Cleaning up of Log files

Log files can be deleted through the HPSA GUI as all other log files.

---

# Appendix A: Service XML File Format

## DTD for XML File Format

The section contains the DTD for the XML file that is used for committing the discovered serviced into the VPN SVP repository. The DTD is located in the \$SOLUTION/etc/config/service\_upload directory and is named ServiceUpload.dtd.

For explanation of the elements and their relationship, please see section 10-1 .

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT VPNServiceDiscovery ((Identifiers+, QOS, VRFs, CERouters, IPNets, Customers))>

<!ELEMENT Identifiers ((Identifier+))>
<!ELEMENT Identifier ((Max_ServiceId, Max_CustomerId))>
<!ELEMENT Max_ServiceId (#PCDATA)>
<!ELEMENT Max_CustomerId (#PCDATA)>

<!ELEMENT QOS ((TrafficClassifiers, QoSProfiles, PolicyMappings))>
<!ELEMENT TrafficClassifiers ((TrafficClassifier+))>
<!ELEMENT TrafficClassifier ((Name, CustomerId, DSCPs, Filter, CoSs, Layer, Compliant))>
<!ATTLIST TrafficClassifier
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT QoSProfiles ((QoSProfile+))>
<!ELEMENT QoSProfile ((QoSProfileName, CustomerId, Prefix, Description, Profilename_in,
Profilename_out, Layer, PEQoSProfileName, Compliant))>
<!ATTLIST QoSProfile
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT PolicyMappings ((PolicyMapping+))>
<!ELEMENT PolicyMapping ((TClassName, ProfileName, Exp, Dscp, Percentage, Position, Plp,
Queue, CoSName ))>
<!ATTLIST PolicyMapping
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT VRFs ((VRF+))>
```

```
<!ELEMENT VRF ((VRFName, RD, RCMemberships))>
<!ATTLIST VRF
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT RCMemberships ((RCMembership+))>
<!ELEMENT RCMembership ((RCName, VRFName))>
<!ATTLIST RCMembership
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT CERouters ((CERouter+))>
<!ELEMENT CERouter ((NetworkElementID, Name, IP, management_IP, Description, Location,
State, NetworkID, ElementType, Vendor, Username, UsernameEnabled, Password, EnablePassword,
ManagementInterface, OSVersion, Role, LifeCycleState, ROCommunity, RWCommunity,
SerialNumber, SchPolicyName, Backup, Managed, Present, __count, __uniqueid,
CE_LoopbackPool, Interfaces))>
<!ATTLIST CERouter
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT Interfaces ((Interface+))>
<!ELEMENT Interface ((TerminationPointID, Name, NE_ID, EC_ID, State, __count, __uniqueid,
Type, ParentIf, IPAddr, SubType, Encapsulation, ifIndex, ActiveState, UsageState, VlanId,
DLCI, Timeslots, SlotsNumber, Bandwidth, LmiType, IntfType, BundleKey, BundleId))>
<!ATTLIST Interface
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT IPNets ((IPNET+))>
<!ELEMENT IPNET ((IPNetAddr, PE1_IPAddr, CE1_IPAddr, PE2_IPAddr, CE2_IPAddr, Netmask,
Hostmask, PoolName, IPNetAddrStr, __count, __uniqueid))>
<!ATTLIST IPNET
Marker CDATA #IMPLIED
UploadStatus CDATA #IMPLIED
DBPrimaryKey CDATA #IMPLIED
>

<!ELEMENT Customers ((Customer+))>
<!ELEMENT Customer ((CustomerId, CustomerName, VPNs, Sites))>
<!ATTLIST Customer
```

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT VPNs ((L3VPN+))>

<!ELEMENT L3VPN ((ServiceId, CustomerId, ServiceName, InitiationDate, ActivationDate, ModificationDate, State, Type, ContactPerson, Comments, \_\_count, \_\_uniqueid, VPNTopologyType, QoSProfile\_PE, QoSProfile\_CE, ParentId, Multicast, RCs))>

<!ATTLIST L3VPN

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT RCs ((RC+))>

<!ELEMENT RC ((RCName, L3VPNId, RTEExport, RTImport, Type))>

<!ATTLIST RC

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT Sites (Site+)>

<!ELEMENT Site ((ServiceId, CustomerId, ServiceName, InitiationDate, ActivationDate, ModificationDate, State, Type, ContactPerson, Comments, \_\_count, \_\_uniqueid, RemoteASN, OSPF\_Area, SiteOfOrigin, Managed, Multicast, PostalAddress, SiteAttachments, L3VPN\_Memberships, L3SiteConnections))>

<!ATTLIST Site

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT L3AccessFlows (L3AccessFlow+)>

<!ELEMENT L3AccessFlow ((ServiceId, CustomerId, ServiceName, InitiationDate, ActivationDate, ModificationDate, State, Type, ContactPerson, Comments, SiteId, VlanId, PE\_Status, CE\_Status, AccessNW\_Status))>

<!ATTLIST L3AccessFlow

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT L3VPN\_Memberships (L3VPNMembership+)>

<!ELEMENT L3VPNMembership ((VPNId, SiteId, SiteName, VPName, JoinDate, CustomerName))>

<!ATTLIST L3VPNMembership

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED



>

<!ELEMENT L3FlowPoints (L3FlowPoint+)>

<!ELEMENT L3FlowPoint ((TerminationPointId, AttachmentId, QoSProfile\_In, QoSProfile\_out, RateLimit\_in, RateLimit\_out, Protocol, Maximum\_Prefix, StaticRoutes, OSPF\_id, Rip\_id, VRFName, PE\_InterfaceIP, CE\_InterfaceIP, mCAR, mCoS, LoopbackId, SOO\_Configured))>

<!ATTLIST L3FlowPoint

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT CEFlowPoints (FlowPoint+)>

<!ELEMENT FlowPoint ((TerminationPointId, AttachmentId, QoSProfile\_In, QoSProfile\_out, RateLimit\_in, RateLimit\_out, NE\_ID))>

<!ATTLIST FlowPoint

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT VPNFPMemberships (VPNFPMembership+)>

<!ELEMENT VPNFPMembership ((VPNId, FlowPointId))>

<!ATTLIST VPNFPMembership

Marker CDATA #IMPLIED

UploadStatus CDATA #IMPLIED

DBPrimaryKey CDATA #IMPLIED

>

<!ELEMENT QoSProfileName (#PCDATA)>

<!ELEMENT Compliant (#PCDATA)>

<!ELEMENT Prefix (#PCDATA)>

<!ELEMENT PEQoSProfileName (#PCDATA)>

<!ELEMENT Profilename\_in (#PCDATA)>

<!ELEMENT Profilename\_out (#PCDATA)>

<!ELEMENT ProfileName (#PCDATA)>

<!ELEMENT Position (#PCDATA)>

<!ELEMENT Percentage (#PCDATA)>

<!ELEMENT Plp (#PCDATA)>

<!ELEMENT Queue (#PCDATA)>

<!ELEMENT CosName (#PCDATA)>

<!ELEMENT Exp (#PCDATA)>

<!ELEMENT ClassName (#PCDATA)>

<!ELEMENT Dscp (#PCDATA)>

<!ELEMENT VRFName (#PCDATA)>

<!ELEMENT management\_IP (#PCDATA)>

<!ELEMENT mCoS (#PCDATA)>

<!ELEMENT mCAR (#PCDATA)>  
<!ELEMENT ifIndex (#PCDATA)>  
<!ELEMENT \_\_uniqueid (#PCDATA)>  
<!ELEMENT \_\_count (#PCDATA)>  
<!ELEMENT VlanId (#PCDATA)>  
<!ELEMENT Vendor (#PCDATA)>  
<!ELEMENT VPNTopologyType (#PCDATA)>  
<!ELEMENT VPNName (#PCDATA)>  
<!ELEMENT VPNId (#PCDATA)>  
<!ELEMENT UsernameEnabled (#PCDATA)>  
<!ELEMENT Username (#PCDATA)>  
<!ELEMENT UsageState (#PCDATA)>  
<!ELEMENT UploadStatus (#PCDATA)>  
<!ELEMENT Type (#PCDATA)>  
<!ELEMENT Timeslots (#PCDATA)>  
<!ELEMENT TerminationPointID (#PCDATA)>  
<!ELEMENT TP2 (#PCDATA)>  
<!ELEMENT TP1 (#PCDATA)>  
<!ELEMENT SubType (#PCDATA)>  
<!ELEMENT StaticRoutes (#PCDATA)>  
<!ELEMENT State (#PCDATA)>  
<!ELEMENT SlotsNumber (#PCDATA)>  
<!ELEMENT SiteOfOrigin (#PCDATA)>  
<!ELEMENT SiteName (#PCDATA)>  
<!ELEMENT SiteId (#PCDATA)>  
<!ELEMENT ServiceName (#PCDATA)>  
<!ELEMENT ServiceId (#PCDATA)>  
<!ELEMENT SerialNumber (#PCDATA)>  
<!ELEMENT SchPolicyName (#PCDATA)>  
<!ELEMENT SOO\_Configured (#PCDATA)>  
<!ELEMENT Role (#PCDATA)>  
<!ELEMENT Rip\_id (#PCDATA)>  
<!ELEMENT RemoteASN (#PCDATA)>  
<!ELEMENT RateLimit\_out (#PCDATA)>  
<!ELEMENT RateLimit\_in (#PCDATA)>  
<!ELEMENT RWCommunity (#PCDATA)>  
<!ELEMENT RTImport (#PCDATA)>  
<!ELEMENT RTEExport (#PCDATA)>  
<!ELEMENT RP (#PCDATA)>  
<!ELEMENT ROCommunity (#PCDATA)>  
<!ELEMENT RL\_CE\_out (#PCDATA)>  
<!ELEMENT RL\_CE\_in (#PCDATA)>  
<!ELEMENT RD (#PCDATA)>  
<!ELEMENT RCName (#PCDATA)>  
<!ELEMENT QoSProfile\_PE (#PCDATA)>  
<!ELEMENT QoSProfile\_CE (#PCDATA)>  
<!ELEMENT QoSProfile\_In (#PCDATA)>

<!ELEMENT QoSProfile\_Out (#PCDATA)>  
<!ELEMENT Protocol (#PCDATA)>  
<!ELEMENT Present (#PCDATA)>  
<!ELEMENT PostalAddress (#PCDATA)>  
<!ELEMENT PoolName (#PCDATA)>  
<!ELEMENT Password (#PCDATA)>  
<!ELEMENT ParentIf (#PCDATA)>  
<!ELEMENT ParentId (#PCDATA)>  
<!ELEMENT PE\_Status (#PCDATA)>  
<!ELEMENT PE\_InterfaceIP (#PCDATA)>  
<!ELEMENT PE2\_IPAddr (#PCDATA)>  
<!ELEMENT PE1\_IPAddr (#PCDATA)>  
<!ELEMENT OSVersion (#PCDATA)>  
<!ELEMENT OSPF\_id (#PCDATA)>  
<!ELEMENT OSPF\_Area (#PCDATA)>  
<!ELEMENT NetworkID2 (#PCDATA)>  
<!ELEMENT NetworkID1 (#PCDATA)>  
<!ELEMENT NetworkID (#PCDATA)>  
<!ELEMENT NetworkElementID (#PCDATA)>  
<!ELEMENT Netmask (#PCDATA)>  
<!ELEMENT Name (#PCDATA)>  
<!ELEMENT NE\_ID (#PCDATA)>  
<!ELEMENT NE2 (#PCDATA)>  
<!ELEMENT NE1 (#PCDATA)>  
<!ELEMENT Multicast (#PCDATA)>  
<!ELEMENT ModificationDate (#PCDATA)>  
<!ELEMENT Maximum\_Prefix (#PCDATA)>  
<!ELEMENT Marker (#PCDATA)>  
<!ELEMENT ManagementInterface (#PCDATA)>  
<!ELEMENT Managed (#PCDATA)>  
<!ELEMENT MDTData (#PCDATA)>  
<!ELEMENT LoopBackId (#PCDATA)>  
<!ELEMENT LoopAddr (#PCDATA)>  
<!ELEMENT Location (#PCDATA)>  
<!ELEMENT LmiType (#PCDATA)>  
<!ELEMENT LifeCycleState (#PCDATA)>  
<!ELEMENT Layer (#PCDATA)>  
<!ELEMENT L3VPNId (#PCDATA)>  
<!ELEMENT JoinDate (#PCDATA)>  
<!ELEMENT IntfType (#PCDATA)>  
<!ELEMENT InitiationDate (#PCDATA)>  
<!ELEMENT IPNetAddrStr (#PCDATA)>  
<!ELEMENT IPNetAddr (#PCDATA)>  
<!ELEMENT IPNet (#PCDATA)>  
<!ELEMENT IPAddr (#PCDATA)>  
<!ELEMENT IP (#PCDATA)>  
<!ELEMENT Hostmask (#PCDATA)>

<!ELEMENT FlowPointId (#PCDATA)>  
<!ELEMENT Filter (#PCDATA)>  
<!ELEMENT Encapsulation (#PCDATA)>  
<!ELEMENT EnablePassword (#PCDATA)>  
<!ELEMENT ElementType (#PCDATA)>  
<!ELEMENT EC\_ID (#PCDATA)>  
<!ELEMENT Domain\_id (#PCDATA)>  
<!ELEMENT Description (#PCDATA)>  
<!ELEMENT DSCPs (#PCDATA)>  
<!ELEMENT DLCI (#PCDATA)>  
<!ELEMENT DBPrimaryKey (#PCDATA)>  
<!ELEMENT CustomerName (#PCDATA)>  
<!ELEMENT CustomerId (#PCDATA)>  
<!ELEMENT ContactPerson (#PCDATA)>  
<!ELEMENT ConnectionID (#PCDATA)>  
<!ELEMENT Comments (#PCDATA)>  
<!ELEMENT CoSs (#PCDATA)>  
<!ELEMENT CE\_Status (#PCDATA)>  
<!ELEMENT CE\_InterfaceIP (#PCDATA)>  
<!ELEMENT CE2\_IPAddr (#PCDATA)>  
<!ELEMENT CE1\_IPAddr (#PCDATA)>  
<!ELEMENT BundleKey (#PCDATA)>  
<!ELEMENT BundleId (#PCDATA)>  
<!ELEMENT Bandwidth (#PCDATA)>  
<!ELEMENT Backup (#PCDATA)>  
<!ELEMENT AttachmentId (#PCDATA)>  
<!ELEMENT ActiveState (#PCDATA)>  
<!ELEMENT ActivationDate (#PCDATA)>  
<!ELEMENT AccessNW\_Status (#PCDATA)>  
<!ELEMENT ASBR\_Status (#PCDATA)>