SCMS Subscriber Manager User Guide

OL-7199-01

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-138629=
Text Part Number: OL-7199-01

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Cisco's installation instructions, it may cause interference with radio and television reception. T his equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

Modifying the equipment without Cisco's written authorization may result in the equipment no longer complying with FCC requirements for Class A or Class B digital devices. In that event, your right to use the equipment may be limited by FCC regulations, and you may be required to correct any interference to radio or television communications at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the Cisco equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following measures:

- Turn the television or radio antenna until the interference stops.
- Move the equipment to one side or the other of the television or radio.
- Move the equipment farther away from the television or radio.
- Plug the equipment into an outlet that is on a different circuit from the television or radio. (That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)

Modifications to this product not authorized by Cisco Systems, Inc. could void the FCC approval and negate your authority to operate the product.

Printed in the USA on recycled paper containing 10% postconsumer waste.

***SCMS Subscriber Manager** User Guide*

CONTENTS

Preface

Audience

This guide is for the networking or computer technician who is responsible for the on-site installation and configuration of the SM. It is also intended for the operator who will be responsible for daily operations of the SM.

Purpose

This document is a guide to the configuration and management of the Subscriber Manager (SM) application. It contains explanations of relevant concepts and terminology, as well as instructions on how to configure, install, and use the SM.

The guide assumes a basic familiarity with the concepts of the Service Control Management Suite, the SCE1000 and SCE2000 Platforms, and their related components.

This *Service Control Management Suite Subscriber Manager SCMS SM) User's Guide* documents the application that processes subscriber records and maintains the subscriber database for use by the SCE Platform. It contains instructions for installing the SM and the TimesTen™ database, for editing and installing the configuration file, and for managing the system using the Command Line Utilities (CLU). It also specifies how to configure a Service Control deployment to support subscriber integration in its various forms.

The following typographic conventions are used in this guide:

Typeface or Symbol	Meaning
Italics	References, new terms, field names, and placeholders.
Bold	Names of menus, options, and command buttons.
`Courier`	System output shown on the computer screen.
`Courier Bold`	CLU command code typed in by the user in examples.
`Courier Italic`	Required parameters for CLU commands.
`[italic in brackets]`	Optional parameters for CLU commands.

Typeface or Symbol	Meaning
➲	A one-step procedure.
✎	Note. Notes contain important information.
⚠	Warning. Warning means danger of bodily injury or of damage to equipment.

Technical Support

Cisco TAC Website

The Cisco TAC website (*http://www.cisco.com/tac* (http://www.cisco.com/tac)) provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The Cisco TAC website is available 24 hours a day, 365 days a year.

Accessing all the tools on the Cisco TAC website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a login ID or password, register at this URL: *http://tools.cisco.com/RPF/register/register.do* (http://tools.cisco.com/RPF/register/register.do)

Opening a TAC Case

The online TAC Case Open Tool (*http://www. cisco.com/tac/caseopen* (http://www.cisco.com/tac/caseopen)) is the fastest way to open P3 and P4 cases. (Your network is minimally impaired or you require product information). After you describe your situation, the TAC Case Open Tool automatically recommends resources for an immediate solution.

If your issue is not resolved using these recommendations, your case will be assigned to a Cisco TAC engineer. For P1 or P2 cases (your production network is down or severely degraded) or if you do not have Internet access, contact Cisco TAC by telephone. Cisco TAC engineers are assigned immediately to P1 and P2 cases to help keep your business operations running smoothly.

To open a case by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete listing of Cisco TAC contacts, go to this URL: *http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml* (http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml)

TAC Case Priority Definitions

To ensure that all cases are reported in a standard format, Cisco has established case priority definitions.

Priority 1 (P1)—Your network is “down” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Priority 2 (P2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Priority 3 (P3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Priority 4 (P4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

CHAPTER 1

Overview

This chapter provides a general overview of the Cisco Service Control solution. It introduces the Cisco Service Control concept and the Service Control capabilities. It also briefly describes the hardware capabilities of the SCE Platform, as well as the Cisco specific applications that together compose the total Cisco Service Control solution.

This chapter contains the following sections:

The Cisco Service Control Concept

The Cisco Service Control concept is delivered through a combination of purpose-built hardware and specific software solutions that address various Service Control challenges faced by service providers. The SCE Platform is designed to support observation, analysis, and control of Internet/IP traffic.

Service Control enables service providers to create profitable new revenue streams while capitalizing on their existing infrastructure. With the power of Service Control, service providers have the ability to analyze, charge for, and control IP network traffic at multi-Gigabit wire line speeds. The Cisco Service Control solution also gives service providers the tools they need to identify and target high-margin, content-based services.

As the downturn in the telecommunications industry has shown, IP service provider business models need to be reworked in order to make them profitable. Having spent billions of dollars to build ever larger data links, providers have incurred massive debts and rising costs. During the same time, access and bandwidth became a commodity where prices continually fell and profits disappeared. Service providers now realize that they must offer value-added services to derive more revenue from the traffic and services running on their networks. However, capturing real profits from IP services requires more than simply running those services over data links; it requires detailed monitoring and precision, real- time control and awareness of services as they are delivered. Cisco provides Service Control solutions that allow the service provider to bridge this gap.

Service Control for Wireless Service Providers

Wireless Service Providers are successfully rolling out 2.5G and 3G-based data services to their subscribers.

These services are expected to significantly increase much needed Average Revenue Per User (ARPU) for sustained business models and rapid rollout of new services. These data services require new ways of service offering and new ways of billing these services to the subscribers. The Cisco Service Control solutions enable:

- Support for multiple billing models
- Elimination of revenue leakage via real-time service control
- Flexible pricing plans: postpaid, prepaid, MRC, pay-per-use
- Content-based billing for various applications
- Subscription-based and tiered application services

Service Control for DSL Providers and ISPs

DSL providers and ISPs targeting residential and business broadband customers must find new ways to get maximum leverage from their existing infrastructures, while differentiating their offerings with enhanced IP services.

Cisco products add a new layer of service intelligence and control to existing networks, and will:

- Provide granular visibility into network usage
- Automatically enforce application SLAs or acceptable use policies
- Implement different service levels for different types of customers, content, or applications
- Deploy from network edge to network core for end-to-end service control
- Integrate Cisco solutions easily with existing network elements and BSS/ OSS systems

Service Control for Cable MSOs

Cable MSOs have successfully deployed high-speed cable modem services to millions of homes. Now, they must move beyond providing commodity broadband access by introducing differentiated services and by implementing the service control necessary to fully manage service delivery through their broadband infrastructure. Cisco Service Control solutions will enable:

- Ability to report/analyze network traffic at subscriber and aggregate level for capacity planning
- Identification of network abusers who are violating the Acceptable Use Policy
- Identification and management of peer-to-peer, NNTP (news) traffic, and spam abusers
- Enforcement of the Acceptable Use Policy (AUP)
- Ability to limit the use of servers in the subscriber residence, as well as the use of multiple (unpaid) computers
- Customer-intuitive tiered application services and guarantee application SLAs
- Full integration with standard or legacy OSS for subscriber management and billing

Service Control Capabilities

At the core of the Cisco Service Control Platform stands the purpose-built network hardware device: the Service Control Engine (SCE). Implementing a complete Service Control solution requires that the Service Control Engine provide certain functionalities and capabilities. The following are the core capabilities of the Cisco Service Control Engine, which support a wide range of applications for delivering Service Control solutions:

- Subscriber and application awareness: Application-level drilling into IP traffic for real-time understanding and controlling of usage and content at the granularity of a specific subscriber.
    - Subscriber awareness: The ability to map between IP flows and a specific subscriber for maintaining the state of each subscriber transmitting traffic through the platform, and enforcing the appropriate policy on this subscriber traffic.

      Subscriber awareness is achieved using dedicated integrations with subscriber management repositories, such as a DHCP or a Radius server.
    - Application awareness: The ability to understand and analyze traffic up to the application protocol layer (Layer 7).

      For an application protocol that is implemented using bundled flows (such as FTP, which is implemented using Control and Data flows), the SCE Platform understands the bundling connection between the flows and treats them accordingly.
- Stateful, real time traffic control: The ability to perform advanced control functions, including granular BW metering and shaping, quota management and redirection, utilizing stateful real-time traffic transaction processing. This requires highly adaptive protocol and application level intelligence.
- Programmability: The ability to quickly add new protocols and easily adapt to new services and applications in the ever-changing service provider environment. Programmability is achieved using the SML language.

  Programmability means that new services can be deployed quickly and provides an easy upgrade path for network, application, or service growth.
- Robust and flexible back office integration: The ability to integrate with existing 3rd party systems at the Service Provider, such as provisioning systems, subscriber repositories, billing systems, and OSS systems. The Service Control Engine provides a set of open and well-documented APIs that allows a quick and robust integration process.
- Scalable High-Performance Service Engines: The ability to execute all operations described above at wire speed.

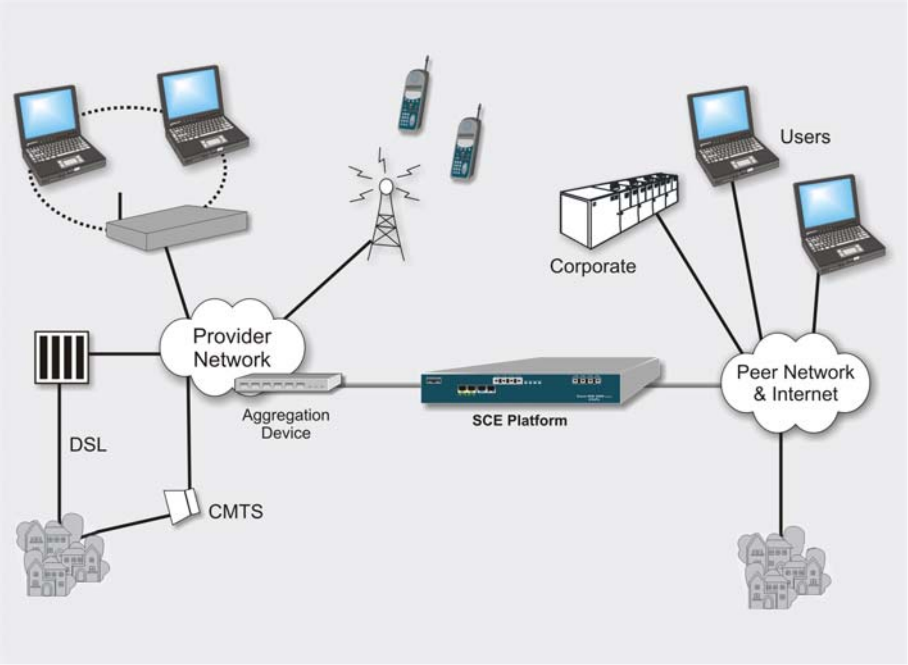
The SCE Platform

The Service Control Engine family of programmable network devices is capable of performing stateful flow inspection of IP traffic, and controlling that traffic based on configurable rules. The Service Control Engine is a purpose-built network device making use of ASIC components and RISC processors to go beyond packet counting and delve deeper into the contents of network traffic. Providing programmable, stateful inspection of bi-direction traffic flows and mapping these flows with user ownership, the Service Control Engine platforms provide a real-time classification of network usage. This information provides the basis of the Service Control Engine advanced traffic control and bandwidth shaping functionality. Where most bandwidth shaper functionality ends, the Service Control Engine provides more control and shaping options including:

- Layer 7-3 stateful wire-speed packet inspection and classification
- Robust support for over 600 protocol/applications including:
  - General: HTTP, HTTPS, FTP, TELNET, NNTP, SMTP, POP3, IMAP, WAP, and others
  - P2P: FastTrack-KazaA, Gnutella, WinMX, Winny, Hotline, eDonkey, DirectConnect, Piolet, and others
- Streaming & Multimedia: RTSP, SIP, HTTP-STREAMING, RTP/RTCP, and others
- Programmable system core for flexible reporting and bandwidth control
- Transparent network and BSS/OSS integration into existing networks
- Subscriber awareness for relating traffic and usage to specific customers

The following diagram demonstrates a deployment of an SCE Platform in the network.

Figure 1-1: SCE Platform in the Network



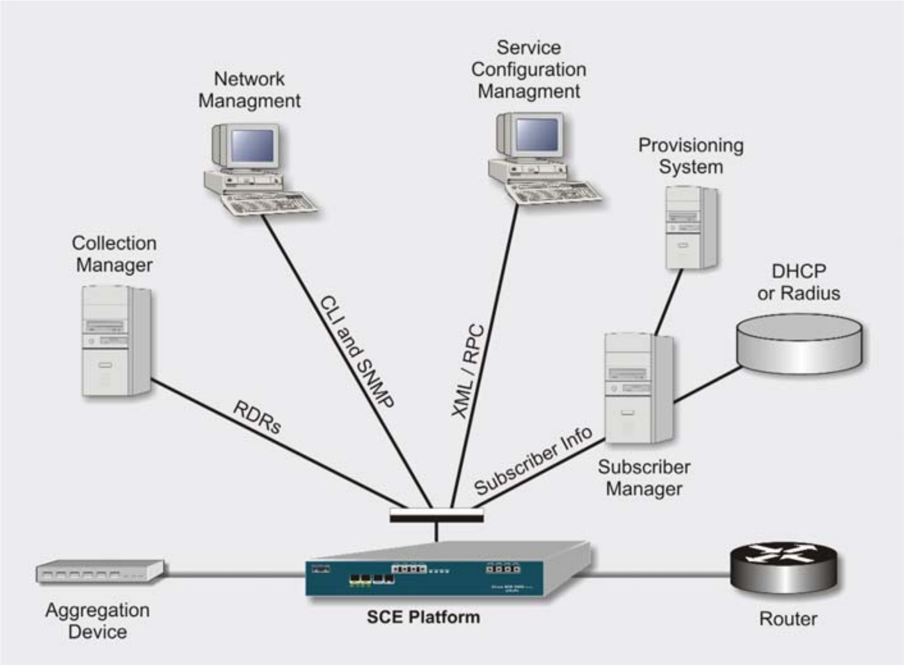
Management and Collection

The Service Control solution includes a complete management infrastructure that provides the following management components to manage all aspects of the Service Control solutions:

- Network management
- Subscriber management
- Service Control Management

These management interfaces are designed to comply with common management standards and to easily integrate with existing OSS infrastructure.

Figure 1-2: Service Control Management Infrastructure



Network Management

Cisco provides complete network FCAPS Management (Fault, Configuration, Accounting, Performance, Security).

Two interfaces are provided for network management:

- **CLI** (Command Line Interface). The CLI is accessible through the Console port or through a Telnet connection.

  CLI is used for configuration and security functions.

- **SNMP** (Simple Network Management Protocol).

  SNMP provides fault management via SNMP traps, as well as performance monitoring functionality.

Subscriber Management

The smartSUB Manager (SM) is a middleware software component used for bridging between the OSS and the SCE Platform(s). Subscriber information is stored in the SM database and can then be distributed between multiple devices according to actual subscriber placement.

The SM provides subscriber awareness, mapping network IDs to subscriber IDs. It obtains subscriber information using dedicated integration modules, which integrate with AAA devices like Radius or DHCP servers.

Subscriber information may be introduced into the SCE platform in one of two ways:

- Push Mode: The SM pushes subscriber information to the SCE Platform automatically upon logon of a subscriber.
- Pull Mode: On-demand, in response to a query from the SCE Platform to the SM.

Service Configuration Management

Service configuration management is the ability to configure the general service definitions of a Service Control application. Service Configuration is performed by creating an XML file and then applying it onto the SCE Platform using the Service Configuration utilities and management commands. This XML based approach is simple to use and easy to automate.

Collection

All the analysis and data processing functions of the SCE Platform result in the generation of Raw Data Records (RDRs). These RDRs are processed by the Collection Manager. The Collection Manager software is an implementation of a collection system, listening in on RDRs from one or more SCE Platforms. It collects these records, and processes them in one of its adapters. Each adapter performs a specific action on the RDR.

RDRs contain a wide variety of information and statistics, depending on the configuration of the system. There are three main categories of RDRs:

- Transaction RDRs: Records generated for each transaction, where a transaction is a single event detected in network traffic. The identification of a transaction will depend on the particular application/protocol.
- Subscriber RDRs: Records generated per subscriber, describing the traffic generated by that subscriber for a defined interval.
- Link RDRs: Records generated per link, describing the traffic carried on the link for a defined interval.

Cisco Service Control Specific Solutions

Cisco provides two specific solutions that run on top of the SCE Platform. Each solution addresses a different IP network control challenge that service providers face.

The Cisco specific solutions are:

- ***Service Control Application Suite for Broadband***
- ***Service Control Application Suite for Mobile***

Service Control Application Suite for Broadband

The ***Service Control Application Suite for Broadband*** allows service providers to detect complex and evasive network application protocols (such as P2P), and to control them as per their business and service delivery requirements. It also enables the creation of differentiated tiered services that the service provider uses to boost revenues and provide competitive services to end customers. ***Service Control Application Suite for Broadband***'s programmable application detection and subscriber awareness makes tiered service possible from one central point in the network. The ***Service Control Application Suite for Broadband*** requires no network changes or upgrades, and is compatible with all existing IP network switches, routers, and infrastructure.

Service Control Application Suite for Mobile

In this solution the SCE Platform is instrumental as a real-time post- and pre-paid network billing and traffic control device. It implements post-paid and pre-paid billing plans that relate subscriber access and network bandwidth consumption. The ***Service Control Application Suite for Mobile*** solution tracks detailed user specific traffic/application metrics and applies service and quota controls depending on their pre-paid balances.

CHAPTER 2

Introducing the Subscriber Manager

The Subscriber Manager (SM) is a middleware software component that supplies subscriber information for multiple SCE Platforms in deployments where dynamic subscriber awareness is required. It does this in one of two ways:

- By pre-storing the subscriber information
- By serving as a stateful bridge between an AAA or a provisioning system and the SCE Platforms

The SCE Platforms use subscriber information to provide subscriber-aware functionality, per-subscriber reporting, and policy enforcement.

Some Cisco Service Control solutions can also operate without subscriber awareness:

- Subscriber-less: Control and link level analysis functions are provided at a global device resolution.
- Anonymous subscriber: The system dynamically creates “anonymous” subscribers per IP address. User-defined IP address ranges may then be used to differentiate between anonymous subscribers policy.
- Static subscriber awareness: Subscriber awareness is required, but allocation of network IDs (mainly IP addresses) to subscribers is static.

In these three modes, the SCE Platform handles all subscriber-related functionality and an SM module is not required.

Note Starting with SM version 2.2, the SM can be configured to operate either with or without a cluster of two SM nodes (see SM Cluster). The added functionality when operating in a *cluster topology* provides powerful new features such as *fail-over* and *high availability*. (For definition of italicized terms, see Glossary of Terms.) The information in most of this chapter is applicable whether using a cluster or not. However, for clarity, information that is applicable **only** when using a cluster is presented separately at the end of this chapter, in the section *Subscriber Manager Fail-Over* ("Subscriber Manager (SM) Fail-Over" on page 2-10).

This chapter contains the following sections:

Subscribers in Cisco Service Control's Solutions

A *subscriber* is defined as a managed entity on the subscriber side of the SCE Platform, to which accounting and policy are applied individually. The *subscriber side of the SCE Platform* is the side of the SCE Platform that points to the access or downstream part of the topology, as opposed to the network side, which points to the core of the network.

Handling Subscribers

The SM addresses the following issues in allowing dynamic subscriber awareness:

- Mapping: The SCE Platform encounters flows with network IDs (IP addresses) that change dynamically, and it requires dynamic mapping between those network IDs and the subscriber IDs. The SM database contains the network IDs that map to the subscriber IDs. This is the main functionality of the SM.
- Policy: The SM serves as a repository of policy information per subscriber. The policy information may be pre-configured to the SM, or dynamically provisioned when the mapping information is provided.
- Capacity: The SCE Platform(s) may need to handle (over time) more subscribers than it can concurrently hold. In this case, the SM serves as an external repository for subscriber information, while the SCE platform is introduced only with the online or active subscribers.
- Location: The SM supports the functionality of sending subscriber information only to the relevant SCE platforms, in case such functionality is required. This is implemented using the domains mechanism or the Pull mode (see *Pull Mode* (on page 2-6)).

The SM database (see *SM Database* (on page 2-5)) can function in one of two ways:

- As the only source for subscriber information: when the SM works in standalone mode
- As a subscriber information cache: when the SM serves as a bridge between a group of SCE devices and the customer AAA and OSS systems.

Flow of Subscriber Information

The following figure shows the flow of subscriber information through the SM.

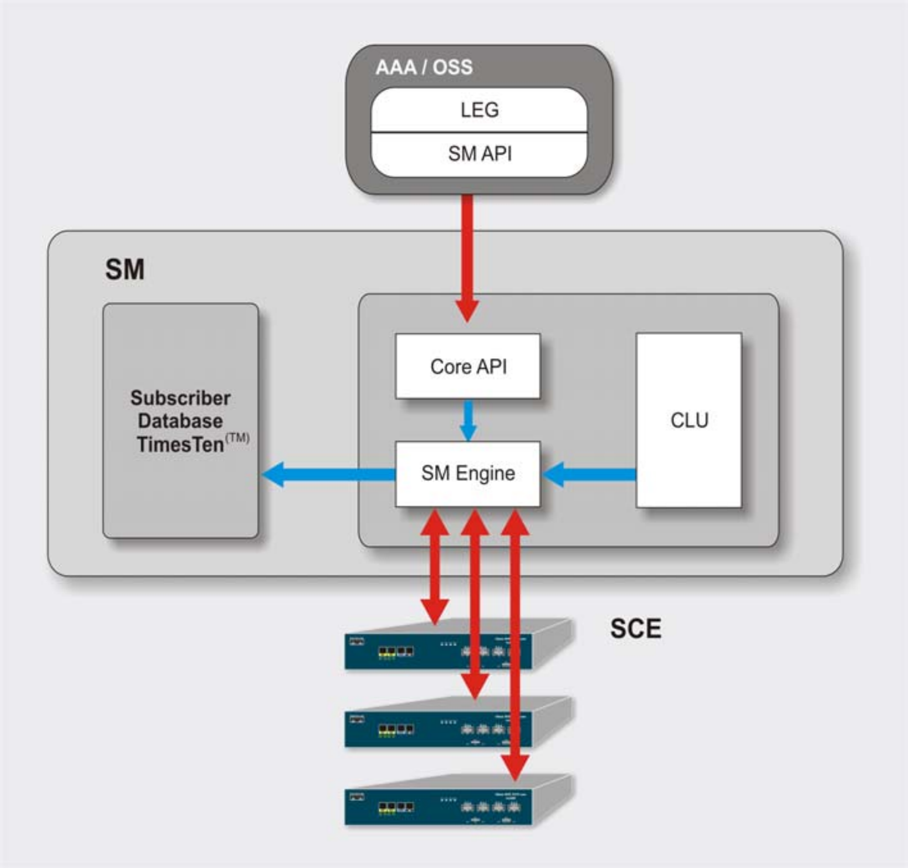


Figure 2-1: SM General Architecture

The flow takes place as follows:

- Subscriber information enters the SM in one of two ways:
    - Automatically upon the subscriber going online: A Login Event Generator (LEG) software module that integrates with the customer AAA system (such as DHCP Server, Radius, or NAS) identifies a subscriber login event, and sends it to the SM by using the SM API.
    - Manual setup: Subscriber information is imported into the SM from a file or by using the Command Line Utilities (CLU).

  Automatic and manual modes can be combined. For example, all subscribers may be loaded to the SM via manual setup, and a subset of the subscriber record (domain, network ID, etc.) changed automatically through the SM API.
- In automatic mode, the SCMS SM's Java or C/C++ APIs are used for delegating subscriber information to the SM (see the SCMS *SM Java API Programming Manual* or the SCMS *SM C/C++ API Programming Manual*).
- The SM Engine:
    - Stores subscribers in the subscriber database

- Introduces subscriber information to SCE Platforms

  The information may be passed automatically to the SCE Platform, or it may reside in the SM database until requested by the SCE Platform.

The SM may be configured with more than one SCE Platform. These SCE Platforms may then be grouped into domains. Each domain represents a group of SCE Platforms that serve the same group of subscribers.

Number of Subscribers in the SM

The subscribers of the service provider may be divided into the following logical types (at a given moment):

- Offline subscriber: a subscriber that currently does not have any IP address and as such does not generate any IP traffic. Such subscribers are not stored in the SCE Platform(s).
- Online subscriber: a subscriber that is currently online.

  At any particular time, a certain number of online subscribers will be idle, that is, connected to the service provider but not generating any IP traffic.
- Active subscriber: an online subscriber that is actually generating IP traffic (such as by surfing the Net or downloading a file).

In addition, the *Total number of subscribers* are all the subscribers whose IP traffic might be traversing through the SCE Platforms in a specific deployment.

There are four general scenarios for a network system using the SCE Platform(s):

- Total number of subscribers can be statically stored in a single SCE Platform.

  This is the simplest, most reliable scenario. It may not require the use of the SM.
- Total number of subscribers exceeds the capacity of the SCE Platform, but the number of online subscribers predicted at any time can be statically stored in the SCE Platform.

  It is recommended to use the SM in Push mode (see *Push Mode* (on page 2-6)).
- Number of online subscribers exceeds the capacity of the SCE Platform, but the number of active subscribers predicted at any one time can be statically stored in the SCE Platform.

  The SM must be used in Pull mode (see *Pull Mode* (on page 2-6)).
- Number of active subscribers predicted at any one time exceeds the capacity of the SCE Platform.

  Multiple SCE devices must be installed to divide the subscribers among the SCE Platforms. If the system is divided into domains (see *Domains* (on page 2-8)), so that the SM knows in advance to which SCE a particular subscriber should be sent, Push mode may be used. Otherwise, Pull mode is required.

For specific scenarios using the SM with multiple servers and/or SCE Platforms, see *System Configuration Examples* (on page 4-13).

SM Database

The SM uses a commercial relational database from TimesTen, optimized for high performance and with background persistency scheme. The In-Memory Database efficiently stores and retrieves subscriber records.

A subscriber record stored in the SM-DB consists of the following components:

- Subscriber name (Key): a string identifying the subscriber in the SM. Maximum length: 40 characters.
- Domain (Secondary key): a string that specifies which group of SCE devices handles this subscriber.
- Subscriber network IDs (mappings): a list of network identifiers, such as IP addresses or VLAN. The SCE device uses these identifiers to associate network traffic with subscriber records.
- Subscriber policy: a list of properties that instruct the SCE what to do with the network traffic of this subscriber. The content of this list is application specific.
- Subscriber state (for example, quota used): a field that encodes the subscriber state, recorded by the last SCE device, to handle the network traffic of this subscriber.

The subscribers are accessed using one of two indexes:

- subscriber name
- subscriber name + domain

Note that in Cluster redundancy topology, the active machine database replicates the subscriber data to the standby machine database. For additional information, see the section Subscriber *Manager Fail-Over* ("Subscriber Manager (SM) Fail-Over" on page 2-10).

SM Fundamentals

The SM API

The SM API is used for:

- altering the fields of an already existing subscriber record
- setting up new subscribers in the SM
- performing queries

The SM API is provided in C++, C, and Java. It serves as the bottom-most layer of every LEG.

SM API programmer references are provided in the SCMS *SM C/C++ API* and the SCMS *SM Java API* manuals (see *Contents of CDROM* (on page 3-2)).

The SM Login Event Generators (LEGs)

The SM Login Event Generators (LEGs) are software components that use the SM API to generate subscriber-record update messages (such as login/logout) and send them to the SM. LEGs are usually installed with AAA/OSS platforms, or with provisioning systems. They translate events generated by these systems to Cisco Service Control subscriber update events.

The unique functionality of each LEG depends on the specific software package with which it interacts. For example, Radius LEGs, DHCP LEGs, or some provisioning 3rd party system LEGs may be implemented. LEGs can set up subscribers or alter any of the fields of an existent subscriber record.

Multiple LEGs can be connected to a single SM. Conversely, a single LEG can generate events for multiple domains.

Subscriber Introduction Mode: Push/Pull

As illustrated in the figure in *Flow of Subscriber Information* (on page 2-3), the SM introduces subscriber data to the SCE Platforms. This operation functions in one of two modes:

- Push: This is the simpler and recommended mode.
- Pull: This mode should be used only in special cases, as explained below.

Push or Pull mode is configured for the entire SM system.

Push Mode

In Push mode, immediately after a subscriber record is added or changed, the SM distributes, or pushes, this information to the relevant SCE Platform(s), as determined by the subscriber's domain. When the subscriber starts producing traffic through the SCE Platform, it is ready with the required subscriber information.

In some scenarios, factors such as capacity limitations make it impossible to use Push mode.

Note Push mode can be used **only** if all online subscribers associated with a domain can be loaded simultaneously into all the SCE Platforms in the domain.

Pull Mode

In Pull mode, the SCE Platform(s) are not notified in advance of subscriber information. When an SCE Platform cannot associate the IP traffic with a subscriber, it will request, or pull, the information from the SM.

The advantage of Pull mode is that there is no need to know in advance which SCE Platform serves which subscriber.

The disadvantages of Pull mode are:

- Increased communication in the SM-SCE link
- Increased load on the SM, as it processes incoming requests from both the SCE device and the LEG.

Note Pull mode must be used when all online subscribers associated with a domain exceed the capacity of the SCE Platforms in the domain (but the number of active subscribers can still be loaded into the SCE Platforms in the domain).

The following table summarizes the differences between the Push mode and Pull mode:

Table 2-1 Differences Between Push Mode and Pull Mode

Aspect of Use	Push Mode	Pull Mode
When to use?	For simple provisioning of subscriber information to the SCE Platform	For real-time, on-demand subscriber information retrieval Used in large scale deployments: • When there is no way of knowing from the IP assignment process which SCE Platform will be serving a particular subscriber • When the required number of logged-in subscribers is greater than the number of concurrently active subscribers that the SCE Platform can handle
Functional flow at access time	• Subscriber network login or access • From subscriber information to LEG to SM • From SM to the relevant SCE Platforms	• Subscriber network login or access • From subscriber information to LEG to SM (hold in the SM database) • When the subscriber starts producing traffic that traverses the SCE Platform • SCE Platform asks for the subscriber information • From SM (SM database) to SCE Platform
Subscriber information at the SCE Platform	SCE Platform always has current subscriber information: • Immediate policy enforcement • Real-time system architecture	SCE gets subscriber information on demand

Domains

The SM provides the option of partitioning SCE Platforms and subscribers into subscriber domains.

The motivation for the domains concept is for enabling a single SM to handle several separate network sections, and for better control of subscriber introduction to the SCEs.

A subscriber domain is a group of SCE Platforms that share a group of subscribers. The subscriber traffic can pass through any SCE Platform in the domain. A subscriber can belong to only a single domain. Usually a subscriber will not be served by more than a single SCE Platform at any given time.

Domains are managed differently in Push and Pull modes:

- In Push mode, all the subscribers in a subscriber domain are sent to all SCEs in the domain. The main reason for the number of SCE Platform in a single domain is redundancy.
- In Pull mode, the pull requests are handled only for subscribers in the domain of the pulling SCE Platform. In Pull mode, usually, a single domain covers all the subscribers.

The system is configured with one default subscriber domain called **subscribers**. When an SCE Platform is added to the SM, it is automatically added to this default domain, unless otherwise specified. Subscribers are also associated with this default subscriber domain, unless otherwise specified. To associate a subscriber with a different domain, first define this domain in the configuration file, and then explicitly specify it when adding the subscriber to the SM. To associate an SCE Platform with a non-default subscriber domain, edit and reload the configuration file. (For details, see *Configuration and Management* (on page 4-1).)

Communication Failures

A communication failure may occur either on the LEG-SM communication link or on the SM-SCE communication link. A communication failure may occur due to a network failure or because the SCE, SM, or LEG has failed. High availability and recovery from an SM failure are discussed in SM Cluster.

When configuring the system, you should consider three issues related to communication failures:

- Communication failure detection: a timeout after which a communication failure is announced
- Communication failure handling: the action to be taken when communication on the link fails
- Communication failure recovery: the action to be taken when communication on the link resumes

Failure detection mechanism

A communication failure is detected by either of two mechanisms:

- Monitoring the TCP socket connection state. The monitoring is done by all peers.
- Using a keep-alive mechanism at the RPC protocol level

Failure handling mechanism

There are two configuration options for handling communication failures:

- Ignore communication failures
- Erase the subscriber mappings in its database and start handling flows without subscriber awareness

Erasing the mappings in the database is useful when you want to avoid incorrect mappings of subscribers to IP addresses. This configuration is implemented by requesting *clear all mappings upon failure*.

Failure recovery mechanism

There are two configuration options for recovering from communication failures:

- Resynchronizing the SCE Platform with the SM database
- Not taking any action

SM Cluster

The SM supports *high availability* using Veritas Cluster (TM) technology. In a high availability topology, the SM software is run on two machines, designated as the active machine and standby machines. Subscriber data is continuously replicated from the active to the standby machine, ensuring there is minimal data loss in case of active SM failure. When the active machine fails, the standby machine discovers the failure and becomes active. For additional information, see *Subscriber Manager (SM) Fail-Over* (on page 2-10).

SM Management

SM management includes configuration, fault management, logging management, and performance management.

The SM is configured using the following:

- Configuration file (*p3sm.cfg*): For setting all configuration parameters of the Subscriber Manager.

  Note Changes that are made in the configuration file take affect only when the configuration file is loaded using the Command Line Utilities (CLU) or when the SM is restarted.

  For a detailed description of this file, see *Configuration File Options* (on page A-1).
- Command Line Utilities (CLUs): For ongoing subscriber management and monitoring of the SM. CLU commands are shell tools that can be used to manage subscribers, to install or update applications, to retrieve the user log, and to load the configuration file when updated.

  For a complete description of the Command Line Utilities, see *Command Line Utilities* (on page B-1).

The CLUs can be invoked locally, through a Telnet (or SSH) session to the SM hosting platform.

Use the SM user log files for logging, fault, and performance management. The log file contains information regarding system events, failures, and periodic system performance reports.

Subscriber Manager (SM) Fail-Over

The SM can be configured to operate with or without a cluster. The added functionality when operating in a *cluster topology* provides powerful new features such as *fail-over* and *high availability*. (For definition of italicized terms, see Glossary of Terms.)

This section describes topics that are related to using SMs together with clusters and redundancy, and is a supplement to the general introduction of the SM in the overall chapter *Introducing the Subscriber Manager* (on page 2-1).

As the Subscriber Manager plays a critical role in Cisco's ***SCAS BB*** and ***SCAS M*** solutions that are deployed in tier-one service provider environments, it also supports, starting with SM version 2.2, a fail-over operational mode. This feature minimizes system downtime that is caused by SM failure (as further discussed below in the *Overview* (on page 2-11) subsection).

This section introduces various concepts that are related to using a cluster of two SM nodes in a fail-over operational mode.

Note For the purposes of this section, it is assumed that the reader is familiar with the Veritas cluster technology.

Overview

The fail-over scheme that is implemented in the SM is based on the cluster technology of Veritas. The cluster includes two machines, each of them running SM TimesTen and Veritas software. The Veritas Cluster Server (VCS) software consolidates the SMs and exposes a single entity by providing a single virtual IP address for the entire cluster.

The cluster software distinguishes an active and a standby machine: the active machine ‘owns’ the virtual IP address (and all network connections), while the standby machine is passive until a fail-over occurs. At fail-over, the IP address is passed from the failing server to the backup server, which then becomes activated and re-establishes all network connections.

When a fail-over occurs, the LEGs lose their connection with the failed SM, and reconnect to the activated (backup) SM and retransmit their un-committed messages. The activated SM connects to the SCE Platform(s) and it performs an SE resynchronization.

The SM’s database is constantly replicated from the active node to the standby node by the TimesTen database replication agent. This enables a fast fail-over from one SM to another, since the subscriber data in the activated machine is always valid. The two SM nodes do not communicate except for passing the subscriber data.

The VCS uses software components called cluster agents to monitor and control the state of resources such as NICs, disks, IP addresses, and processes. Cisco supplies cluster agents to monitor the SM and the TimesTen database daemon and replication agent.

As part of the cluster operation, the TimesTen database daemon and replication agents are up and running regardless of the fail-over state. The SM Veritas agent monitors the daemon and the replication agent process. In case one of them fails, a fail-over takes place.

Note The SM software on both the active and the standby machines must be **configured identically**: apply the same configuration file and the same application PQI module to both machines.

These concepts are described in additional detail in the following subsections.

Normal Operation

The two SM nodes operate in hot-standby mode, where at any given time one node (the *active* node) receives and processes all the SM events, while the other node (the *standby* node) waits and is ready to go into operation on fail-over. For enabling seamless fail-over and for minimizing the fail-over time, the two SM nodes operate without an external storage device.

During the normal operation of the cluster, the active node (selected by the cluster):

- Performs all SM functionality of a non-cluster environment.
- Provides 'health' information for the cluster agent.
- Periodically replicates its subscriber database to the standby node.

On the standby node, both the SM and the Times-Ten software are running:

- The SM is fully configured - it is applied with the same configuration file and PQI application module as the active node - but does not interfere with the active node’s work.

- The SM connects to the TimesTen database, but does not connect to the LEG and the SCE devices.
- The Times-Ten software is operating as a replication client for the subscriber database, receiving and applying updates from the active node's TimesTen software.

Fail-Over Topology

The following figure depicts an SM cluster configuration in a topology with a redundant AAA server and two SCE2000s that are cascaded for redundancy.

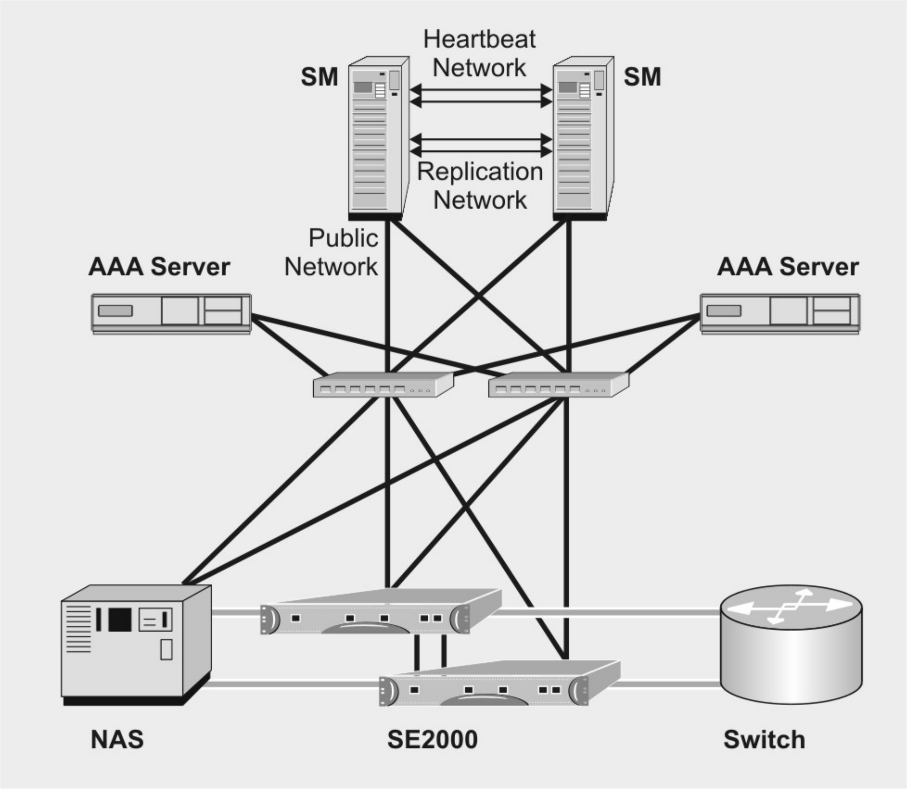


Figure 2-2: Cluster Topology

As already mentioned, an SM fail-over topology includes two SM nodes connected in a cluster scheme.

The two nodes are inter-connected by two dedicated (private) redundant networks:

- Heartbeat network: Used by the Veritas Cluster Server to perform cluster monitoring and control.
- Replication network: Used by the replication process to pass the subscriber records.

The two nodes should be located in the same site, where the heartbeat network is implemented using back-to-back connectivity between the two nodes or via redundant switches. Each node in the cluster has redundant network paths (NICs) connecting it to the all of the external entities that the SM communicates with (AAA, LEG, SE).

Each node in the cluster has (minimum of) six Ethernet NICs, where

- Two NICs are used for the (private) heartbeat network.
- Two NICs are used for the (private) replication network
- Two NICs are used for the public network (connectivity to SCEs and LEGs, and management of the SM).

The cluster has a virtual IP (VIP) used for communication with the external entities. Each node in the cluster has also an IP(s) address for administration of the node/cluster, as well as an IP(s) address for replication use.

Upon failure of the primary NIC of the public network, there is a fail-over to the secondary NIC on the same node, keeping the same IP addresses (VIP1), with no fail-over of the cluster. Upon failure of the primary NIC of the replication or heartbeat networks, there is fail-over to the secondary NIC on the same node, keeping the same IP addresses (VIP2 and VIP3), with no fail-over of the cluster.

The following diagram shows the usage of the regular and virtual IPs used in cluster configuration:

- IP1/IP2 and IP3/IP4 are used for administration of the respective nodes.
- VIP1 is used for the cluster IP for external clients over the public network.

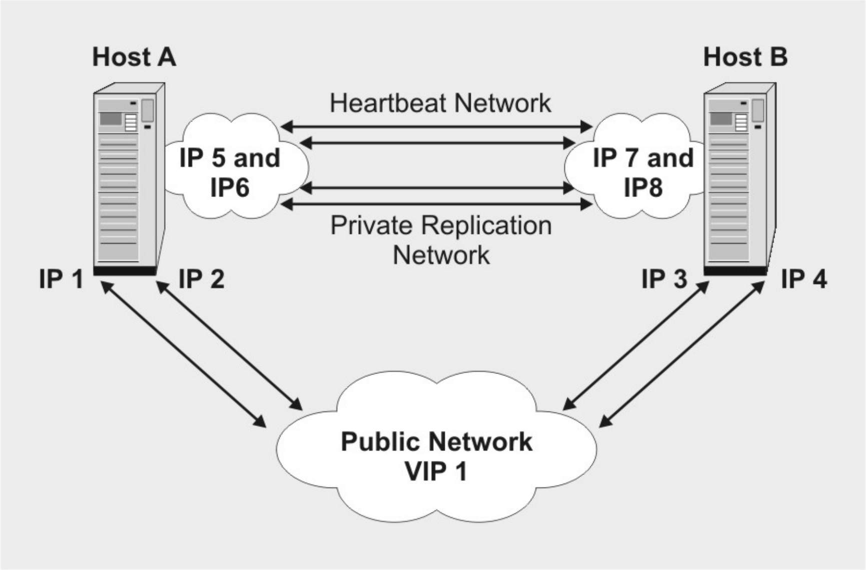


Figure 2-3: Regular and Virtual IPs in Cluster Configuration

For further information about replication IP configuration, see *Veritas Cluster Server Requirements and Configuration* (on page E-1).

Fail-over

Fail-over Operation

During normal operation, the Veritas Cluster Server mechanism automatically selects one of the SM servers to be active and the other to be standby.

The active SM server performs all the normal SM functionality. The two servers maintain the heartbeat mechanism between them, and the active server continuously replicates the subscriber database to the standby server’s database.

The standby SM server acts as a hot-standby machine, so it is completely ready for taking over (becoming activated) in a minimal fail-over time.

The following types of failures trigger the fail-over mechanism:

- SM application failure, including failure of the TimesTen database.
- Failure of the TimesTen daemon of the Times-Ten replication process.
- SUN server failure, due to failure of one of the resources of the server; for example, failure of both of the public network NICs.
- Manual activation of fail-over is also supported.

Note Communication failure does **not** cause a fail-over if there is a redundant NIC. Therefore, since each SUN machine has two NICs for connecting to external devices, a failure of one of the NICs merely causes switching to the redundant NIC, without activating the fail-over mechanism.

Once a failure is detected, the standby SM becomes activated, and the following occurs:

- The activated SM takes over the IP resources of the virtual IP mechanism.
- The LEGs reconnect to the activated SM.
- The activated SM creates IP connections with the SCE(s) and resynchronizes with them.
- The activated SM starts processing information that is sent from the different LEGs and forwarding it to the SCEs.

Recovery

Different types of failures require different triggering for the recovery procedure. Some failures may recover automatically (such as intra-node ports link-failure, which recover automatically when the link revives), others may need manual intervention.

Recovery may take place when an SM that experienced a failure is self-recovered or after it was replaced (if needed). The purpose of the recovery procedure is to take the cluster back to a fully functional mode. When the recovery procedure ends, the behavior is the same as it was after installation.

The failed SM server is recovered manually or automatically, according to the type of failure that occurred. The recovery procedures, and when they are used, are described in the following subsections.

Machine Reboot

Recovering from a machine reboot is a fully automatic recovery process, where the failed SM server reboots, and after establishing a connection with the other server, and synchronizing the databases, the cluster of the two SM servers is ready again for fail-over operation.

Automatic steps:

Step 1 The reboot process is run on the node.

Step 2 VCS makes the node standby.

Step 3 The node boots.

Step 4 VCS establishes intra-node communication and the new node joins the cluster.

Step 5 The TimesTen database replication process is started from the point before the reboot.

Step 6 The SM in the recovered server is ready after the database recovery process is running and the SM moves from Init state to Standby state.

Replacing the Server

Replacing the server is needed when the machine has an unrecoverable physical failure. The server is replaced with a new machine that is installed with fresh SM, TimesTen, and VCS installations.

Replacing the server is a manual recovery, where the failed SM server is physically replaced. After the new SM server machine is connected to the network and configured, and the two databases synchronize, the cluster of the two SM servers is ready again for fail-over operation.

Manual steps:

Step 1 Connect a new server to the inter-node ports and intra-node ports (but leave the network ports disconnected).

Step 2 Basic network and cluster configurations: do manually (the first time).

Step 3 Copy the configuration files from the active node.

If only the SM configuration file needs replication, use the CLU command `p3sm --load-config --remote=<NEW-SM_IP>`.

Step 4 Perform the TimesTen database duplication operation (see *Database Duplication Recovery* (on page 2-16)).

Step 5 Start the VCS operation on the recovered node.

Step 6 Connect the network ports.

Automatic steps:

Step 1 The SM in the recovered server is ready after the database recovery process is completed and the SM moves from Init state to Standby state.

Database Duplication Recovery

Database duplication recovery is a manual recovery, which is needed when the standby node's database loses synchronization with the active node’s database. Loss of synchronization can occur when one of the SM machines is replaced or when the replication process on the active node fails to replicate all of the data inserted to its database (replication NICs were disconnected).

Manual steps (all performed on the standby node):

Step 1 Stop the cluster server’s monitoring of the resources. Use the VCS CLU `hastop -local` to stop the VCS.

Step 2 Stop the SM, so it will not be affected by database being cleared. Use the CLU command `p3sm --stop`.

Step 3 Stop the replication agent. Use the CLU command `p3db –-rep-stop`.

Step 4 Destroy the database. Use the CLU command `p3db –-destroy-rep-db`.

Step 5 Duplicate the remote database. Use the CLU command `p3sm –-duplicate--local LOCAL_HOSTNAME --remote REMOTE_HOSTNAME`.

Step 6 Start the cluster server’s monitoring of the resources (use the VCS CLU `hastart`), which will automatically start the replication process and the SM.

Management

The configuration of the two SM servers is performed using Command Line Utilities and a configuration file (see *Configuration and Management* (on page 4-1) and *Configuring a Subscriber Management Solution* (on page 4-1)). The actual configuration is performed for the active SM and then manually replicated for the standby SM.

To perform configuration duplication:

Step 1 Establish an FTP connection between the active and standby machines.

Step 2 Copy the configuration file from *`~pcube/sm/server/root/config/p3sm.cfg`* on the active node to the standby node, and apply the SM configuration file by using the CLU command `p3sm –-load-config`.

Alternatively, you can replicate the SM configuration file to the standby machine by running the CLU command `p3sm –-load-config --–remote=<NEW-SM_IP>` on the active machine.

Step 3 Copy the application PQI file you installed on the active node to the standby node.

Step 4 Install the PQI file. Use the CLU command `p3inst --install -f <PQI file path>`.

Step 5 If you have made changes in the database-related configuration files, copy the files to *`/etc/system`* (for **Solaris**) or to /etc/sysctl.conf (for **Linux**), and *`/var/TimesTen/sys.odbc.ini`* from the active node to the standby node.

Note If this step (copying files) is performed, a reboot of the standby node is required.

Note If the database is located in different directories in the two nodes, then the files *`sys.odbc.ini`* are not identical and the actual parameter changed in the file must be copied.

All the configuration and administration of the Veritas Server Cluster is performed using Veritas tools.

Notifications are enabled through SNMP traps that the Veritas Cluster Server provides. The Cluster Server supports SNMP traps such as:

- Fatal failure detected (local or remote)
- Secondary node starts fail-over procedure
- Secondary node is operational (end of fail-over)

CHAPTER 3

Installation and Upgrading

This chapter describes how to install the Cisco Service Control Management Suite Subscriber Manager (SCMS SM), as well as how to upgrade and uninstall. This chapter also discusses topics related to installation, upgrading and uninstalling.

This chapter contains the following sections:

Installation Overview

Installing the SM is an automated process. It consists of executing an installation script residing on the root of the SM CD-ROM supplied by Cisco.

Note For **Solaris**: the procedure also requires modifying the */etc/system* file. This can be done manually or by using some other automated utility.

Note For **RedHat**:The procedure also requires modifying the */etc/sysctl.conf* file. This can be done manually or by using some other automated utility.

The procedure can install the following components:

- SM and Command Line Utilities (CLUs)
- TimesTen database and DSN
- Java Runtime Environment (JRE)
- SM's Veritas Cluster Agents

The procedure also includes:

- Setting up a **pcube** user and group
- Adding startup and shutdown scripts
- System configuration for TimesTen (performed manually or using a script)

After completing installation and configuration, you can use the SM to introduce subscribers to the system.

Contents of CDROM

The following table lists the contents of the SM installation CD-ROM.

Table 3-1 Contents of SM CD-ROM

Path	Name	Description
CD_ROOT		Cross-platform files
	cnr-leg-dist.tar.gz	CNR LEG distributions
	dhcp_forwarder.tar.gz	DHCP Forwarder distribution
	hooks.sh	User defined function for upgrade
	install-dsn.sh	TT DSN configuration script
	install-forwarder.sh	DHCP Forwarder installation script
	installjava.sh	JRE installation script

Path	Name	Description
	install-sm.sh	SM installation script
	install-tt.sh	TimesTen installation script
	install-vcs-agents.sh	VCS agents installation script
	leaseq.leg.tar.gz	DHCP Lease Query LEG distribution
	MANIFEST	CD information
	p3sm.sh	Startup and shutdown script
	pqb_resp_uninst.txt	Response file for TT uninstall
	pqb-odbc-ini.txt	ODBC definitions
	pqb-response50.txt	Response file for TT installation
	pqb-sys-odbc-ini.txt	ODBC definitions
	rdrdhcp-leg.tar.gz	RDR DHCP LEG distribution
	se_agent_VV_BBB.pkg	EM agent distribution
	sm-c-api-dist.tar.gz	C API distributions
	sm-common.sh	General installation script
	sm-dist.tar.gz	SM distribution
	sm-java-api-dist.tar.gz	SM Java API distribution
	tt-sysconf.sh	TT system configuration script
	upgrade-sm.sh	SM upgrade script
	vcs-agents-dist.tar.gz	VCS agents distribution
CD_ROOT		Platform-specific files
	linux-def.sh	Linux-specific definitions
	solaris-def.sh	Solaris-specific definitions
CD_ROOT /Java/		Java Runtime Environment files
	j2re1.4.2_05-linux.tar.gz	JRE for Linux
	j2re1.4.2_05-solaris.tar.gz	JRE for Solaris
	LICENSCE	JRE license
CD_ROOT /TimesTen/		TimesTen files
	TT5035LinuxRH32.tar.Z	TimesTen for Linux
	TT5035Sparc32.tar.Z	TimesTen for Solaris (32-bit)
	TT5035Sparc64.tar.Z	TimesTen for Solaris (64-bit)
CD_ROOT/sm_doc/		Documentation files
	CNR_LEG.pdf	CNR LEG Documentation
	DHCP_Lease_Query.pdf	DHCP Lease Query LEG Documentation
	RDR_DHCP_LEG.pdf	RDR DHCP LEG Documentation

Path	Name	Description
	Radius_Listener.pdf	Radius Documentation
	SM_C_API.pdf	SM C API Documentation
	SM_Java_API.pdf	SM Java API Documentation
	SM_User_Guide.pdf	SM User Guide

Documentation

The SM installation CD-ROM contains the following documents:

- **Manifest** – contains the version/build numbers for all components from which the CD-ROM was built
- **User Guide** – the SCMS SM User Guide in PDF format
- **SCMS SM C/C++ API Programmer's Guide**
- **SCMS SM Java API Programmer's Guide**
- **SCMS SM CNR LEG Reference Guide**
- **SCMS SM Radius Listener LEG Reference Guide**
- **SCMS SM RDR DHCP Listener LEG Reference Guide**
- **SCMS SM DHCP Lease Query LEG Reference Guide**

System Requirements

The SM can be installed on the following platforms:

- **Solaris:** SUN SPARC machine running Solaris. . See below, Table 4-2, Minimal System Hardware Requirements, and Table 4-3, Solaris Minimal System Software Requirements.
- **Red Hat:** Machine with Intel-based processor running Linux. See below, Table 4-2, Minimal System Hardware Requirements, and Table 4-4, Red Hat Minimal System Software Requirements.

The machine should conform to the system requirements listed in the following tables.

Note The specifications listed below are minimal. They should be verified in order to guarantee specific performance and capacity requirements.

Table 3-2 Minimal System Hardware Requirements

Item	Requirement
CPU	• SUN SPARC, minimum 500 MHz (for Solaris) • INTEL processor, minimum 1 GHz (for Linux Red Hat)

Item	Requirement
RAM	Minimum 1 GB; see table in RAM and Configuration Parameters Versus Number of Subscribers
Free Disk Space	Minimum 3 GB total, of which: • Minimum 1 GB free on partition where VARDIR (SM database repository) is installed • Minimum 0.5 GB free on partition where PCUBEDIR (SM files) is installed • Minimum 200 MB free on partition where `/tmp` is mounted
Network Interface	Depends on whether or not the configuration includes a cluster: • **No** cluster: One (**1**) 100baseT Ethernet • **With** cluster: Six (**6**) 100baseT Ethernet
CD-ROM drive	Recommended

Note For the hardware and software system requirements for the Veritas Cluster Server, see *Veritas Cluster Server System Requirements* (on page E-1).

Table 3-3 Solaris System Software Requirements

Item	Requirement
OS	Solaris 5.8 64-bit build 04/01 or later; currently, only Solaris 5.8 is supported Solaris Core Installation
System Packages	Mandatory: • SUNWbash – GNU Bourne-Again shell (bash) • SUNWgzip – GNU Zip (gzip) compression utility • SUNWzip – Info-Zip (zip) compression utility • SUNWlibC – Sun Workshop Compilers Bundled libC • SUNWlibCx – Sun WorkShop Bundled 64-bit libC Optional: • SUNWadmap – system administration applications • SUNWadmc – system administration core libraries

It is strongly recommended that the latest patches from SUN be applied; see *http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/xos-8&nav=pub-patches* (http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/xos-8&nav=pub-patches)

Table 3-4 Red Hat System Software Requirements

Item	Requirement
OS	Red Hat Enterprise Linux AS/ES 3.0 Red Hat Core Installation
System Packages	Mandatory: • GNU Bourne-Again shell (bash-2.05b-29.i386.rpm) • GNU Data Compression Program (gzip-1.3.3-9.i386.rpm) • File compression and packaging utility (zip-2.3-16.i386.rpm) • Standard C++ libraries for Red Hat Linux 6.2 backward compatibility (compat-gcc-7.3-2.96.122.i386.rpm) For integrating with the C API: • GNU cc and gcc C compilers (gcc-3.2.3-20.i386.rpm) • C++ support for the GNU gcc compiler (gcc-3.2.3-20.i386.rpm)

It is strongly recommended that the latest patches from Red Hat be applied.

Installation Procedures

All installations can be performed by executing one of the installation scripts located on the **root** of the SM CD-ROM. The CD-ROM drive must be mounted prior to performing the installation. You can choose to mount the CD-ROM on a remote machine, and make its contents available to the target machine via your network.

Note **Solaris:** It is important when mounting the SM CD-ROM on a Solaris machine, the mount option **nomaplcase** must be used.

You can choose to install the SM, TimesTen, and Java separately. In most cases, the SM installation script is the only script needed for completing the installation.

Each installation script displays messages describing the significant steps that are being performed. These messages are also sent to the system log (see *Logging Script Messages* (on page 3-19)) for future reference.

If you try to install SM on a machine on which the SM is currently running, or to a directory on which the SM is already installed (even if not running), the operation will fail and you will be requested to *upgrade* ("Upgrading" on page 3-21).

Typical Installation

This section assumes that you want to install the SM and TimesTen components, as well as the Java Runtime Environment, and that you want to perform the necessary system configurations in order for these components to work.

The installation is performed in four steps, which are further described in the following sections:

Step 1 Executing the *install-sm.sh* script. This script must be invoked by the **root** user.

Step 2 Setting a password for the user **pcube**.

Step 3 Configuring the system for TimesTen.

Step 4 Rebooting the computer.

Note In a *high availability* setup (see SM Cluster), you must *install SM's custom VCS agents* ("Installing SM Cluster Agents" on page 3-18).

Step 1 - Executing the install-sm.sh Script

Note Before starting the installation, make sure that disk space requirements listed in *System Requirements* (on page 3-4) are satisfied.

To execute the install-sm.sh script:

➲ From your workstation shell prompt, use the following command:

```
# install-sm.sh -d <install directory>
```

where <install directory> is the location in which you want the SM to be installed.

If you use the default install location, */opt/pcube*, you need not specify the **-d** flag.

For additional information regarding the script operation, see *Installing SM* ("Installing the SM" on page 3-13).

Step 2 - Setting the Password for the pcube User

After the installation script completes, set the password for the **pcube** user by running the following command:

```
# passwd pcube
```

Note It is important to make a note of the password you select.

Step 3 - Configuring the System Memory Settings

System Memory Settings Versus Number of Subscribers

System memory configuration requirements should be set according to the maximum number of subscribers. Recommended memory configuration values, based on the number of supported subscribers, are listed in the following table.

Table 3-5 RAM and Memory Configuration Parameters Versus Number of Subscribers

Maximum Number of Subscribers	Required RAM	SM Process Memory Setting	(TimesTen Memory Settings) Shared Memory	(TimesTen Memory Settings) PermSize	(TimesTen Memory Settings) TempSize
100,000	2 GB	256 MB	512 MB	200 MB	100 MB
500,000	2 GB	256 MB	1.45 GB	1 GB	250 MB
1,000,000	3 GB	512 MB	2 GB	1.5 GB	250 MB
2,000,000	4 GB	768 MB	2.85 GB	2.4 GB	250 MB
3,000,000	5 GB	1024 MB	4 GB	3.55 GB	250 MB
4,000,000	7 GB	1280 MB	5.05 GB	4.6 GB	250 MB

Description of the table columns:

- Maximum Number of Subscribers – The maximum number of subscribers that the SM has to support.
- Required RAM – The RAM requirement for the machine running the SM.
- SM Process Memory Setting – The required memory configuration for the SM process itself. For additional information about the SM process memory configuration, see *Configuring SM Process Memory Settings* (on page 3-11).
- TimesTen Memory Settings – Three values (Shared Memory, PermSize, TempSize) that specify the configuration required for TimesTen to run correctly. For additional information, see Configuring the TimesTen Memory Settings.

If the maximum number of subscribers you require is not listed in the above table, use the settings specified for the next higher value of Maximum Number of Subscribers. For example, for 1,200,000 subscribers, use the values specified for 2,000,000 subscribers (4 GB of RAM, etc.).

Configuring the System for TimesTen

TimesTen requires that certain changes be made in the Solaris kernel configuration file */etc/system*. These changes increase the shared memory and semaphore resources on the Solaris machine from their defaults. For additional information regarding these changes, please refer to the TimesTen documentation.

Making the changes automatically:

Note It is recommended that you review the */etc/system* file before running the *tt-sysconf.sh* script, because the script overwrites the current file settings with the values listed in Making the changes manually. If you want to keep some or all of the current file settings, edit */etc/system* and perform the changes manually.

To make the required changes automatically, run the *tt-sysconf.sh* script file. This script file must be invoked by the **root** user, without arguments, as follows:

```
# tt-sysconf.sh
```

Making the changes manually:

To make the required changes manually, add the following lines to */etc/system file*:

```
*  ---- Begin settings for TimesTen
set semsys:seminfo_semmni = 20
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semmns = 2000
set semsys:seminfo_semmnu = 2000
set shmsys:shminfo_shmmax = 0x20000000
* ---- End of settings for TimesTen
```

Configuring the System Kernel Configuration File

TimesTen requires that certain changes be made in the operating system kernel configuration file:

- For **Solaris**, modify file `/etc/system`.
- For **Linux**, modify file `/etc/sysctl.conf`.

These changes increase the shared memory and semaphore resources on the machine from their defaults. For additional information regarding these changes, please refer to the TimesTen documentation.

Making the changes automatically:

Note It is recommended that you review the system configuration file before running the *tt-sysconf.sh* script, because the script overwrites the current file settings with the values listed below, in *Making the changes manually*. If you want to keep some or all of the current file settings, edit the configuration file by performing the changes manually.

- To make the required changes automatically, run the *tt-sysconf.sh* script file. This script file must be invoked by the **root** user, without arguments, as follows:

```
# tt-sysconf.sh
```

Making the changes manually:

Editing the configuration file manually is required in case you require support for more than 100,000 subscribers in the SM. Only the shared memory size is affected by your system’s sizing requirements. To determine the correct configuration values for your system, see the table in *System Memory Settings Versus Number of Subscribers* (on page 3-8).

- For **Solaris**, make the required changes manually by adding the following lines to file */etc/system* and configuring the shared memory size:

```
*---- Begin settings for TimesTen
set semsys:seminfo_semmni = 20
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semmns = 2000
set semsys:seminfo_semmnu = 2000
set shmsys:shminfo_shmmax = 0x20000000
*---- End of settings for TimesTen
```

- For **Red Hat** (Linux), make the required changes manually by adding the following lines to file */etc/sysctl.conf* and configuring the shared memory size:

```
*---- Begin settings for TimesTen
kernel.shmmax = 536870912
kernel.sem = “SEMMSL_250 SEMMNS_32000 SEMOPM_100 SEMMNI_100
*---- End of settings for TimesTen
```

Configuring /var/TimesTen/sys.odbc.ini

Some installations might require changing TimesTen parameters, as listed below, so that the database will run as desired. However, if the default values suit your requirements then no changes are needed.

Setting the multi-processor optimization:

If your system is a multi-processor machine set the value of the *SMPOptLevel* parameter of the Pcube_SM_Repository in file `sys.odbc.ini`; to 1. Otherwise leave it set to 0 or not set at all.

```
SMPOptLevel=1
```

Setting the database size:

If your system needs to support more than 100,000 subscribers, set the values of parameters *PermSize* and *TempSize* of the Pcube_SM_Repository in file *sys.odbc.ini*; see the table *in System Memory Settings Versus Number of Subscribe*rs (on page 3-8).

For example:

```
PermSize=500
TempSize=150
```

Note **Solaris:** Remember to set the value of parameter *shmsys:shminfo_shmmax* in the */etc/system* file to be larger than the sum of *PermSize* and *TempSize*.

Note **Red Hat:** Remember to set the value of parameter *kernel.shmmax* in the `/etc/sysctl.conf` file to be larger than the sum of *PermSize* and *TempSize*.

Configuring SM Process Memory Settings

The SM process uses by default 256 MB of RAM memory. However, in certain configurations of application components the SM process needs to be allocated additional memory to work correctly. Setting an environment variable called *PCUBE_SM_MEM_SIZE* with the desired memory size (in megabytes) instructs the SM start-up scripts to allocate that defined memory size for the SM process.

You can set the memory size value for this environment variable for the user **pcube**, or you can configure the desired process memory size in the `sm.sh` file located in the root directory of the user **pcube** (*~*`pcube/sm.sh`).

The following example, which shows a line in the `sm.sh` file, defines a memory size of 512 MB for the SM process:

```
PCUBE_SM_MEM_SIZE=512
```

Note To prevent performance degradation due to memory swap, make sure that the machine has enough RAM memory for the SM process, the SM database, and all of the other application running on this machine.

Note To determine the correct memory values for your installation, see the table in *System Memory Settings Versus Number of Subscribers* (on page 3-8).

Step 4 - Rebooting

Reboot the computer to complete the installation.

Verifying That Installation Was Successful

To verify that the installation was successful, run a CLU utility (such as **p3sm**) that displays general information about the SM.

To verify that the SM installation was successful:

➲ From your workstation shell prompt, type:

```
# p3sm --sm-status
```

The above command displays the current status of the SM.

Note Wait a few minutes after installation before running this command, to allow the SM to become operational.

The output of this command should indicate that the SM is running.

In case of errors during installation the command will output a description of these errors.

Configuring the SM

After installing the SM, you may want to configure it to your specific needs. In particular, the following parameters should be addressed at this point:

- topology – cluster or standalone
- introduction_mode – pull or push
- support_ip_ranges – whether IP-ranges should be used in the installed setup

To configure the SM, first edit its configuration file, *p3sm.cfg*, using any standard text editor. (The configuration file is described in detail in *Configuration and Management* (on page 4-1), and in *Configuration File Options* (on page A-1).) After you finish editing the *p3sm.cfg* configuration file, use the p3sm utility for updating the SM with the new settings:

To load the SM with a new configuration file (*p3sm.cfg*):

➲ From your workstation shell prompt, type:

```
# p3sm --load-config
```

The configuration file is loaded, and the SM configuration updated accordingly.

Additional Installation Procedures

The following procedures complement the ones described in Typical Installation:

- *Installing an SM Cluster* (on page 3-13) – should be used if installing two SM nodes for the first time.
- *Installing the SM* (on page 3-13) – should be used if the user **pcube** is already defined in the machine, or the TimesTen or the DSN or the JRE are installed in a non-default directory.
- *Installing TimesTen* (on page 3-15) – should be used if the TimesTen database utilities must be placed in a special directory.
- *Installing the DSNs* (on page 3-16) – should be used if the TimesTen database must be placed in a special directory.
- *Installing Java* (on page 3-16) – should be used if the JRE must be placed in a special directory.

Note If you plan on installing Java in a special directory, you **must** set your path to the installation directory of Java.

- *Installing SM's Cluster Agents* ("Installing SM Cluster Agents" on page 3-18) – must be installed when in a High Availability setup where Veritas Cluster Server (VCS) is used.

Note It is strongly recommended **not** to install Java manually. The SM installation script automatically installs Java under the user **pcube** home directory, so it will not affect any existing Java installations.

Note It is recommended to reboot the machine after running **any** of the additional installation procedure(s), especially after installing the SM.

Installing an SM Cluster

The installation of an SM cluster is very similar to installing the SM on two machines.

The following procedure describes the installation:

Step 1 Before installing the SM cluster you must first install Veritas Cluster Server software on both nodes. Please see the *VCS Installation Guide*.

Step 2 Install the SM on both machines, as described in *Installing the SM* (on page 3-13).

Step 3 Configure the SM topology parameter to the cluster, as describer in *Configuring the SM* (on page 3-12).

Step 4 Install P-Cube's VCS agents, as described in *Installing SM's Cluster Agents* ("Installing SM Cluster Agents" on page 3-18).

Step 5 Configure the VCS, as described in *Veritas Cluster Server Requirements and Configuration* (on page E-1).

Installing the SM

Note This installation is customizable.

To execute the install-sm.sh script:

➲ From your workstation shell prompt, use the following command:

```
# install-sm.sh [command-options]
```

The following table lists the command options.

Table 3-6 Options for install-sm.sh

Options	Description
-d	Specifies the install directory for ~pcube. This directory must **not** be an existing directory. This directory must be specified as a full pathname that begins with '/'. Default: */opt/pcube*
-o	Specifies the existing home of user pcube as the install directory. Note that the options **–d** and **–o** are mutually exclusive.
-v	Specifies the directory for data storage. This directory must **not** be an existing directory. This directory must be on a partition with at least 1 GB of free space. This directory must be specified as a full pathname that begins with '/'. Default: *InstallDirectory/var*
-32	Forces the installation of a 32-bit SM version on Solaris. Default SM installation option for Solaris is 64-bit option. Default SM installation option for Linux is 32-bit option.
-j	Specifies not to call the JRE installation script.
-n	Specifies not to call the TimesTen installation script.
-m	Specifies not to call the TimesTen DSN installation script.
-h	Prints a help message and exits.

The script performs the following steps:

- Checks for validity of arguments and sufficient disk space.
- Adds (or verifies the existence of) a user **pcube** and a group **pcube**.
- Populates the **pcube** home directory with SM and CLU directory structure.
- (Optional) Invokes the JRE installation script with **pcube** home as the target directory.
- (Optional) Invokes the TimesTen installation script with **pcube** home as the target directory.
- (Optional) Creates the SM DSN for TimesTen with **pcube** home as the target directory.
- Creates startup and shutdown scripts in */etc*.
- Creates the shell preamble *~pcube/sm.sh*, which contains environment variables that depend on the actual folder in which the SM was installed.

Example 1:

The following example installs only the SM and CLU components, but disables automatic installation of the TimesTen database and DSN.

The installation directory is the existing *pcube* home directory, using the default data storage directory definition.

```
# install-sm.sh -o –n -m
```

Example 2:

The following example installs the SM and CLU to a directory named */usr/local/pcube*, using the default data storage directory definition.

Automatic installation of the TimesTen database and DSN is enabled.

```
# install-sm.sh –d /usr/local/pcube
```

Installing TimesTen

TimesTen is installed by invoking the *install-tt.sh* script. This script is invoked automatically by the *install-sm.sh* script unless the **–n** switch is specified when invoking the *install-sm.sh* script. This following procedure describes how to invoke the *install-tt.sh* script manually.

To install the TimesTen database:

➲ From your workstation shell prompt, type:

```
# install-tt.sh <installation_directory>
```

where <Installation_Directory> specifies where to install TimesTen. It is recommended (but not required) to install TimeTen in the home directory of the **pcube** user.

Table 3-7 Options for install-tt.sh Script

Options	**Description**
-32	Forces the installation of a 32-bit TimesTen version on Solaris. The default TimesTen installation option for Solaris is the 64-bit option. The default TimesTen installation option for Linux is the 32-bit option.

The script performs the following steps:

- Checks for validity of arguments and sufficient disk space.
- Computes a CRC sum of the TimesTen distribution file to verify its validity.
- Extracts the TimesTen distribution archive.
- Invokes the TimesTen installer in batch mode using a constructed response file.
- Inspects the status returned from the installer to verify that the installation succeeded.
- Creates symbolic links for TimesTen classes and libraries in *~pcube/{classes,lib}.*

Installing the DSNs

The TimesTen DSNs for the SM are installed by invoking the *install-dsn.sh* script. This script is invoked automatically by the *install-sm.sh* script, unless the **-m** switch is specified when invoking *install-sm.sh*. This section describes how to invoke the *install-dsn.sh* script manually.

To install the TimesTen DSNs:

➲ From your workstation shell prompt, type:

```
# install-dsn.sh <storage_directory>
```

where <storage_directory> specifies the location of the DSNs permanent storage. It is recommended (but not required) to install the TimeTen DSNs in the home directory of the **pcube** user.

The script performs the following steps:

- Checks for validity of arguments and sufficient disk space (1 GB).
- Backs up any existing DSN definition in *~pcube* (no merge is performed).
- Builds DSNs with the names *Pcube_SM_Repository* and *Pcube_SM_Local_Repository* for the specified storage directory.

Note The script does not actually create the DSNs. It specifies the **AutoCreate** attribute in the TimesTen DSN definitions file.

Installing Java

Note It is strongly recommended **not** to install Java manually. The SM installation script automatically installs Java under the user **pcube** home directory, so it will not affect any existing Java installations.

The installation CD-ROM contains the Java Runtime Environment (JRE) version 1.4.2_05 for Solaris and Linux (repackaged), as well as the *installjava.sh* script for automating the installation process. This script is invoked automatically by the *install-sm.sh* script, unless the **-j** switch is specified when invoking *install-sm.sh*.

The remainder of this section describes how to invoke the *installjava.sh* script manually.

To install Java:

➲ From your workstation shell prompt, type:

```
# installjava.sh  [-d <installation_directory>] [-o] [-h]
```

Table 3-8 Options for installjava.sh

Options	Description
-d	Specifies the installation directory for the Java Runtime Environment. This directory must be an existing directory. This directory must be specified as a full pathname that begins with '/'. Default: *~pcube*
-o	Specifies the existing home of user **pcube** as the install directory. **Note:** The **-d** and **-o** options are mutually exclusive.
-h	Prints a help message and exits.

Note The file *installjava.sh* must be invoked only **after** the SM is installed successfully (by *install-sm.sh*), otherwise an error message appears.

The script performs the following steps:

- Computes a CRC sum of the JRE distribution file to verify its validity.
- Launches the self-extracting package under *installation directory*.

Note The file *installjava.sh* must be invoked only **after** the SM is installed successfully (by *install-sm.sh*), otherwise an error message appears.

Note If you plan on installing Java in a special directory, you **must** set your path to the installation directory of Java.

Installing SM Cluster Agents

The installation CD-ROM contains a set of customized Veritas Cluster Agents for supporting monitoring and controlling of SM-related resources in Cluster topology. The cluster agents must be installed under the VCS *bin* directory.

To install VCS agents:

➲ From your workstation shell prompt, type:

```
# install-vcs-agents.sh [command-options]
```

The following table lists the command options.

Table 3-9 Options for install-vcs-agents.sh

Options	Description
-d	Specifies the installation directory for the agents, which must be the *bin* directory of the VCS. This directory must be an existing directory. This directory must be specified as a full pathname that begins with ‘/’. Default: */opt/VRTSvcs/bin*
-h	Prints a help message and exits.

The script performs the following steps:

- Checks that the installation directory exists.
- Extracts the agent distribution file to the specified directory.
- Copies the VCS default-script-agent-executable from the installation directory to all agent directories.

Troubleshooting the Installation

For troubleshooting the installation, see *Troubleshooting* (on page D-1).

Installing an Application

An application can be installed on the SM in order to customize the components. You can also upgrade an existing application to a new version, or return to a previous version (rollback) of an application. Use the **p3inst** utility to install or uninstall an application (*pqi* file).

For additional details of how to install a specific application (such as SCAS BB or SCAS M), refer to that application's Installation Guide.

To install/uninstall an application:

➲ From your workstation shell prompt, use a command having the following general format:

```
# p3inst <operation> <filename> [<installation/upgrade parameters>]
```

The following table lists the **p3inst** operations.

Table 3-10 p3inst Operations

Option	Description
--install	Installs the specified application *pqi* file to the SM. It may be necessary to specify arguments for the installation procedure in the command line.
--uninstall	Uninstalls the specified application from the SM.
--upgrade	Upgrades the specified application. It may be necessary to specify arguments for the upgrade procedure in the command line.
--rollback	Returns the specified application to the previous version.
--describe	Displays the contents of the specified application file.
--show-last	Lists the last installed *pqi* file

Example 1: Install

The following example shows how to install the specified installation file to the device.

```
# p3inst --install --file=eng216.pqi
```

Example 2: Uninstall

The following example shows how to uninstall the specified installation file from the device.

```
# p3inst --uninstall –-file=oldInstallation.pqi
```

System Changes Made by Installation Scripts

This section lists the system changes applied automatically by the SM installation. The SM installation adds a dedicated user and group, as well as startup and shutdown scripts.

Logging Script Messages

- For **Solaris**: The installation scripts log all their messages into the system log, which is usually the file located at */var/adm/messages*. The messages are logged to the *user.info* syslog category.

- For **Red Hat**: The installation scripts log all their messages into the system log, which is usually the file located at */var/log/messages*. The messages are logged to the *user.info* syslog category.

pcube User and Group

During installation, a user **pcube** is created (unless it already exists) with its own group. This user owns all installed SM and CLU files. The user's home directory is the installation directory selected during installation. For security, the user is initially created with a locked password. A new password must be assigned.

Startup and Shutdown Scripts

The SM is started on boot to run level 2, and is stopped when leaving this run level (for example, when the machine is shut down).

The installer installs the following files for startup and shutdown:

- For **Solaris**:

```
-rwxr--r--   1 root   other   /etc/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc0.d/K44p3sm ->
/etc/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc1.d/K44p3sm ->
/etc/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc2.d/S92p3sm ->
/etc/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rcS.d/K44p3sm ->
/etc/init.d/p3sm
```

- For **Red Hat**:

```
-rwxr--r--   1 root   other   /etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc.d/rc0.d/K44p3sm ->
/etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc.d/rc1.d/K44p3sm ->
/etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc.d/rc2.d/S92p3sm ->
/etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc.d/rc3.d/S92p3sm ->
/etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other   /etc/rc.d/rc5.d/S92p3sm ->
/etc/rc.d/init.d/p3sm
lrwxrwxrwx   1 root   other  /etc/rc.d/rc6.d/K44p3sm ->
/etc/rc.d/init.d/p3sm
```

The TimesTen installer creates similar startup and shutdown scripts.

Upgrading

SM 2.5 supports several types of upgrade procedures, according to the SM version that was previously installed and the requirement (or lack of requirement) for fail-over in the new installation.

Note that in some topologies, such as cluster, the upgrade procedure does not consist of simply running a script, because it is required to have minimum down time.

The following sections describe the various procedures needed for upgrading in different topologies and for moving from one topology to the other.

Performing the Upgrade

SM 2.5 provides an upgrade script that implements an upgrade from version 1.5.X functionality. The upgrade procedure script preserves the subscriber database and the entire SM configuration, including network elements, domains, and application-specific components.

dd

Note For **SOLARIS**: Previous versions of the SM on Solaris used a 32-bit JVM and database. SM 2.5 is by default installed with a 64-bit JVM and database, so you can choose whether or not to upgrade to 64-bit (the default is **not** to upgrade to 64-bit).

Note For **RED HAT**: Upgrades on Linux systems are only between SM 2.5.X releases. The Linux platform is used **only** with a 32-bit JVM and database.

For additional information, see the following sections:

- *Upgrading from Non-cluster Setup to Non-cluster Setup* (on page 3-23)
- *Upgrading from Non-cluster Setup to Cluster Setup* (on page 3-23)
- *Upgrading from a Cluster Setup* (on page 3-24)

To execute the upgrade-sm.sh script:

➲ From your workstation shell prompt, use the following command:

```
# upgrade-sm.sh [command-options]
```

The following table lists the command options.

Table 3-11 Options for upgrade-sm.sh Script

Options	Description
-d	Destroy the database during the upgrade of TimesTen.
-64	Force upgrade to 64-bit version (32-bit systems are upgraded automatically to 32-bit).
-p	Pause the upgrade for PQI installation.
-h	Shows this message.

The script performs the following steps:

Step 1 Detects existing MSW version.

Step 2 Detects new MSW version.

Step 3 Verifies that java is installed on the machine.

Step 4 Verifies that the user **pcube** exists.

Step 5 Verifies that an SM of version 1.5 and up is present on the system.

Step 6 For SM 2.2 and up, stops LEG updates.

Step 7 Backs up existing contents of the subscriber database to a CSV file. This step is performed only when upgrading from SM 2.2 or earlier, or when upgrading from a 32-bit version to a 64-bit version.

Step 8 Stops the current SM (if running).

Step 9 Removes the TimesTen database (unless specified otherwise).

Step 10 Backs up SM configuration files.

Step 11 Installs the updated versions of SM and (unless specified otherwise) TimesTen.

Step 12 Invokes a separate program for upgrading the SM and database configuration files.

Step 13 Starts the upgraded SM.

Step 14 If pause option was specified, pauses execution to let the operator install PQIs, etc.

Step 15 Restores the backed up contents of the subscriber database (unless specified otherwise).

Example:

The following example upgrades the SM, keeps the current database, and does not pause the upgrade for PQI installation.

```
# upgrade-sm.sh
```

The following example upgrades the SM, removes the current database, and pauses the upgrade for PQI installation.

```
# upgrade-sm.sh –d -p
```

Note No SM reboot is required after the upgrade procedure.

Data Duplication Procedure

The data duplication procedure enables the user to duplicate (copy) the entire database from one machine to the other, and then keep the databases synchronized by running the replication agent at the end. This procedure is used in some of the upgrade procedures described in the following sections.

For details of the procedure, see *Database Duplication Recovery* (on page 2-16).

Upgrading from Non-cluster Setup to Non-cluster Setup

To upgrade from a non-cluster setup to a non-cluster setup, run the SM upgrade script on the SM (see *Performing the Upgrade* (on page 3-21)), using the options as needed. If an update in the PQI is needed, use the *pause* option and install the PQI before the database restore is done.

Upgrading from Non-cluster Setup to Cluster Setup

This section lists the basic procedure for upgrading from version 1.5 and up (from a non-cluster setup to a cluster setup).

Note This procedure attempts to minimize the SM's down-time as much as possible. Therefore, *if service is not an issue*, then the trivial procedure of installing a new machine and upgrading a new machine can be used instead.

In the following procedure let SM-A be the original SM machine running SM 1.5 and up, and SM-B be the new SM machine being added for redundancy.

To upgrade from a non-cluster setup to a cluster setup:

Step 1 Install the VCS on both machines.

Step 2 Install SM-B:

Install SM on the new machine by using the install-sm.sh script.

Install the application PQIs.

Step 3 Disconnect the LEGs:

For SM 1.5 and 2.0, stop the LEGs or remove the network connection with the LEGs.

> **Note:** From this step until after the finish of the procedure there is no SM to handle subscribers.

Step 4 Upgrade SM-A:

Run the `upgrade-sm.sh` script; if needed, use the pause option.

If needed, install the application PQI.

Step 5 Restore the LEG connection:

For SM 1.5 and 2.0, restart the LEGs or restore the network connection with the LEGs.

Step 6 Replicate the SM configuration from SM-A to SM-B (copy the `p3sm.cfg` configuration file manually from SM-A to SM-B; for loading, see *Reloading the SM Configuration (p3sm)* (on page 4-20)).

Step 7 Duplicate the subscriber database, as described in *Data Duplication Procedure* (on page 3-23).

Step 8 Create a cluster:

Configure SM-A and SM-B to support a cluster.

Make SM-B standby. Use the CLU command `p3cluster --standby`.

Make sure SM-A is active. Use the CLU command `p3cluster -–active`.

Configure the VCS.

Run the VCS on the setup.

Step 9 Configure the LEGs to send logins to the cluster virtual IP.

Upgrading from a Cluster Setup

In the following procedure, let SM-A be the active SM, and SM-B the standby SM.

To upgrade from a cluster setup:

Step 1 Upgrade SM-B:

Stop VCS monitoring of SM-B by using the VCS CLU `hastop –local`.

Step 2 Do one of the following:

If a database upgrade is needed (see the release notes):

a) Destroy SM-B’s database.

b) Upgrade SM-B by running the *`upgrade-sm.sh`* script with the database installation option.

c) If needed, install the application PQI.

d) Duplicate the database from SM-A, as described in *Data Duplication Procedure* (on page 3-23).

If no database upgrade is needed (see the release notes):

a) Run the upgrade-sm.sh script on SM-B with no database installation or restore.

b) If needed, install the application PQI.

Step 3 Restart VCS monitoring of SM-B by using the VCS CLU `hastart –local`.

Step 4 Make SM-B the active SM by using VCS to perform a manual fail-over (see Manual Fail-Over).

Step 5 Upgrade SM-A:

Stop VCS monitoring of SM-A by using the VCS CLU `hastop –local`.

Step 6 Do one of the following:

If a database upgrade is needed (see the release notes):

a) Destroy SM-A's database.

b) Upgrade SM-A by using the *`upgrade-sm.sh`* script with the database installation option.

c) If needed, install the application PQI.

d) Duplicate the database from SM-B, as described in *Data Duplication Procedure* (on page 3-23).

If no database upgrade is needed (see the release notes):

a) Run the *`upgrade-sm.sh`* script on SM-A with no database installation or restore.

b) If needed, install the application PQI.

Step 7 Restart VCS monitoring of SM-A by using the VCS CLU `hastart –local`.

Uninstalling

To uninstall the SM, perform the following procedure:

Step 1 Log in as the **root** user.

Step 2 If **in cluster setup**, stop the VCS monitoring of the SM, by running the following VCS CLU command from /opt/VRTSvcs/bin:

```
#./hastop –local
```

Step 3 Stop the SM, by running the following command:

```
>p3sm --stop
```

Step 4 If **in cluster setup**, stop the database replication, by running the following command:

```
>p3db –-rep-stop
```

Step 5 (Optional) Remove the datastore, either by using the TimesTen ttDestroy utility or by running the following command:

```
>p3db –-destroy-rep-db
>p3db –-destroy-local-db
```

Note: It is recommended to perform this step.

Step 6 Remove the SM installation directory, by running the following command:

```
# rm -r ~pcube/sm
```

Step 7 Remove the startup and shutdown scripts, by running the following command, as appropriate:

- For **Solaris**:

```
# rm /etc/rc*.d/[SK]*p3sm /etc/init.d/p3sm
```

- For **Red Hat**:

```
# rm /etc/rc.d/rc*.d/[SK]*p3sm /etc/rc.d/init.d/p3sm
```

Step 8 (Optional) Remove TimesTen, by running the following command:

```
# ~pcube/lib/tt/TimesTen/pcubesm22/bin/setup.sh -uninstall
```

Step 9 (Optional) Remove the Veritas Cluster agents by performing **one** of the following operations:

- Uninstalling the VCS.
- Removing configuration of SM's Veritas Cluster agents.

  To remove the VCS agent scripts, run the following commands:

```
# rm –rf /opt/VRTSvcs/bin/ProcessOnOnly
# rm –rf /opt/VRTSvcs/bin/PcubeSm
# rm –rf /opt/VRTSvcs/bin/TimesTenRep
```

Step 10 (Optional) Remove the **pcube** user, by running the following command:

```
# userdel -r pcube
```

Note If you chose to keep TimesTen installed, do **not** remove the **pcube** user.

CHAPTER 4

Configuration and Management

This chapter describes how to configure and manage the SM.

This chapter contains the following sections:

SM Management and Configuration Methods

The SM is configured and managed using:

- *Configuration File* (on page 4-1): Used for configuring the parameters that define the system's behavior.
- *Command Line Utilities* (on page 4-2) (CLU): Used mainly for viewing and for subscriber management.

The configuration file and CLU give you complete control over the SM, including subscriber management, database management, and SCE network configuration and management.

Configuration File

The SM has a configuration file, `p3sm.cfg`, located under `~pcube/sm/server/root/config/p3sm.cfg`. For a detailed description of the configuration file parameters, see *Configuration File Options* (on page A-1).

The configuration file (together with the *Command Line Utilities* (on page 4-2)) is used for configuring all the parameters that define the system's behavior. The configuration file contains the following kinds of parameters:

- General, system-wide parameters, such as subscriber state saving, persistency, subscriber introduction mode (pull-mode or push-mode), and topology
- Parameters for handling SM-LEG connection failure
- Parameters for handling SM-SCE connection failure
- Parameters for SCE Platform configuration
- Parameters for domain configuration

    - Associating domains and SCE platforms
    - Specifying domain aliases
    - Specifying domain properties
- Auto-logout parameters, for controlling automatic logout of subscribers after timeout
- Parameters for Radius Listener configuration
    - Specifying NAS configuration
    - Specifying properties configuration
- Parameters for FTP, HTTP, and RPC server configuration
- Parameters for Cable Adaptor configuration
- Parameters for configuring SM operation with TimesTen Database

Usually, the parameters in the configuration file are specified once when setting up the system, and are valid throughout the system lifetime. To modify the configuration file parameters, edit the file using any text editor and reload it using the CLU (see *Reloading the SM Configuration (p3sm)* (on page 4-20)). The configuration file is loaded on the boot of the SM, as well as by explicitly calling a CLU.

The configuration file is designed so that the same configuration file can be used in each of the SMs of a high-availability setup. This enables the user to replicate the configuration by simply copying the file from one machine to another.

Command Line Utilities

The SM provides a set of Command Line Utilities (CLU), which, together with the *Configuration File* (on page 4-1), are used for configuring the parameters that might change during the operation of the SM.

The CLU enables the user to configure the SM using shells installed on the machine. CLU commands are executable only when the user is logged in to the machine using the **pcube** account, which is always installed (see *Installation and Upgrading* (on page 3-1)). The CLU is used mainly for viewing and for subscriber management.

In high-availability setups, the CLU cannot be used to perform subscriber management operations on the standby SM. Moreover, the standby SM refreshes the database before performing subscriber display operations, so the operation takes longer (than for the active SM). It is therefore recommended to perform all subscriber operations on the active SM.

This chapter explains how to perform various tasks using the appropriate CLU, but it does not describe all the CLU utilities in detail. For a complete, detailed description of all the CLU utilities, see *Command Line Utilities* (on page B-1).

CHAPTER 5

Configuring a Subscriber Management Solution

This section explains the procedure for configuring a Cisco Service Control deployment consisting of several SCE Platforms and Subscriber Manager (SM) systems in order to make it ready for subscriber integration. See also *Configuration and Management* (on page 4-1).

This section uses the terminology and tools explained in previous chapters and, when needed, terms and configuration tools explained in the *SCE1000 and SCE 2000 User Guides*.

This chapter contains the following sections:

Prerequisites

Before configuring any components in your subscriber management solution, verify that all the items on the following checklist have been successfully completed.

- The SCE Platforms in your network are installed and configured as explained in the *SCE1000 and SCE2000 User Guides, Chapter 3*.
- The SM systems in your network are installed as explained in this user guide, in *Installation and Upgrading* (on page 3-1).
- The Cisco Service Control application (for example, *SCAS BB* or *SCAS M*) is installed on all SCE Platforms and SM systems in your network. Please refer to the relevant application installation guide for an explanation of how to install an application on the SCE Platforms and SM systems.
- The subscriber integration concept has been determined, and an appropriate solution was designed for driving subscriber mappings and policy information into the SM. This can be implemented automatically, using a LEG, or manually, using CLU.
- The subscriber introduction mode (push or pull) has been determined for each SM system, based on the number of subscribers that the relevant SCE Platforms should be serving.
- The association between SCE Platforms and the relevant SMs has been determined.
- For each SM system, the association between the SCE Platforms that it serves and subscriber domains has been designed.

Step-by-step Configuration Procedure

The procedure below applies to a single group, consisting of the following:

- An SM
- A set of LEGs that connects to this SM
- The SCE Platforms that this SM serves

Every subscriber management solution can be divided into such groups, and this procedure can be applied to each of these groups.

This procedure consists of the following steps:

- *Step 1 - Editing the SM Configuration File* (on page 4-2)
- *Step 2 - Importing Subscribers to the SM from a CSV File* (on page 4-11)
- *Step 3 - Configuring the SCE Platforms* (on page 4-12)

The last two steps (2 and 3) are not always required.

Step 1 - Editing the SM Configuration File

Edit the SM configuration file (*`p3sm.cfg`*) according to your system definition, and then reload it using the CLU `p3sm --load-config`.

For additional details regarding the SM configuration file, see *Configuration File Options* (on page A-1).

Following are descriptions of the sections of the SM configuration file and of their parameters.

Configuring SCE Platform Repository

Use **p3net** to add, connect and verify the connection state of each SCE Platform that should be provisioned by the SM:

Step 1 Use '**`p3net --add`**' to add the SCE Platform to the repository.

Step 2 Use '**`p3net –connect`**' to instruct the SM to connect to the SCE Platform.

Step 3 Use '**`p3net --show`**' to verify that the SCE Platform was successfully connected.

Step 4 When finished, use '**`p3net --show-all`**' to verify your configuration before you move on to configuring domains.

SE.XXX

After an SCE Platform has been physically installed (by being connected to the management network), it must be explicitly added to the SM list, or repository, of existing SCE Platforms before the SM will recognize it (see Configuring SCE Platform Repository). Conversely, after an SCE Platform is removed from that list, the SM will no longer recognize it, even though it is still physically connected.

Each SE.XXX section defines the following configuration parameters that represent a single SCE Platform:

- **ip**

  Defines the IP address of the SCE box.
- **port**

  Defines the port through which to connect to the SCE Platform. The default is 14374.

To view the SCE platforms, use the CLU command `p3net` ("Viewing/connecting Network Elements (p3net)" on page 4-26).

For additional information, see *SE.XXX Section* (on page A-12).

Configuring Domains

Use **p3domains** to add domains and then add SCE Platforms to these domains:

Step 1 Use '**`p3domains --add`**' to add domains to the domain repository.

Step 2 Use '**`p3domains --add-ne`**' to add SCE Platforms to each domain.

Step 3 When finished, use '**`p3domains --show-all`**' to verify your configuration before you start editing the configuration file.

Domain.XXX

When a system has more than one SCE Platform, they can be configured into groups, or domains (see Configuring Domains). A subscriber domain is one or more SCE Platforms that share a specified group of subscribers. Before adding an SCE Platform to a domain, you must add the SCE Platform to the SCE Platform repository.

Note The SM comes pre-configured with a single subscriber domain called **subscribers.**

Each Domain.XXX section specifies the elements (SCE Platforms), aliases, and properties for one domain. It contains the following parameters:

- elements=<logical_name1[,logical_name2,...]>

  Specifies the name(s) of the SCE platforms that are part of the domain.
- aliases=alias_name1[,alias_name2,...]

Defines domain aliases. When subscriber information is received from the LEG with one of the aliases (for example, `alias1`), the information is distributed to the domain that matches this alias (for example, `domain_name1`). A typical alias could be a network device IP address, where, for example, each string in the values can be the IP address of a NAS or a CMTS.

Note Each alias (for example, alias_name1) may appear in only **one** [Domain.XXX] section.

The specification `aliases=*` means that every subscriber that does not have a domain will be put in this domain. **Note:** Only **one** domain at any given time may specify this option (aliases=*).

- property.<name1>=<value1>[,property.<name2>=<value2>,...]

  Defines the default policy properties values for a domain. Unless these defaults are overridden by the LEG/API when it introduces the subscriber to the SM, the subscriber's policy is set according to the default policy properties values of its domain. Note that property values must be integers.

To view the domains, use the CLU command `p3domains` ("Viewing Subscriber Domains (p3domains)" on page 4-27).

For additional information, see *Domain.XXX Section* (on page A-6).

SM General

This section of the configuration file is relevant to any deployment topology. It addresses the following system-wide parameters:

- **introduction_mode**

  Defines whether the SM introduces the subscribers to the SCEs immediately after a login operation (Push mode), or only when the SCE requests the subscriber specifically (Pull mode).

- **application_subscriber_lock**

  Defines whether subscriber-related operations (login, logout, etc.) should be locked at the application level. Set this flag to `true` in cases subscribers can be updated simultaneously by several LEGs.

- **logon_logging_enabled**

  Defines whether the logging of subscriber logon events is enabled.

To view the SM settings, use the CLU command `p3sm` ("Managing the SM (p3sm)" on page 4-20).

Note Setting either `application_subscriber_lock` or `logon_logging_enabled` to `true` will cause performance degradation. For additional information, see *SM General Section* (on page A-2).

Data Repository

The Data Repository section defines the SM's operation with the TimesTen In-Memory Database, via the following parameters:

- **support_ip_ranges**

  Defines whether the SM supports IP-Ranges.

Note Disabling this support provides better performance.

- **checkpoint_interval_in_seconds**

  Defines the interval, in seconds, for calling the TimesTen checkpoints. Reducing the value affects performance, increasing the value increases vulnerability to power-down.

- **max_range_size**

  Determines the maximum IP range size used in the system. This parameter is used for improving performance of the SM in Pull mode when the `[Data Repository]` section is configured with `support_ip_ranges=yes`.

Note Defining this parameter with too large a value may cause performance degradation in handling pull requests.

For additional information, see *Data Repository Section* (on page A-14).

High Availability Setup

The High Availability section defines in what kind of topology the SM should work, via the parameter:

- topology

  Defines in what kind of topology the SM should work (cluster or standalone).

For additional information, see *High Availability Setup Section* ("SM High Availability Setup Section" on page A-3).

Subscriber State Persistency

The Subscriber State Persistency section defines the storage of subscriber information by the SM and SCE:

- **save_subscriber_state**

  Defines whether the SM stores the subscriber applicative state in case the subscriber logs out.

- **SCE_subscriber_persistency**

  Defines whether SCE stores subscriber information in persistent memory. Use this option **only** when there are just a few subscribers, which rarely change, in the system.

Note Setting SCE_subscriber_persistency to `true` may cause performance degradation and reduce the rate of subscriber updates. For additional information, see *Subscriber State Persistency Section* (on page A-3).

SM-LEG Failure Handling

The SM-LEG Failure handling section deals with the configuration of the behavior of the SM upon a LEG-SM communication link failure. Note that in order to activate the LEG-SM failure action you must also define LEG-Domain associations (see *LEG-Domains Association* (on page 4-6)).

- **clear_all_mappings**

  Defines the behavior of the SM in case of LEG-SM connection failure. This parameter is relevant only for cases where the SM and LEG are running on **different** machines.

- **timeout**

  Defines the time in seconds from a LEG-SM connection failure until clearing the mappings in the SM database.

For additional information, see *SM-LEG Failure Handling Section* (on page A-4).

LEG-Domains Association

The LEG-Domains association section relates to the SM-LEG Failure handling section. It associates LEGs and domains for the SM-LEG failure action. It contains pairs having the following form:

- <LEG name>=domain_name1[,domain_name2,...]

  Defines the domains whose subscriber mapping will be cleared on LEG-SM connection failure. The key is the <LEG name>. A <LEG name> of **"*"** specifies all LEGs. The (comma-separated) values are the domains (`domain_name`) to be cleared in the event of a network link failure (connection failure) between the specified LEG and the SM. A value of **"*"** for the `domain_name` specifies all subscriber domains in the system.

Several examples follow:

- All subscribers of domain **subscribers** will be cleared in case of connection failure between the SM and the specified LEG:

```
[LEG-Domains Association]
MYSERVER12.CNR.LEG=subscribers
```

- All subscribers of domains **domain1** and **domain2** will be cleared in case of connection failure between the SM and LEG *MYSERVER13.CNR.LEG*:

  In addition, all subscribers of domain **domain3** will be cleared in case of connection failure between the SM and LEG *MYSERVER14.CNR.LEG*:

```
[LEG-Domains Association]
MYSERVER13.CNR.LEG=domain1,domain2
MYSERVER14.CNR.LEG=domain3
```

- All domains of the specified LEG will be cleared in case of communication failure between this LEG and the SM:

```
[LEG-Domains Association]
MYSERVER15.CNR.LEG=*
```

- The specified domains will be cleared in case of communication failure between any LEG and the SM.

```
[LEG-Domains Association]
*=my_domain1,my_domain2
```

- All domains will be cleared in case of communication failure between any LEG and the SM:

```
[LEG-Domains Association]
*=*
```

For additional information, see *LEG-Domains Association Section* (on page A-5).

SM-SE Failure Handling

The SM-SE Failure handling section configures the behavior of the SCE Platform and the SM upon an SM-SE communication link failure, via the following parameter:

- **push_mode_auto_resync**

  Defines whether the SM automatically resynchronizes the SCE subscriber information when an SCE reconnects to the SM after a connection failure. This parameter is relevant only when working in Push mode. In Pull mode the SM always synchronizes the SCE on reconnection. (See *Push Mode* (on page 2-6) and *Pull Mode* (on page 2-6).)

For additional information, see *SM-SE Failure Handling Section* (on page A-6).

Default Domains Configuration

The Default Domains Configuration section defines the default policy properties values for all domains, via the parameter:

- property.<name1>=<value1>[,property.<name2>=<value2>,...]

  Defines the default policy properties values for a domain. Unless these defaults are overridden by the LEG/API when it introduces the subscriber to the SM, the subscriber's policy is set according to the default policy properties values of its domain. Note that property values are integers.

All domains are assigned all the properties defined in this section, unless they are overridden in the [*Domain.XXX* (on page 4-3)] section. This section is particularly useful when all domains in the system have same set of properties.

For example, using the following configuration,

```
[Default Domains Configuration]
property.A=1
property.B=2

[Domain.d1]

[Domain.d2]
property.A=2
property.C=1

[Domain.d3]
property.A=3
property.B=4
```

properties are assigned to the domains as follows:

Domain	Value of Property A	Value of Property B	Value of Property C
d1	1 (assigned by default)	2 (assigned by default)	
d2	2	2 (assigned by default)	1
d3	3	4	

To view the domains, use the CLU command `p3domains` ("Viewing Subscriber Domains (p3domains)" on page 4-27).

For additional information, see *Default Domains Configuration Section* (on page A-7).

Auto Logout

The Auto Logout section deals with the configuration of the SM to automatically logout a subscriber's IP after a certain period of time has passed since their last login operation. This time is called the lease-time; it is set by the LEG as part of the login operation. It contains the following parameters:

- *auto_logout_interval*

  Configures the timeout value, in seconds, between consecutive runs of the SM's auto-logout mechanism. When the mechanism identifies that the subscriber’s IP address lease-time has expired (and the *grace_period*, see below, has also passed), the subscriber’s IP address is logged out automatically (the IP mapping is removed).

- *grace_period*

  Defines the grace period, in seconds, for each subscriber IP logout. After a subscriber’s lease-time has expired, the subscriber is logged out automatically after the grace period has also expired.

- *max_rate*

Defines the maximum rate (logouts per second) that the auto-logout task is allowed to perform logouts from the system. This limit spreads out the load of the logout operations over time, reducing the performance impact on other operations. The value for this parameter should be calculated so that it spreads the logouts over at least half of the `auto_logout_interval` time.

These parameters are applicable in automatic integrations where the LEG cannot provide the SM with logout operations. For additional information, see *Auto Logout Section* (on page A-7).

Radius Listener

The Radius Listener section contains the general configuration parameters for the Radius Listener:

- **start**

  Defines whether the SM should run the Radius Listener at startup.

- **accounting_port**

  Defines the Radius Listener's accounting port number.

For additional information, see *Radius Listener Section* (on page A-9).

Radius.NAS.XXX

Each Radius.NAS.XXX section specifies the configuration parameters for a single Network Access System (NAS):

- **domain**

  Specifies the domain.

- **IP_address**

  Specifies the IP address in dotted notation (xxx.xxx.xxx.xxx).

- **NAS_identifier**

  Specifies the name of the NAS that exists in the **NAS-ID** attribute.

- **secret**

  Specifies a secret key defined in the NAS for this connection.

For additional information, see *Radius.NAS.XXX Section* (on page A-10).

Radius.Subscriber ID

The Radius.Subscriber ID section defines the RADIUS attribute on which subscriber ID association is based, via the following parameters:

- **radius_attribute**

  Specifies the Radius protocol attribute number. Use the value `26` in case of Vendor Specific Attributes (VSA).

- **radius_attribute_type**

  Specifies the Radius attribute type.

- **radius_attribute_vendor_id**

Used only if the association is based on Vendor Specific Attributes (VSA).

- **radius_sub_attribute**

  Used only if the association is based on Vendor Specific Attributes (VSA).

For additional information, see *Radius.Subscriber ID Section* (on page A-11).

Radius.Property.Package

The Radius.Property.Package section defines the RADIUS attribute from which a subscriber package is retrieved. It contains the following configuration parameters:

- **radius_attribute**

  Specifies the Radius protocol attribute number. Use the value `26` in case of Vendor Specific Attributes (VSA).

- **radius_attribute_type**

  Specifies the Radius attribute type.

- **packet_types**

  Specifies the Radius protocol packet types in which to look for the attribute.

- **use_default**

  Defines whether to use a default value if the attribute was not found (when not mandatory).

- **default**

  Defines the default value to use if the attribute was not found (when not mandatory).

- **radius_attribute_vendor_id**

  Used only if the association is based on Vendor Specific Attributes (VSA).

- **radius_sub_attribute**

  Used only if the association is based on Vendor Specific Attributes (VSA).

For additional information, see *Radius.Property.Package Section* (on page A-10).

RPC.XXX

Each RPC.XXX section defines the following configuration parameter that represents a single RPC server:

- **port**

  Defines the RPC server port. The default is 14374.

To view the RPC settings, use the CLU command `p3rpc` ("Viewing Information of RPC Interface Server (p3rpc)" on page 4-30).

For additional information, see *RPC.XXX Section* ("RPC.Server Section" on page A-12).

FTP

The FTP section contains the FTP server configuration parameters:

- **start**

Defines whether the SM should run the FTP server at startup.

- **port**

  Defines the FTP server port. The default is 21000.

For additional information, see *FTP Section* (on page A-13).

HTTP Tech-IF

The HTTP Tech-IF section contains the HTTP server configuration parameters:

- **start**

  Defines whether the SM should run the HTTP server at startup.

- **port**

  Defines the HTTP server port. The default is 8082.

For additional information, see *HTTP Tech-IF Section* (on page A-13).

Cable Adaptor

The Cable Adaptor section defines whether logins from unknown CMs are allowed, via the following parameter:

- **allow_dynamic_CM**

  Defines whether logins from unknown CMs are allowed.

To view the cable settings, use the CLU command `p3cable` ("Managing the Cable Support Module (p3cable)" on page 4-28).

For additional information, see *Cable Adaptor Section* (on page A-14).

Step 2 - Importing Subscribers to the SM from a CSV File

This step is optional and should be performed only when using manual integration, or when performing a setup prior to the beginning of the automatic integration.

A *`csv`* file is a simple text file where each line consists of comma-separated values. Since each line may contain subscriber properties, which are application dependant, the format of a *`csv`* import file is described in the documentation of the application that you installed in your system.

In most cases, when importing *`csv`* files you should use the CLU **`p3subsdb --import`**. When integrating with a cable AAA system and working in the *CPE as Subscriber* mode (see *CPE as Subscriber in Cable Environment* (on page C-1)), importing cable modems requires the CLU **`p3cable --import-cm`**.

Note The format of the *`csv`* file is different for the SCAS BB and SCAS M solutions. The specific format for each type of *`csv`* file can be found in the User Guide of each solution.

Step 3 - Configuring the SCE Platforms

This step is optional and should be performed only when using Pull mode to introduce subscribers or if a special operation should be performed on SM-SE connection failure. Several configuration parameters, discussed below, should be configured using the SCE Platform CLI.

Configuring these parameters ensures that the SCE Platform correctly applies appropriate defaults to subscribers in the period between subscriber detection and pull response (unmapped subscribers). For additional details, see the *SCE1000 or SCE2000 User Guide*, Chapter 8.

Anonymous Groups and Subscriber Templates

When the SCE Platform encounters network traffic that is not classified to any introduced subscriber, it checks whether the mapping of the unfamiliar subscriber belongs to one of the anonymous groups. If the subscriber belongs to an anonymous group, a new anonymous subscriber is spawned, and a request for an updated subscriber record is sent to the SM. The properties of the anonymous subscriber are taken from the subscriber template that is assigned to the newly spawned subscriber’s anonymous group.

Anonymous Subscriber Groups

An anonymous group is a specified IP range, possibly assigned a subscriber template (defined below). If a subscriber template has been assigned to the group, the anonymous subscribers generated have subscriber properties as defined by that template. If no subscriber template has been assigned, the default template is used.

Use the appropriate CLI commands to import anonymous group information from a *csv* file, or create/edit these groups explicitly.

Subscriber Templates

Values for various subscriber properties for unmapped or anonymous subscriber groups are assigned in the system based on subscriber templates. Subscriber templates are identified by a number from 0 to 199. Subscriber templates 1–199 are defined in *csv* formatted subscriber template files. However, template 0 cannot change; it always contains the default values. If an anonymous group is not explicitly assigned a template, the group uses template 0.

Use the appropriate CLI commands to import subscriber templates from a *csv* file, or edit these templates from the command line. Also, use the appropriate CLI commands to assign subscriber templates to the anonymous groups.

Subscriber Aging Parameters

To prevent SCE capacity problems in Pull mode, the aging of introduced subscribers should be configured. The aging parameter defines a timeout, and any subscriber that does not generate traffic during this timeout interval will be automatically logged out from the SCE.

SM-SCE Connection Failure

To prevent incorrect classification of a subscriber’s traffic during a lengthy connection failure between the SM and the SCE, the SM connection failure parameters should be configured. The SCE has several alternatives for operations: it can clear the mappings of all of the subscribers, it can put the line in cut-off mode, or it can do nothing. The timeout between the connection detection and actually performing the operation is also configurable.

System Configuration Examples

This section presents and explains common subscriber management scenarios, including the correct configuration parameters for these scenarios. The following scenarios are described:

- Automatic introduction of subscribers, with Push mode and fail-over of SCE platforms
- Manual introduction of subscribers with Pull mode
- SM fail-over scenario

Example 1: Automatic Introduction of Subscribers, with Push Mode and Fail-over of SCE Platforms

This scenario assumes the following:

- Automatic introduction of subscriber, that is, the subscribers are introduced by a provisioning system or an AAA system. This example assumes that integration with a DHCP server allows automatic introduction of subscribers to the SM.
- The SM is operating in push mode.
- The application that is used includes states that should be preserved (such as volume quotas states in SCAS BB).

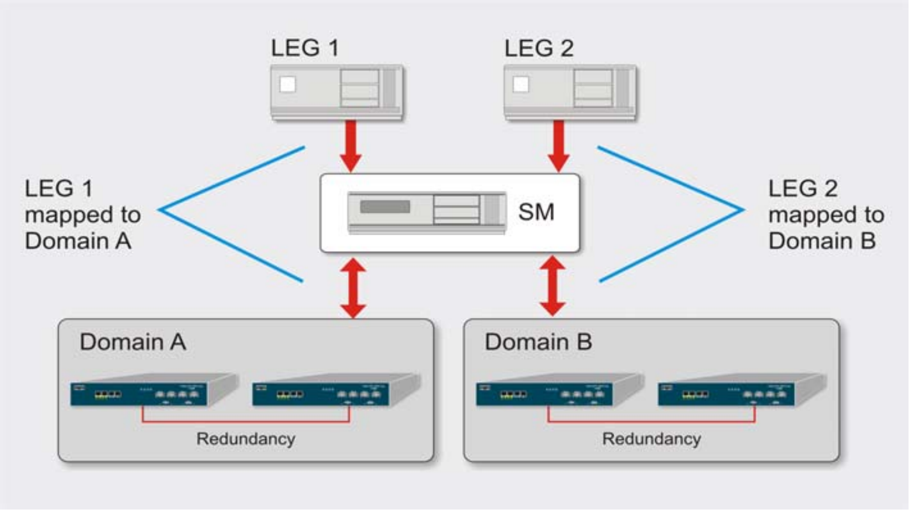


Figure 4-1: Cable Topology with Automatic Integration with a DHCP server, Push Mode, and Fail-over of SE Platforms

Note Make sure everything is properly installed before proceeding with the next steps.

Step 1 Edit the SM configuration file (`p3sm.cfg`) to add the SCE devices to the SCE device repository and then to group them to domains, as depicted in the above figure.

Step 2 Further edit the SM configuration file, as shown in the table below.

Step 3 Reload the SM configuration file using the **p3sm** CLU.

Step 4 Import the cable modems to the SM database using the **p3cable** CLU.

No SCE Platform configuration is needed in this scenario.

Table 4-1 Configuration File Parameters for Automatic Integration with Push Mode in a Cable Environment

Section and Parameter	Value and Description
SM General	
introduction_mode	push
High Availability Setup	
topology	standalone
	The value should be set to `standalone` since the described scenario has just one SM.
Subscriber State Persistency	
save_subscriber_state	true
	The value must be set to `true` if working with an application that requires preserving the state. In any case, however, it is advisable to set the value to `true`.
SCE_subscriber_persistency	false
	The value should be set to `false` due to limited space on the SCE platform that probably will not suffice in an automatic integration scenario. Moreover, setting this value to `true` causes performance degradation.
SM-LEG Failure Handling	
timeout	300 sec (more tolerance may be advisable for SM-LEG failures in actual configurations)
clear_all_mappings	true
	The value should be set to `true` because under the scenario conditions (automatic integration in cable environment), subscribers are likely to change their state or to logout from the SCE during an SM-LEG connection failure. We would therefore like to clean their mapping when the SM and LEG are connected again.
LEG-Domain Association	
<LEG name>	Define association(s) between LEG(s) and domain(s)
	LEG-Domain associations should be defined in order that all subscriber mappings will be cleared on SM-LEG disconnection. If no association is defined, subscriber mappings will not be cleared (the `clear_all_mappings` value will be ignored).

Section and Parameter	Value and Description
Default Domains Configuration	
property	None
	Configure the policy here if all domains in the system have the same policy.
Domain.XXX	
aliases	Define aliases, to facilitate working in cable environment
	Define aliases if working with LEGs that are not aware of system domains. Alternatively, you can define domains with names that match values produced by LEGs.
property	None
	Configure the policy here only if you want it to be applied to all subscribers in the domain.
Cable Adaptor	
allow_dynamic_CM	no
	The value should be set to `no` in order to prevent uninstalled CMs from using the network.
Auto Logout	
auto_logout_interval	60 min
	Auto-logout should be activated assuming that the AAA system cannot provide logout events (which is true in cable environment).
	The value defined here should be in the order but smaller than the CPE lease time in this cable environment.
grace_period	10 sec
	You should define a relatively high value, in order to eliminate mistakes due to management network delay.
max_rate	100 logouts per seconds
	You should define a value similar to the max-login rate to the SM.
SM-SE Failure Handling	
push_mode_auto_resync	true
	The value must be set to `true` when working in push mode, so that the SM will automatically synchronize with the SCE on reconnection. (Setting the value to `false` will result in loss of SM-SCE synchronization.)

Example 2: Manual Introduction of Subscribers in Pull Mode

This scenario assumes the following:

- Manual introduction of subscribers
- Pull mode
- Application that does not require preserving state

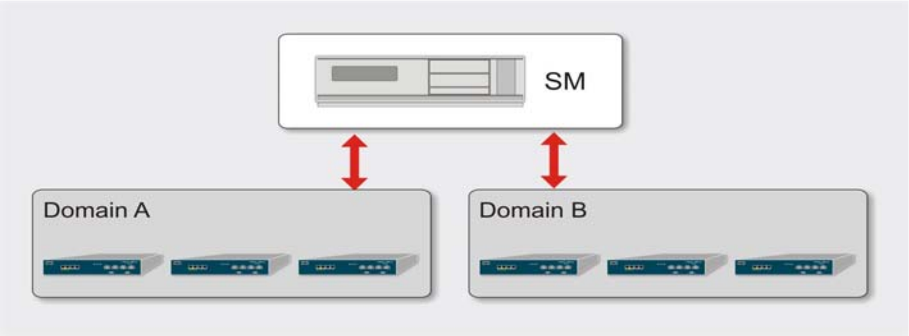


Figure 4-2: Topology with Manual Introduction of Subscribers and Pull Mode

Note Make sure everything is properly installed before proceeding with the next steps.

Step 1 Edit the SM configuration file (`p3sm.cfg`) to add the SCE devices to the SCE device repository and then to group them to domains, as depicted in the above figure.

Step 2 Further edit the SM configuration file, as shown in the table below.

Step 3 Reload the SM configuration file using the **p3sm** CLU.

Step 4 Import the subscribers using **p3subsdb** CLU (required for manual integration; there is no other way to bring subscribers into the SM).

Step 5 Use the SCE Platform CLI to configure the system for pull mode:

- Subscriber templates: in accordance with application
- Anonymous groups: in accordance with your network and subscribers
- Introduced subscriber aging: in accordance with your network and IP address allocation scheme.

Table 4-2 Configuration File Parameters for Manual Integration with Pull Mode

Section and Parameter	Value
SM General	
introduction_mode	pull
High Availability Setup	
topology	standalone The value should be set to `standalone` since the scenario involves just one SM.

Section and Parameter	Value
Subscriber State Persistency	
save_subscriber_state	false
SCE_subscriber_persistency	false
	Defines whether the SCE stores subscriber information in persistent memory. Use this option only when there are just a few subscribers, which rarely change, in the system.
SM-LEG Failure Handling	
timeout	Not applicable – no LEGs are involved in the scenario (use default – 60 seconds)
clear_all_mappings	Not applicable – no LEGs are involved in the scenario (use default – no action)
LEG-Domain Association	
<LEG name>	Not applicable – no LEGs are involved in the scenario (Use default – no mappings)
Default Domains Configuration	
Default policy (property.XXX)	None
	Configure the policy here only if all domains in the system have the same policy.
Domain.XXX	
aliases	Not applicable (use default – none)
property.XXX	None
	Configure the policy here only if you want it to be applied to all subscribers in the domain.
Cable Adaptor	
allow_dynamic_CM	Not applicable – valid for cable environment only (use default – no)
Auto Logout	
auto_logout_interval	Not applicable (use default – 0)
grace_period	Not applicable (use default – 10)
max_rate	Not applicable (use default – 50)
push_mode_auto_resync	Not applicable – used in push mode only (use default – true)

Example 3: SM Fail-over Configuration (General)

When using a set-up with SM fail-over (based on two SM nodes connected in a cluster), the configuration is identical to the regular configuration, with one exception:

- The *topology* parameter (in the *High Availability Setup Section* ("SM High Availability Setup Section" on page A-3)) should be set to *cluster* for both SM machines.

Other than that, SM fail-over configuration is performed normally via the `p3sm.cfg` configuration file, which should be copied manually from the active SM to the standby SM.

For additional information of how to configure the VCS, see *Veritas Cluster Server Requirements and Configuration* (on page E-1).

Using the CLU

This section starts with an introduction to the Command Line Utilities (CLU), and then describes how to use the CLU for viewing, subscriber management, and other tasks when working with the SM.

Note Some of the CLU operations and options can be specified by abbreviations. For this, as well as additional information regarding the CLU, see *Command Line Utilities* (on page B-1).

This section contains the following main subsections:

- Operations for Informative Output
- Parsing CLU Options
- Reloading the SM Configuration file
- Managing the SM
- Managing Subscribers, Mappings, and Properties
- Managing the Subscriber Database
- Viewing/connecting Network Elements
- Viewing Subscriber Domains
- Managing the Cable Support Module
- Installing an Application
- Viewing Statistics of the RPC Interface Server
- Managing the Cluster
- Managing the User Log
- Viewing Statistics of the RADIUS Listener
- Utilities (running batch files, printing help, database operations)

The procedures explained below invoke the following CLU commands:

- **p3batch**
- **p3cable**
- **p3clu**
- **p3cluster**
- **p3db**

- **p3domains**
- **p3inst**
- **p3log**
- **p3net**
- **p3radius**
- **p3rpc**
- **p3sm**
- **p3subs**
- **p3subsdb**

Informative Output

All CLU commands support the following operations for informative output:

Operation	Description
--help	Prints the help for the specified CLU command, then exits.
--version	Prints the SM program version number, then exits.

Parsing CLU Operations and Options

A command operation or option that contains any of the following characters must be placed in quotation marks:

- a space character
- a separation sign (comma “,”; ampersand “&”; colon “:”)
- an escape character (backslash “\”)

A command operation or option that contains any of the following characters must have that character preceded by an escape character:

- an equal sign (=)
- a quotation mark (“ or ”)
- an escape character (backslash “\”)

Following are several examples of the above rules:

Operation/option contains the character	Example of how operation/option should be written
Space character	--property=“file name”
Escape character (backslash “\”)	--property=“good\\bad”
Equal sign (=)	--property=“x\=y”
Quotation marks (“ or ”)	--name=“\“myQuotedName\””
	(in above example, inner quotation marks are escaped)
Separation characters	
• comma (,)	--names=“x,y”

• ampersand (&)	--names="x&y"
• colon (:)	--names="myHost:myDomain":8082

One-letter abbreviations are available for some of the operations and options. For example, "-d" is an abbreviation for "--domain". Note that the letter for an abbreviation is preceded by only one hyphen (-), not two, and that if the operation or option takes a parameter, there is a space and not an equal sign before the parameter.

• Example of using full name	--domain=subscribers
• Example of using abbreviated name	-d subscribers

Reloading the SM Configuration (p3sm)

Use the **p3sm** utility to configure the SM, by reloading the SM configuration file *p3sm.cfg*. The configuration file can be edited using any standard text editor.

To reload the SM configuration:

➲ From the shell prompt, type:

```
p3sm --load-config  [--ignore-warnings] [--
remote=<OTHER_SM_IP[:port]>
```

The configuration file is loaded, and the SM configuration updated accordingly.

The `–remote` option loads the configuration first to the local SM, and afterward to the remote SM (in High Availability setups).

Managing the SM (p3sm)

Use the **p3sm** utility to manage the SM on an ongoing basis. The **p3sm** utility enables you to start, stop, and resynchronize the SM.

To manage the SM:

➲ From the shell prompt, type a command having the following general format:

```
p3sm <operation> [--ne-name=<SCE NAME>]
```

The following table lists the **p3sm** operations relevant to managing the SM.

Table 4-3 p3sm Management Operations

Operation	Description
--show	Displays the current SM configuration.
--resync	Resynchronizes the subscribers of the specified SCE with the SM's database. The SCE is specified using the format `--ne-name=SCE_NAME`.
-- start [--wait]	Starts the server. The option `--wait` causes the CLU to return only after the SM is up.

Operation	Description
-- stop	Stops the server. Note: When using fail-over, a simple shut-down of the SM does not work (the Veritas Cluster Server identifies that the SM is down and attempts to restart it). The correct procedure is: 1. Perform the manual fail-over, as described in Manual Fail-Over. 2. Use the Veritas Cluster Manager Application to stop the monitoring (probing) of the SM. 3. Use the SM's CLU (p3sm --stop) to stop the SM.
-- restart [--wait]	Stops the server operation and then restarts it. The option `--wait` causes the CLU to return only after the SM is up.
--sm-version	Displays the version of the currently installed SM.
--sm-status [--detail]	Displays the SM operational status: whether the SM is running or not and whether it is Active or Standby. If errors have occurred, their summary is displayed. To receive a detailed description, use the option `--detail`.
--reset-sm-status	Clears the errors displayed, enabling the user to see if new errors have occurred.
--extract-support-file	Retrieves the support file from the agent. This command extracts the SM support information to a defined file. The SM support information should be extracted and sent to Cisco’s customer support with each support request.
--show-stats	Displays statistics information regarding logon operations. The rate results are updated once every 10 seconds.
--reset-stats	Resets the statistics information.
--logging=[on/off]	Enables/disables the logging of user logon to the UserLog. **Note:** When this flag is enabled it may affect performance.

EXAMPLE 1:

The following example shows how to stop the server operation and then restart it.

```
p3sm -–restart
```

EXAMPLE 2:

The following example shows how to resynchronize an SCE whose logical name is *SCE_1000A*.

```
p3sm -–resync --ne-name=SCE_1000A
```

EXAMPLE 3:

The following example shows how to extract the SM support information to a file named *support.zip*.

```
p3sm –-extract-support-file --output=support.zip
```

Managing Subscribers, Mappings, and Properties (p3subs)

Use the **p3subs** utility to manage subscribers. You can add or remove subscribers. You can also manage subscriber properties and mappings with this utility.

To manage subscribers:

➲ From the shell prompt, type a command having the following general format:

```
p3subs <operation> --subscriber=<Subscriber-Name> [--ip=<IP-address>]
[--property=<property-name=value>] [--domain=<domain-name>] [--overwrite]
```

The subscriber on whom the operation is to be performed is specified using the format `--subscriber=Subscriber-Name`. An IP address (mapping), property, or domain, if specified, uses the format shown.

Note If a domain is not specified, the subscriber is added to the default **subscribers** domain.

This section describes the following:

- *Managing Subscribers* (on page 4-22)
- *Managing Mappings* (on page 4-23)
- *Managing Properties* (on page 4-23)
- *Clearing Subscriber's Applicative State* (on page 4-24)

Managing Subscribers

The following table lists the **p3subs** operations relevant to managing subscribers.

Table 4-4 p3subs Subscriber Operations

Operation	Description
--add	Adds a subscriber or replaces the existing subscriber configuration.
--set	Updates mappings and/or properties for the specified subscriber. If the subscriber does not exist, this operation acts like the **add** operation.
--remove	Removes the specified subscriber.
--show	Displays information for specified subscriber.

EXAMPLE 1:

The following example shows how to add a subscriber with the specified IP address.

```
p3subs --add --subscriber=jerry --ip=96.142.12.7
```

EXAMPLE 2:

The following example shows how to overwrite subscriber information. Since the subscriber **jerry** already exists, the add operation would fail; however, the **overwrite** option allows the IP address to be overwritten.

```
p3subs --add --subscriber=jerry --ip=96.128.128.42 --overwrite
```

Managing Mappings

The following table presents all the **p3subs** operations relevant to managing mappings.

Table 4-5 p3subs Mapping Operations

Operation	Description
--set	Add/update a subscriber. If the mappings exist, replaces the existing mappings, unless the **additive-mapping** option is used.
--remove-all-mappings	Removes all the mappings of specified subscriber.
--remove-mappings	Removes specified mapping of specified subscriber.
--show-all-mappings	Displays all the mappings for specified subscriber.

Table 4-6 p3subs Mapping Options

Option	Description
--additive-mapping	Adds the specified mappings to the existing ones (instead of replacing the existing mappings when this option is not used). Used with the **set** operation.

EXAMPLE:

The following example shows how to removes all the mappings for a specified subscriber.

```
p3subs --remove-all-mappings --subscriber=jerry
```

Managing Properties

The following table lists the **p3subs** operations relevant to managing subscriber properties. The property names depend on the application running on your system. To find descriptions of the application property names and values, see the documentation provided with the application installed on your system.

Table 4-7 p3subs Subscriber Property Operations

Operation	Description
--set	When a property is specified, the subscriber record is updated with the specified property and value.
--reset-property	Resets specified property of specified subscriber to its default value.
--show-property	Displays value of specified property of specified subscriber.

Operation	Description
--show-all-properties	Displays all the properties of specified subscriber.
--show-all-property-names	Displays all the property names and descriptions.
--remove-properties	Removes all properties and custom properties from the specified subscriber record.

EXAMPLE:

The following example shows how to set a property value for a specified subscriber.

```
p3subs --set --property=rdr.transaction.generate=true --subscriber=jerry
```

Clearing Subscriber’s Applicative State

The following table presents the **p3subs** operation that clears a subscriber’s applicative state.

Table 4-8 p3subs Mapping Operations

Operation	Description
--clear-state	Clears the subscriber’s applicative state. This command clears only the backup copy at the SM. It does **not** clear the applicative state record in the SCE Platform.

EXAMPLE:

The following example shows how to clear the applicative state of the specified subscriber. Note that this command clears only the backup copy at the SM. It does **not** clear the applicative state record in the SCE Platform.

```
p3subs –-clear-state --subscriber=jerry
```

Managing the Subscriber Database (p3subsdb)

Use the **p3subsdb** utility to manage the SM database. You can import subscriber information for a group of subscribers from a CSV file into the SM database. You can also export subscriber information from the SM database to a CSV file.

Note The format of the CSV file depends on the application. A specific application’s CSV file format is specified in the documentation for that application.

To manage the SM database:

➲ From the shell prompt, type a command having the following general format:

```
p3subsdb <operation> [--domain=<domain-name>] [<filename>]
```

The following table lists the **p3subsdb** operations.

Table 4-9 p3subsdb Operations

Operation	Description
--clear-all	Removes all subscriber records from the database.
--clear-domain	Removes all subscriber records from the specified domain.
--show-num	Displays the number of subscribers in the database for the specified domain.
--show-all	Lists all of subscriber names.
--show-domain	Lists all subscriber names in the specified domain.
--import	Imports subscribers to the database from a specified CSV file. The filename that is to be imported is specified using the format `--file=import-filename`. The results go to a results files, *import-results.txt*, which is created in the same directory as the CSV file.
--export	Exports subscribers from the database to a specified CSV file. The filename that is to be exported is specified using the format `--output=export-filename`. The results go to a results file, *export-results.txt*, which is created in the same directory as the CSV file.
--clear-all-states	Clears the state of all subscribers in the SM database.

Table 4-10 p3subsdb Options

Operation	Description
--domain=DOMAIN	Performs the operation on the specified DOMAIN. If DOMAIN is **none**, then the operation refers to the subscribers who have no domain specified.
--prefix=NAME	Used in the export command for filtered export.

Table 4-11 p3subsdb File Options

Operation	Description
--file=FILE	One or more subscriber CSV FILEs to import from.
--output=FILE	Subscriber CSV FILE to export to.

EXAMPLE 1:

The following example shows how to list all subscribers in a specified domain.

```
p3subsdb --show-domain --domain=mainDomain
```

EXAMPLE 2:

The following example shows how to import subscribers from the specified CSV file.

```
p3subsdb --import --file=goldSubscriberFile.csv
```

EXAMPLE 3:

The following example shows how to export subscribers with filtering options to a specified CSV file.

```
p3subsdb --export --prefix=a -–output=silverSubscriberFile.csv
```

Viewing/connecting Network Elements (p3net)

Use the **p3net** utility for viewing connection status of network elements and trying to reconnect disconnected elements.

To view connection status of a network element or try to reconnect a disconnected element:

➲ From the shell prompt, type a command having the following general format:

```
p3net <operation> --ne-name=<logical-name>
```

The following tables list **p3net** operations and options.

Table 4-12 p3net Operations

Operation	**Description**
--show-all	Shows all the configured network elements.
--show	Shows an element's connection status.
--connect	Tries to connect a disconnected element

Table 4-13 p3net Options

Network Element Option	**Description**
--ne-name=NAME	Specifies the logical NAME for the network element.
--detail	(Optional) Used with the **--show-all** operation for displaying additional information as a table.

EXAMPLE:

The following example shows how to display a network element's connection status.

```
p3net --show -–ne_name=mainNE
```

Viewing Subscriber Domains (p3domains)

Use the **p3domains** utility for viewing the subscriber domains. (As explained in *Introducing the Subscriber Manager*, subscriber domains are groups of SCE devices that serve the same subscribers.)

To view subscriber domains:

➲ From the shell prompt, type a command having the following general format:

```
p3domain <operation> --domain=<domain-name>
```

The following tables list **p3domain** operations and options.

Table 4-14 p3domain Operations

Operation	Description
--show-all	Shows all configured domains.
--show	Shows a domain and its associated network elements.

Table 4-15 p3domain Options

Domain/NE Option	Description	Notes
--domain=DOMAIN	DOMAIN specifies logical name	"**none**" cannot be used, it is a reserved word

Managing the Cable Support Module (p3cable)

In the cable environment, the SM supports two modes of operation: *CM as Subscriber* and *CPE as Subscriber*, as described in *CPE as Subscriber in Cable Environment* (on page C-1).

This section discusses only the support of the *CPE as Subscriber* mode. In this mode the CPE is modeled as the subscriber, and it inherits its policy and domain from the cable modem (CM) through which it connects to the network. Each cable modem is linked with one or more CPEs. (For background information about special characteristics of the *CPE as Subscriber* mode in the cable environment, see *CPE as Subscriber in Cable Environment* (on page C-1).)

Use the **p3cable** utility commands to import cable modem information from a CSV file to the SM, and to export the cable modem information from the SM to a CSV file. You can also use this utility to clear the repository of all cable modems, and to show whether the login of CPEs that belong to unfamiliar cable modems (cable modems that do not exist in SM database) is allowed or denied. However, for specifying whether such login is allowed or denied, you should use the *Cable Adaptor* (on page 4-11) section of the configuration file `p3sm.cfg`.

To manage the cable modems:

➲ From the shell prompt, type a command having the following general format:

```
p3cable <operation> --cm=<CM-name> [<filename>] [<other CM
options>]
```

The following table lists the **p3cable** operations.

Table 4-16 p3cable Operations

Operation	Description
--import-cm	Imports cable modems from a CSV file. The filename that is to be imported is specified using the format `--file=import-filename`. The results go to a results file.
--export-cm	Exports cable modems to a CSV file. The filename that is to be exported is specified using the format `--output=export-filename`. The results go to a results file.
--clear-all-cm	Clears the repository of all cable modems.
--show-dynamic-mode	Shows whether the login of CPEs that belong to unfamiliar cable modems (cable modems that do not exist in SM database) is allowed or denied.

EXAMPLE 1:

The following example shows how to import cable modems from a CSV file.

```
p3cable --import-cm --file=CMFile.csv
```

EXAMPLE 2:

The following example shows how to clear the repository of all cable modems:

```
p3cable --clear-all-cm
```

EXAMPLE 3:

The following example shows how to display login status (allow/deny) of CPEs that belong to cable modems that do not exist in the SM database:

```
p3cable --show-dynamic-mode
```

Installing an Application (p3inst)

Use the **p3inst** utility to install or uninstall an application (*pqi* file). An application can be installed on the SM in order to customize the components. You can also upgrade an existing application to a new version, or uninstall a previously installed application.

To install/uninstall an application:

➲ From the shell prompt, type a command having the following general format:

```
p3inst <operation> [file options]  [<installation/upgrade
parameters>]
```

The following table lists the **p3inst** operations.

Table 4-17 p3inst Operations

Operation	Description
--install	Installs the specified application *pqi* file to the SM. It may be necessary to specify arguments for the installation procedure in the command line.
--uninstall	Uninstalls the specified application *pqi* file from the SM.
--upgrade	Upgrades an existing application using the specified application *pqi* file. It may be necessary to specify arguments for the upgrade procedure in the command line.
--rollback	Returns the specified application to the previous version. Rollback is the reverse of an **upgrade** operation: it undoes the **upgrade**.
--describe	Displays the contents of the specified application *pqi* file.
--show-last	Lists the details of the last installed application *pqi* file

EXAMPLE 1:

The following example shows how to install the specified installation file.

```
p3inst --install --file=myInstallation.pqi
```

EXAMPLE 2:

The following example shows how to uninstall the specified installation file.

```
p3inst --uninstall –-file=oldInstallation.pqi
```

Viewing Information of RPC Interface Server (p3rpc)

Cisco provides a proprietary RPC (Remote Procedure Call) interface to the SM. Use the **p3rpc** utility to view the configuration and statistics of the RPC server.

To display the RPC interface server information:

➲ From the shell prompt, type a command having the following general format:

```
p3rpc <operation>
```

The following table lists the **p3rpc** operations.

Table 4-18 p3rpc Operations

Operation	**Description**
--show	Displays the port number the RPC server listens to, the maximum number of connections, the current number of active connections, and the host IP.
--show-statistics	Displays the RPC server statistics. They contain information about the number of current RPC sessions, as well as statistics for RPC server actions such as invocations and errors.
--reset-statistics	Clears the RPC server statistics.
--show-client-names	Displays the names of the connected clients.

EXAMPLE:

The following example displays the statistics of the RPC server.

```
p3rpc ––show-statistics
```

Managing a Cluster of Two SM Nodes(p3cluster)

Use the **p3cluster** utility to view the redundancy state of the SM and its components. This utility also supports operations that alter the redundancy state of the SM. These operations are used by the SM Cluster Agent, as well as for administrative tasks.

To manage the cluster:

➲ From the shell prompt, type a command having the following general format:

```
p3cluster <operation>
```

The following table lists the **p3cluster** operations.

Table 4-19 p3cluster Operations

Operation	**Description**
--show	Displays the redundancy status of the SM and its components.
--active	Make the SM become the active SM in the cluster.
--standby	Make the SM become the standby SM in the cluster.

EXAMPLE:

The following example displays the redundancy status of the SM and it components.

```
p3cluster ––show
```

Managing the User Log (p3log)

Use the **p3log** utility to configure and manage the user log. All user-related events and errors are directed to the SM user log. You can extract the contents of the user log to a specified file in order to read/save its contents. You can also clear the user log.

To extract the user log to a specified file:

➲ From the shell prompt, type:

```
p3log –-extract --output=my_file
```

To clear the user log:

➲ From the prompt, type:

```
p3log –-reset
```

Viewing Statistics of RADIUS Listener (p3radius)

Use the **p3radius** utility to view the statistics of the RADIUS Listener LEG. For information about this CLU, see the SCMS SM *RADIUS Listener LEG Guidelines* document.

Utilities

This section describes the following:

- Running a batch file (p3batch)
- Printing general help about CLU commands (p3clu)
- Database operations (p3db)

Running a Batch File (p3batch)

Use the **p3batch** utility to run a batch file and execute its commands. (Using a text editor, you can create a batch file that contains a series of CLU commands, one command per line.) This operation (**p3batch**) enables you to run multiple operations on a single connection to the SM.

To run a batch file:

➲ From the shell prompt, type a command having the following general format:

```
p3batch [file-options] [error-options]
```

Table 4-20 p3batch File Option

Operation	Description
--file=FILE	Runs a batch file, where FILE specifies the CLU script (batch) file to run.

Table 4-21 p3batch Error Option

Error Option	Description
--skip-errors	Specifies that the batch operation should *not* halt if an error occurs. If this flag is not used, the batch operation *will* halt if an error occurs.

EXAMPLE 1:

The following example shows how to run a batch file that will halt if an error occurs.

```
p3batch --file=mainBatchFile.txt
```

EXAMPLE 2:

The following example shows how to run a batch file that will not halt if an error occurs.

```
p3batch --file=mainBatchFile.txt --skip-errors
```

Printing General Help About CLU Commands (p3clu)

Use the **p3clu** utility to print a list of all supported CLU utilities and operations.

To print all CLU commands:

➲ From the shell prompt, type:

```
p3clu --help
```

Database Operations

Use the **p3db** utility to manage and monitor the TimesTen database. The CLU exposes capabilities of the some of TimesTen CLUs with respect to specific needs of the SM.

To manage/monitor the TimesTen database:

➲ From the shell prompt, type a command having the following general format:

```
p3db <operation> [ options]
```

The following tables list the **p3db** operations and options.

Note Take care when activating commands that can affect the database; if used incorrectly, these commands may possibly **damage** the database.

Table 4-22 p3db Operations

Operation	Description
--rep-status	Displays status of the replication agent.
--rep-start	Starts the replication agent. **Note:** Use only for database recovery.
--rep-stop	Stops the replication agent. **Note:** Use only for database recovery.
--status	Displays the database status.
--destroy-rep-db	Destroys the replicated data-store.
--destroy-local-db	Destroys the local data-store.

Operation	Description
--keep-in-mem [SECS]	Indicates to the database daemon how many seconds to keep the database in the memory, after the last connection to the database is down. This option should be used in large databases to reduce the SM restart time. **Note:** To prevent limitations in performing database destroy, do not use values above a few minutes (that is, above a few hundred seconds).
--duplicate	Copies the data-store from the *remote* machine (specified by "--remote=REMOTE_HOSTNAME") to the *local* machine (specified by "--local=LOCAL_HOSTNAME"). **Note:** This option is applicable only for a **cluster setup**. For additional information, see *Data Duplication Procedure* (on page 3-23).

Table 4-23 p3db Options

Option	Description
--local=LOCAL_HOSTNAME	Specifies the local machine.
--remote=REMOTE_HOSTNAME	Specifies the remote machine.

EXAMPLE

The following example shows how to request the status of the replication agent, and also lists a typical response:

```
bash-2.03$ p3db --rep-status
Peer name         Host name                 Port    State  Proto
----------------  ------------------------ ------  ------- -----
PCUBE_SM_REPOSITORY  SHAY-SUNFIRE_REP          Auto   Start      11

Last Msg Sent Last Msg Recv Latency TPS     RecordsPS Logs
------------- ------------- ------- ------- --------- ----
00:00:02      00:00:00         1.15    2723      5447    1

the subscriber DB is ok
Command terminated successfully
```

Configuration File Options

This appendix describes in detail all the parameters that can be configured by using the Subscriber Manager (SM) configuration file. (The shorter description of SM configuration *given in Configuration and Ma*nagement (on page 4-1) is more oriented toward the routine configuration tasks that can be performed online using the CLU.)

Introduction

The SM can be configured only by using its configuration file (the CLU is used for displaying, not configuring). The SM has one configuration file, *p3sm.cfg*, which is located under *~pcube/sm/server/root/config/p3sm.cfg*. To change any configuration parameter, edit the configuration file using a standard text editor, and then use the CLU to reload it (see *Reloading the SM Configuration (p3sm)* (on page 4-20)).

The *p3sm.cfg* configuration file can be used for setting parameters for the following:

- SM configuration
- Radius Listener configuration
- Redundancy (cluster/standalone) configuration
- Domain configuration
- SCE configuration
- Cable adaptor configuration
- RPC port configuration
- FTP port configuration
- HTTP port configuration
- Database configuration

Description of the Configuration File Options

The following sections correspond the to the sections of the SM configuration file, *p3sm.cfg*.

- SM General
- SM High Availability Setup

- Subscriber State Persistency
- SM-LEG Failure Handling
- LEG-Domains Association
- SM-SE Failure Handling
- Domain.XXX
- Default Domains Configuration
- Auto Logout
- Radius Listener
- Radius.NAS.XXX
- Radius.Subscriber ID
- Radius.Property.Package
- SE.XXX
- RPC.Server
- FTP
- HTTP Tech-If
- Cable Adaptor
- Data Repository

For an explanation of the terms and concepts used in these sections, see *Configuring a Subscriber Management Solution* (on page 4-1).

SM General Section

The `[SM General]` section contains the following parameters:

introduction_mode

Defines whether the SM introduces the subscribers to the SCEs immediately after a login operation (Push mode), or only when the SE requests the subscriber specifically (Pull mode).

Possible values for this parameter are `push` and `pull`. The default value is `push`.

Following is an example of assigning a value to this parameter:

```
introduction_mode=push
```

application_subscriber_lock

Defines whether subscriber-related operations (login, logout, etc.) should be locked at the application level. Set this flag to `true` only in case the same parameters of a subscriber can be updated simultaneously by several LEGs (but if, for example, one LEG updates IP addresses and another LEG updates packages, there is no reason to set this flag to `true`).

Note Setting this flag to `true` will cause performance degradation.

Possible values for this parameter are true and false. The default value is false.

Following is an example of assigning a value to this parameter:

```
application_subscriber_lock=false
```

logon_logging_enabled

Defines whether the logging of subscriber logon events is enabled.

Note Setting this flag to true will cause performance degradation.

Possible values for this parameter are true and false. The default value is false.

Following is an example of assigning a value to this parameter:

```
logon_logging_enabled=false
```

SM High Availability Setup Section

The [SM High Availability Setup] section contains the following parameter:

topology

Defines in what kind of topology the SM should work.

Possible values for this parameter are standalone and cluster. The default value is standalone.

Following is an example of assigning a value to this parameter:

```
topology=standalone
```

Subscriber State Persistency Section

The [Subscriber State Persistency] section contains the following parameters:

save_subscriber_state

Defines whether the SM stores the subscriber applicative state in case the subscriber logs out.

Possible values for this parameter are true and false. The default value is true.

Following is an example of assigning a value to this parameter:

```
save_subscriber_state=true
```

SCE_subscriber_persistency

Defines whether SCE stores subscriber information in persistent memory. Use this option **only** when there are just a few subscribers, which rarely change, in the system.

Note Setting this parameter to true causes performance degradation, reducing the rate of subscriber updates.

Possible values for this parameter are true and false. The default value is false.

Following is an example of assigning a value to this parameter:

```
SCE_subscriber_persistency=false
```

SM-LEG Failure Handling Section

The `[SM-LEG Failure Handling]` section contains parameters that affect the discovery of an SM-LEG connection failure event and the actions taken by SM upon a connection failure event. An SM-LEG connection failure event may be caused either by a network problem or by a severe failure (such as reboot) of the SM or the LEG.

The [SM-LEG Failure Handling] section contains the following parameters:

clear_all_mappings

Defines the behavior of the SM in case of LEG-SM connection failure. This parameter is relevant only for cases where the SM and LEG are running on **different** machines.

Note This parameter defines a behavior that is similar for **all** LEGs connected to the SM.

If this parameter is set to `true`, then in case of LEG-SM connection failure that is not recovered within the defined `timeout`, the mappings of all subscribers in the domains defined in the *[LEG-Domains Association]* ("LEG-Domains Association Section" on page A-5) section for the LEG that was disconnected, will be removed.

Note **Important:** If you set the **clear_all_mappings** parameter to `true`, you must **also** set the **LEG-Domains Association** parameter in order for the SM to actually clear the mappings in case of LEG-SM connection failure.

Possible values for this parameter are `true` and `false`. The default value is `false`.

Following is an example of assigning a value to this parameter:

```
clear_all_mappings=false
```

timeout

Defines the time in seconds from a LEG-SM connection failure until clearing the mappings in the SM database.

Note It takes several seconds for the SM to detect an SM–LEG connection failure. You must add this time to the value of the **Timeout** parameter when calculating how long it will take the SM to react in case of an SM–LEG connection failure. For example, if **Timeout** is set to 80, it will take 80 seconds **from the time the failure is detected** until the SM clears the subscribers database.

The default value for this parameter is 60 (seconds).

Following is an example of assigning a value to this parameter:

```
timeout=60
```

LEG-Domains Association Section

The `[LEG Domains Association]` section defines the domains for which the mapping of all subscribers that belong to them will be cleared on LEG-SM connection failure. This section is relevant only if the **clear_all_mappings** parameter has been set in the *[SM-LEG Failure Handling]* ("SM-LEG Failure Handling Section" on page A-4) section.

Note Even though you set the **LEG-Domains Association** parameter, you must **also** set the **clear_all_mappings** parameter (to `true`) in order for the SM to actually clear the mappings in case of SM-LEG connection failure.

This section contains a list of LEG-Domain associations, each item in a separate line. Each LEG-Domain association is specified as shown for the following parameter:

<LEG name>=domain_name1[,domain_name2,...]

Defines the domains whose subscriber mapping will be cleared on LEG-SM connection failure. The key is the `<LEG name>`.

To determine which value(s) should be used for the `<LEG name>` key, consult the documentation of the LEG that you use. The `<LEG name>` usually is divided into two parts: <hostname>.`<common suffix>`. The first part is a general LEG identifier. The second part is extracted from the machine on which the LEG is running. Alternatively, you can use the CLU command `p3rpc --show-client-names`.

A `<LEG name>` of **"*"** specifies all LEGs. The (comma-separated) values are the domains (`domain_name`) to be cleared in the event of a network link failure (connection failure) between the specified LEG and the SM. A value of **"*"** for the `domain_name` specifies all subscriber domains in the system.

Note The LEG name is **case sensitive**.

By default, there are no LEG domain mappings.

Following is an example of assigning values to this parameter:

```
10.1.12.76.NB.SM-API.J=home_users
10.1.12.77.B.SM-API.C=office_users
```

Following is an example of specifying all subscriber domains:

```
10.1.12.31.CNR.LEG=*
```

Following is an example of specifying all LEGs and all subscriber domains:

```
*=*
```

SM-SE Failure Handling Section

The [SM-SE Failure Handling] section contains the following parameter:

push_mode_auto_resync

Defines whether the SM automatically resynchronizes the SCE subscriber information when an SCE reconnects to the SM after a connection failure. This parameter is relevant only when working in Push mode. In Pull mode the SM always synchronizes the SCE when it reconnects to SM. (See *Push Mode* (on page 2-6) and *Pull Mode* (on page 2-6)).

Possible values for this parameter are `true` and `false`. The default value is `true`.

Following is an example of assigning a value to this parameter:

```
push_mode_auto_resync=true
```

Domain.XXX Section

Each `[Domain.XXX]` section specifies one domain, where XXX represents the domain name.

This section contains the following parameters:

elements=<logical_name1[,logical_name2,...]>

Specifies the name(s) of the SE platforms that are part of the domain.

Note Each name must be the **exact** "XXX" name (**case sensitive**) that appears in the *SE.XXX* ("SE.XXX Section" on page A-12) section.

Following is an example of assigning a value to this parameter:

```
elements=se0,se1
```

aliases=alias_name1[,alias_name2,...]

Defines domain aliases. When subscriber information is received from the LEG with one of the aliases (for example, `alias1`), the information is distributed to the domain that matches this alias (for example, `domain_name1`). A typical alias could be a network access device's IP address, where, for example, each string in the values can be the IP address of a NAS or a CMTS.

Note Each alias can appear in only one domain section.

By default, there are no domain aliases.

Following is an example of assigning a value to this parameter:

```
aliases=10.10.88.99,10.10.88.98
```

property.<name1>=<value1>[,property.<name2>=<value2>,...]

Defines the default policy properties values for a domain. Unless these defaults are overridden by the LEG/API when it logs in the subscriber to the SM, the subscriber's policy is set according to the default policy properties values of its domain.

The policy format is a comma-separated list of `property_name=property_value`, where each property value is an integer.

Note To learn more about policy specification, read the user guide of the solution you are using: SCAS BB or SCAS M.

Following is an example of assigning a value to this parameter:

```
property.packageId=1
```

Default Domains Configuration Section

The `[Default Domains Configuration]` section defines default policy for the domain. It is used for those domain properties that were not defined in the domain's policy (see `[Domain.XXX]` ("Domain.XXX Section" on page A-6)), as well as for properties of domains for which no policy whatever was defined.

This section contains the following parameter:

property.<name1>=<value1>[,property.<name2>=<value2>,...]

Defines the default policy properties values for all domains. Unless these defaults are overridden by the LEG/API when it logs in the subscriber to the SM, or unless they are overwritten by the default policy property values of the subscriber's domain, the subscriber's policy is set according to the global default policy property values defined in this section.

The policy format is a comma-separated list of `property_name=property_value`.

Note To learn more about policy specification, read the user guide of the SCAS solution you are using.

Following is an example of assigning a value to this parameter:

```
property.packageId=1
```

Auto Logout Section

The `[Auto Logout]` section defines the parameters for the **Auto Logout** feature, which is relevant mainly for cable environments. This feature is relevant to automatic integrations where the LEG/API cannot provide logout indications. In such a case, you can turn on the automatic logout mechanism, which instructs the SM to log out a subscriber automatically after a certain period of time. Note that a login event for a subscriber resets that subscriber’s logout timer.

Note *Not* using the **Auto Logout** feature in the scenario described above (a provisioning system that can provide subscriber login events to the SM but cannot provide subscriber logout events) might result in exhausting the SCE resources, because subscribers are logged in but are never logged out.

This section contains the following parameters:

auto_logout_interval

Configures the interval value, in seconds, of the SM's auto-logout mechanism. Every interval, the SM checks for which subscriber IP-addresses the lease time has expired, and begins to automatically remove these IP addresses from the system.

Lease time is the timeout defined by the LEG during the login operation per each IP address. All subscriber login events will start a timer of lease-time seconds. When the timer expires (and the grace period, see below, has also passed), the subscriber's IP-addresses are removed (causing the subscriber to be removed from the SCE Platform's database). Note that any login event by the subscriber with an existing IP-address during the timer countdown period resets the timer, causing it to restart.

If the interval value is set to zero (0), the SM's auto-logout mechanism is disabled.

If the interval is set to a value greater than zero, the SM's auto-logout mechanism is enabled.

Note The interval should be smaller, but of the order of the lease time used in the system. It is recommended that during a lease time the auto-logout task will run several times.

The default value for this parameter is 0 (seconds), meaning the auto-logout mechanism is disabled.

Following is an example of assigning a value to this parameter:

```
auto_logout_interval=600
```

Following is an example of deactivating the **Auto Logout** feature:

```
auto_logout_interval=0
```

grace_period

Defines the grace period, in seconds, for each subscriber. After a subscriber’s auto-logout timeout has expired, the subscriber’s IP address is logged out automatically after the grace period has also expired.

The default value for this parameter is 10 (seconds).

Following is an example of assigning a value to this parameter:

```
grace_period=10
```

max_rate

Defines the maximum rate (logouts per second) that the auto-logout task is allowed to perform logouts from the system. This limit spreads out the load of the logout operations over time, reducing the performance impact on other operations.

The value for this parameter should be calculated so it spreads the logouts over at least half of the **auto_logout_interval** time. The default value is 50.

Following is an example of assigning a value to this parameter:

```
max_rate=50
```

Note It is recommended to use the lowest rate possible, so as to reduce the influence of the auto-logout process on other operations. However, as a rule of thumb, the value should be calculated so that the auto-logout process takes about half of the **auto_logout_interval** and be similar to the maximum login rate to the SM .

Radius Listener Section

The `[Radius Listener]` section is used for integrating with the RADIUS Listener LEG.

Note For additional Radius Listener configuration information, see the document *Radius Listener LEG Guidelines*.

This section contains the following parameters:

start

Defines whether the SM should run the Radius Listener at startup.

Possible values for this parameter are `yes` and `no`. The default value is `no`.

Following is an example of assigning a value to this parameter:

```
start=no
```

accounting_port

Defines the Radius Listener's accounting port number.

The default value is `1813`.

Following is an example of assigning a value to this parameter:

```
accounting_port=1813
```

authentication_port

Defines the Radius Listener's authentication port number.

The default value is `1812`.

Following is an example of assigning a value to this parameter:

```
authentication_port=1812
```

Radius.NAS.XXX Section

Each `[Radius.NAS.XXX]` section specifies a single Network Access System (NAS), where XXX represents the NAS name.

Note The Radius Listener LEG refers to all of its Radius clients as NAS devices, even though they might be Radius servers.

This section contains the following parameters:

domain

Specifies the Cisco Service Control's subscriber domain name.

Following is an example of assigning a value to this parameter:

```
domain=my_domain
```

IP_address

Specifies the IP address in dotted notation (xxx.xxx.xxx.xxx).

Following is an example of assigning a value to this parameter:

```
IP_address= 1.1.1.1
```

NAS_identifier

Specifies the name of the NAS that exists in the **NAS-ID** attribute. For use of this parameter, see the SCMS SM *Radius Listener LEG Guidelines* support document.

Following is an example of assigning a value to this parameter:

```
NAS_identifier=RedHat37
```

secret

Specifies a secret key defined in the NAS for this connection.

Following is an example of assigning a value to this parameter:

```
secret=mysecret
```

Radius.Property.Package Section

The `[Radius.Property.Package]` section defines the *RADIUS attribute* from which a subscriber package is retrieved.

Note This section is commented out in the configuration file. If you want to retrieve a subscriber package from an attribute other than *RADIUS attribute*, you should uncomment this section.

This section contains the following parameters:

radius_attribute

Specifies the Radius protocol attribute number. Use the value `26` in case of Vendor Specific Attributes (VSA). Use –1 if you do not want to extract from any attribute.

Following is an example of assigning a value to this parameter:

```
radius_attribute=26
```

radius_attribute_type

Specifies the Radius attribute type.

Possible values for this parameter are `integer` and `string`. The default value is `integer`.

Following is an example of assigning a value to this parameter:

```
radius_attribute_type=integer
```

packet_types

Specifies the Radius protocol packet types in which to look for the attribute.

Possible values for this parameter are `accounting-request`, `accounting-response`, `all`, and/or `none`; where multiple choices are separated by a comma (,). The default value is `all`.

Following is an example of assigning a value to this parameter:

```
packet_types=accounting-request
```

use_default

Defines whether to use a default value if the attribute was not found.

Possible values for this parameter are `true` and `false`. The default value is `true`.

Following is an example of assigning a value to this parameter:

```
use_default=true
```

default

Defines the default value to use if the attribute was not found.

There is no default value. (If this parameter is used, it should have non-empty value.)

Following is an example of assigning a value to this parameter:

```
default=0
```

Additional parameters for VSA association

Use the following additional parameters only if the association is based on Vendor Specific Attributes (VSA):

```
radius_attribute_vendor_id=<insert the vendor identifier number
here>
radius_sub_attribute=<insert the attribute number here>
```

Radius.Subscriber ID Section

The `[Radius.Subscriber ID]` section defines the RADIUS attribute on which subscriber ID association is based. Default association is based on the **user name** attribute.

Note This section is commented out in the configuration file. If you want to base subscriber ID association on an attribute other than **user name**, you should uncomment this section.

This section contains the following parameters:

radius_attribute

Specifies the Radius protocol attribute number. Use the value 26 in case of Vendor Specific Attributes (VSA). Use the value of –1 to use the default.

Following is an example of assigning a value to this parameter:

```
radius_attribute=26
```

radius_attribute_type

Specifies the Radius attribute type.

Possible values for this parameter are integer and string. The default value is integer.

Following is an example of assigning a value to this parameter:

```
radius_attribute_type=string
```

Additional parameters for VSA association

Use the following additional parameters only if the association is based on Vendor Specific Attributes (VSA). For example, to use the 3GPP_IMSI attribute, set:

```
radius_attribute_vendor_id=10415
radius_sub_attribute=1
```

RPC.Server Section

The [RPC.Server] section represents the RPC server configuration.

This section contains the following parameter:

port

Defines the RPC server port. The default is 14374.

Following is an example of assigning a value to this parameter:

```
port=14374
```

SE.XXX Section

Each [SE.XXX] section represents a single SCE Platform, where XXX represents the SCE logical name.

This section contains the following parameters:

ip

Defines the IP address of the SCE box.

Following is an example of assigning a value to this parameter:

```
ip=11.12.13.1
```

port

Defines the port through which to connect to the SCE Platform. The default is 14374.

Following is an example of assigning a value to this parameter:

```
port=14374
```

FTP Section

The SM manages an internal FTP server for various purposes.

The `[FTP]` section contains the following parameters:

start

Defines whether the SM should run the FTP server at startup.

Possible values are `yes` and `no`. The default is `no`.

Following is an example of assigning a value to this parameter:

```
start=yes
```

port

Defines the FTP server port. The default is 21000.

Following is an example of assigning a value to this parameter:

```
port=21000
```

The following parameters define the TCP ports range when the FTP server works in passive mode.

Note: Uncomment these parameters when working with the FTP server via a firewall.

```
#first_passive_port = 21001
#last_passive_port = 21100
```

HTTP Tech-IF Section

The SM manages an internal HTTP adaptor server that is a technician interface.

The `[HTTP Tech-IF]` section contains the following parameters:

start

Defines whether the SM should run the HTTP server at startup.

Note This is a technician interface and normally should not be used.

Possible values are `yes` and `no`. The default is `no`.

Following is an example of assigning a value to this parameter:

```
start=yes
```

port

Defines the HTTP server port. The default is 8082.

Following is an example of assigning a value to this parameter:

```
port=8082
```

Cable Adaptor Section

The SM manages a Cable Adaptor - a special cable support module - that is a translator between the cable world (DHCP events) and the SM. For additional information, see *CPE as Subscriber in Cable Environment* (on page C-1).

The `[Cable Adaptor]` section contains the following parameter:

allow_dynamic_CM

Defines whether logins from CMs that are not in the SM’s database are allowed.

Possible values are `yes` and `no`. The default is `no`.

Following is an example of assigning a value to this parameter:

```
allow_dynamic_CM=no
```

Data Repository Section

The `[Data Repository]` section defines the SM's operation with the TimesTen In-Memory Database.

Note Some of the parameters in this section are discarded on regular configuration loading, and resetting them requires restarting the SM.

The `[Data Repository]` section contains the following parameters:

support_ip_ranges

Defines whether the SM supports IP-Ranges. Disabling this support provides better performance.

Note Resetting this parameter requires restarting the SM. This parameter is discarded on regular configuration loading (using CLU).

Possible values are `yes` and `no`. The default is `yes`.

Following is an example of assigning a value to this parameter:

```
support_ip_ranges=yes
```

checkpoint_interval_in_seconds

Defines the interval, in seconds, for calling the TimesTen checkpoints. Reducing the value affects performance, increasing the value increases vulnerability to power-down.

The default value is 60 (seconds).

Following is an example of assigning a value to this parameter:

```
checkpoint_interval_in_seconds=60
```

max_range_size

Determines the maximum IP range size used in the system.

This parameter is used for improving performance of the SM in Pull mode when the Data Repository section is configured with `support_ip_ranges=yes`.

Note Defining this parameter with too low a value may cause incorrect operation in handling pull requests.

The default value is 256.

Following is an example of assigning a value to this parameter:

```
max_range_size=256
```

APPENDIX B

Command Line Utilities

This appendix describes the Command Line Utilities that come with the Subscriber Manager (SM).

Introduction

The SM provides a set of Command Line Utilities (CLU). The CLU is designed mainly for local use - for viewing and for subscriber management - therefore only subscriber-related CLU changes are persistent. The CLU is used for configuration only indirectly, in that it loads the edited configuration file to the SM.

This appendix describes in detail all the CLU commands: their operations, options, etc. The shorter description of the CLU commands given in *Configuration and Management* (on page 4-1) is more oriented toward the performance of routine management and configuration tasks.

CLU commands are executable only when the user is logged in to the machine using the **pcube** account, which is always installed (see *Installation and Upgrading* (on page 3-1)). In general the CLU runs as a separate process to the configured entity and communicates with it via a predefined communication port and interface. Therefore, the configured entity must keep a certain communication port open at all times, at least locally on the configured machine.

Description of the CLU Commands

This section starts with an introduction to the CLU:

- Informative Output
- Parsing CLU Operations

This section then describes the following CLU commands:

- **p3batch**
- **p3cable**
- **p3clu**
- **p3cluster**
- **p3db**
- **p3domains**

- **p3ftp**
- **p3http**
- **p3inst**
- **p3log**
- **p3net**
- **p3radius**
- **p3rpc**
- **p3sm**
- **p3subs**
- **p3subsdb**

Informative Output

All CLU commands support the following operations for informative output:

Operation	**Description**
--help	Prints the help for the specified CLU command, then exits.
--version	Prints the SM program version number, then exits.

Parsing CLU Operations and Options

A command operation or option that contains any of the following characters must be placed in quotation marks:

- a space character
- a separation sign (comma “,”; ampersand “&”; colon “:”)
- an escape character (backslash “\”)

A command operation or option that contains any of the following characters must have that character preceded by an escape character:

- an equal sign (=)
- a quotation mark (“ or ”)
- an escape character (backslash “\”)

Following are several examples of the above rules:

Operation/option contains the character	**Example of how operation/option should be written**
Space character	--property=“file name”
Escape character (backslash “\”)	--property=“good\\bad”
Equal sign (=)	--property=“x\=y”
Quotation marks (“ or ”)	--name=“\“myQuotedName\””
	(in above example, inner quotation marks are escaped)
Separation character	

• comma (,)	--names=“x,y”
• ampersand (&)	--names=“x&y”
• colon (:)	--names=“myHost:myDomain”:8082

One-letter abbreviations are available for some of the operations and options. For example, "-d" is an abbreviation for "--domain". Note that the letter for an abbreviation is preceded by only one hyphen (-), not two, and that if the operation or option takes a parameter, there is a space and not an equal sign before the parameter.

• Example of using full name	--domain=subscribers
• Example of using abbreviated name	-d subscribers

p3batch Utility

The **p3batch** utility enables you to run many operations on a single connection with the SM. You can use a text editor to create a batch file that contains a series of CLU commands, one command per line (terminated by a new-line sign). Use the **p3batch** utility to run this file and execute the commands, where empty lines are skipped.

All of the batch file’s command line operations use the same connection option (see Configuring Remote Agents). Any connection options in the script file's commands are ignored. While processing the operations in the batch file, a progress indicator is shown.

The command format is: `p3batch [FILE-OPTION] [ERROR-OPTION]`

The following tables list the **p3batch** options.

Table B-24 p3batch File Option

File Operation	Abbreviation	Description	Notes
--file=FILE	-f	Runs a batch file, where FILE specifies the CLU script (batch) file to run.	Progress indicator is shown.

Table B-25 p3batch Error Option

Error Option	Description
--skip-errors	Specifies that the batch operation should *not* halt if an error occurs. If this flag is not used, the batch operation *will* halt if an error occurs.

Examples of using the p3batch utility

- To run a batch file that will halt if an error occurs.

```
p3batch --file=mainBatchFile.txt
```

- To run a batch file that will *not* halt if an error occurs.

```
p3batch --file=mainBatchFile.txt --skip-errors
```

p3cable Utility

In a cable environment, the CPE is modeled as the subscriber, and it inherits its policy and domain from the cable modem (CM) through which it connects to the network. Each cable modem is linked with one or more CPEs. (For background information about special characteristics of the cable environment, see *CPE as Subscriber in Cable Environment* (on page C-1).)

The **p3cable** utility commands are used for importing cable modem information from a CSV file to the SM, and for exporting the cable modem information from the SM to a CSV file. You can also use this utility to clear the repository of all cable modems, and to show whether the login of CPEs that belong to unfamiliar cable modems (cable modems that do not exist in SM database) is allowed or denied. However, for specifying whether such login is allowed or denied, you should use the *Cable Adaptor* (on page 4-11) section of the configuration file *`p3sm.cfg`*.

The command format is: `p3cable <OPERATION> [FILE-OPTIONS]`

The following tables list the **p3cable** operations and options.

Table B-26 p3cable Operations

Operation	Description
--import-cm	Imports cable modems from a CSV file. The filename that is to be imported is specified using the format `--file=import-filename`. The results go to a results file.
--export-cm	Exports cable modems to a CSV file. The filename that is to be exported is specified using the format `--output=export-filename`. The results go to a results file.
--clear-all-cm	Clears the repository of all cable modems.
--show-dynamic-mode	Shows whether the login of CPEs that belong to unfamiliar cable modems (cable modems that do not exist in SM database) is allowed or denied.

Table B-27 p3cable File Options

File Option	Abbreviation	Description
--file=FILE	-f FILE	Specifies the CSV FILE to import from.
--output=FILE	-o FILE	Specifies the subscriber CSV FILE to export to.

Examples of using the p3cable utility

- To import cable modem information from the specified csv file:

```
p3cable –-import-cm  –f CMFile.csv
```

- To export cable modem information to the specified csv file:

```
p3cable –-export-cm  –-outfile=myCMFile.csv
```

- To clear the repository of all cable modems:

```
p3cable --clear-all-cm
```

- To display whether the login of CPEs that belong to unfamiliar cable modems (cable modems that do not exist in SM database) is allowed or denied:

```
p3cable --show-dynamic-mode
```

p3clu Utility

The **p3clu** utility is used for printing a list of all supported CLU utilities and options.

The command format is: `p3clu <OPERATION>`

The following table lists the **p3clu** operations.

Table B-28 p3clu Operations

Operation	Description
--help	Prints the CLU commands that are currently supported.

Example of using the p3clu utility

- To display a listing of all supported CLU utilities and operations:

```
p3clu --help
```

p3cluster Utility

The **p3cluster** utility is used for displaying the redundancy state of a cluster of two SM nodes and its components. This utility also supports operations that alter the redundancy state of the SM. These operations are used by the SM Cluster Agent, as well as for administrative tasks.

The command format is: `p3cluster <OPERATION>`

The following table lists the **p3cluster** operations.

Table B-29 p3cluster Operations

Operation	Description
--show	Displays the redundancy status of the SM and its components.
--active	Make the SM become the active SM in the cluster.
--standby	Make the SM become the standby SM in the cluster.

Example of using the p3cluster utility

- To display the redundancy status of the SM and its components:

```
p3cluster –-show
```

p3db Utility

The **p3db** utility is used for managing and monitoring the TimesTen database. The CLU exposes capabilities of the some of TimesTen CLUs with respect to specific needs of the SM.

The command format is: `p3db <OPERATION> [OPTIONS]`

The following tables list the **p3db** operations and options.

Note

Caution: Take care when activating commands that can affect the database; if used incorrectly, these commands may possibly **damage** the database.

Table B-30 p3db Operations

Operation	Description
--rep-status	Displays status of the replication agent.
--rep-start	Starts the replication agent. **Note:** Use only for database recovery.
--rep-stop	Stops the replication agent. **Note:** Use only for database recovery.
--status	Displays the database status.
--destroy-rep-db	Destroys the replicated data-store.
--destroy-local-db	Destroys the local data-store.
--duplicate	Copies the data-store from the *remote* machine (specified by "--remote=REMOTE_HOSTNAME") to the *local* machine (specified by "--local=LOCAL_HOSTNAME"). **Note:** This option is applicable only for a **cluster setup**. For additional information, see *Data Duplication Procedure* (on page 3-23).
--keep-in-mem [SECS]	Indicates to the database daemon how many seconds to keep the database in the memory, after the last connection to the database is down. This option should be used in large databases to reduce the SM restart time. **Note:** To prevent limitations in performing a database destroy, do not use values above a few minutes (that is, above a few hundred seconds).

Table B-31 p3db Options

Option	Description
--local=LOCAL_HOSTNAME	Specifies the local machine.
--remote=REMOTE_HOSTNAME	Specifies the remote machine.

Examples of using the p3db Utility

- To display the status of the replication agent:

```
p3db ––rep-status
```

p3domains Utility

The **p3domains** utility is used for displaying the subscriber domains. (When a system has more than one SCE Platform, they can be configured into groups, or domains. A subscriber domain is one or more SCE Platforms that share a specified group of subscribers. You must add the SCE Platform to the network and create the domain before you can add an SCE Platform to a domain.)

The command format is: `p3domains <OPERATION> [OPTIONS]`

The following tables list the **p3domains** operations and options.

Table B-32 p3domains Operations

Operation	Description
--show-all	Shows all configured domains.
--show	Shows a domain and its associated network elements.

Table B-33 p3domains Domain/Network Element (NE) Options

Domain/NE Option	Abbreviation	Description	Notes
--domain=DOMAIN	-d DOMAIN	DOMAIN specifies logical name.	"**none**" cannot be used, it is a reserved word.
--ne=ADDRESS		ADDRESS of SCE Platform.	
--ne-name=NAME	-n NAME	Logical NAME for SCE Platform.	

Examples of using the p3domains utility

- To display all configured domains:

```
p3domains -–show-all
```

- To display the specified domain and its associated network elements:

```
p3domains –-show –-domain=myDomain
```

p3ftp Utility

The **p3ftp** utility is used for monitoring the SM's internal FTP server.

The command format is: `p3ftp <OPERATION>`

The following tables list the **p3ftp** operations and options.

Table B-34 p3ftp Operations

Operation	Description	Notes
--show	Displays the port number the FTP server listens to, the passive FTP port range the server uses, the current number of open sessions, the maximum number of sessions supported, and the state (ONLINE/OFFLINE) of the FTP server.	

Examples of using the p3ftp utility

- To display the port number that the FTP server listens to, the passive FTP port range that the server uses, the current number of open sessions, the maximum number of sessions supported, and the state (ONLINE/OFFLINE) of the FTP server.

```
p3ftp --show
```

p3http Utility

The p3http utility is used for monitoring the HTTP adaptor server.

Note The HTTP adaptor server is a technician interface and normally should not be used.

The command format is: p3http <OPERATION>

The following tables list the p3http operations and options.

Table B-35   p3http Operations

Operation	Description	Notes
--show	Displays the port number that the server listens to, the state of the server, and the current number of open sessions.	

Examples of using the p3http utility

- To display the port number that the server listens to, the state of the server, and the current number of open sessions:

```
p3http --show
```

p3inst Utility

The **p3inst** utility is used for installing or uninstalling an application (*pqi* file).

Note Before using p3inst to install an application *pqi* file, read the application installation instructions that came with the application you are using.

The command format is: `p3inst <OPERATION> [FILE-OPTIONS] [ARGUMENT-OPTIONS]`

The following tables list the **p3inst** operations and options.

Table B-36 p3inst Operations

Operation	Abbreviation	Description	Notes
--install	-i	Installs the specified application *pqi* file to the SM. It may be necessary to specify arguments for the installation procedure in the command line. Requires a file option.	Progress indicator
--uninstall		Uninstalls the specified application *pqi* file from the SM. Requires a file option.	Progress indicator
--upgrade		Upgrades an existing application using the specified application *pqi* file. It may be necessary to specify arguments for the upgrade procedure in the command line. Requires a file option.	Progress indicator
--rollback		Returns the specified application to the previous version. Rollback is the reverse of an **upgrade** operation: it undoes the **upgrade**.	Progress indicator
--describe	-d	Displays the contents of the specified application *pqi* file.	
--show-last		Lists the details of the last installed application *pqi* file.	

Table B-37 p3inst File Options

File Option	Abbreviation	Description
--file=FILE[;FILE...]	-f FILE[;FILE...]	Specifies one or more installation FILEs to use; if there is more than one FILE, they should be separated by semicolons.

Table B-38 p3inst Argument Options

Argument Option	Description
--arg=ARG1[,ARG2…]	Specifies one or more arguments for the install and upgrade procedures.

Examples of using the p3inst utility

- To install the specified installation file:

```
p3inst --install --file=myInstallation.pqi
```

- To uninstall the specified installation file:

```
p3inst --uninstall –f oldInstallation.pqi
```

- To upgrade an existing application using the specified application *pqi* file:

```
p3inst –-upgrade –-file=newInstallation.pqi
```

- To upgrade an existing application using the specified application *pqi* file, using arguments in the command line:

```
p3inst –-upgrade –f newInstallation.pqi
```

- To return the specified application to the previous version:

```
p3inst --rollback
```

- To display the contents of the specified application *pqi* file:

```
p3inst ––describe ––file=myInstallation.pqi
```

- To list the details of the last installed application *pqi* file:

```
p3inst --show-last
```

p3log Utility

The **p3log** utility is used for configuring and managing the SM user log. The user log contains all user-related events and errors, and should be used for viewing the history of the system’s events and errors.

The command format is: `p3log <OPERATION> [FILE-OPTIONS]`

The following tables list the **p3log** operations and options.

Table B-39 p3log Operations

Operation	Description	Notes
--extract	Retrieves the user log from the agent.	Progress indicator
--reset	Clears the user log.	

Table B-40 p3log File Option

File Option	Abbreviation	Description
--output=FILE	-o FILE	Specifies where the SM user log file should be extracted to.

Examples of using the p3log utility

- To extract the SM user log to the specified file:

```
p3log ––extract –o aug20.log
```

- To clear the SM user log:

```
p3log ––reset
```

p3net Utility

The **p3net** utility is used for showing connection status of network elements and trying to reconnect disconnected elements.

The command format is: `p3net <OPERATION> [NETWORK-ELEMENT-OPTION]`

The following tables list the **p3net** operations and options.

Table B-41 p3net Operations

Operation	Description
--show-all	Shows all the configured network elements.
--show	Shows an element's connection status/general information.

Operation	Description
--connect	Tries to connect a disconnected element.

Table B-42 p3net Network Element Options

Network Element Option	Abbreviation	Description
--ne-name=NAME	-n NAME	Specifies the logical NAME for the network element.
--detail		(Optional) Used with the **--show-all** operation for displaying additional information as a table.

Examples of using the p3net utility

- To connect a disconnected element to the network:

```
p3net –-connect –n mainNE
```

- To display the names of all configured network elements:

```
p3net –-show-all
```

- To display all configured network elements details (as a table):

```
p3net –show-all --detail
```

- To display the connection status of the specified network element:

```
p3net –-show –-ne_name=mainNE
```

p3radius Utility

The **p3radius** utility is used for viewing the statistics of the RADIUS Listener LEG. For information about this CLU, see the SCMS SM *RADIUS Listener LEG Guidelines* support document.

p3rpc Utility

The **p3rpc** utility is used for displaying the information of the proprietary Cisco RPC (Remote Procedure Call) server interface to the SM.

The command format is: `p3rpc <OPERATION>`

The following tables list the **p3rpc** operations and options.

Table B-43 p3rpc Operations

Operation	Description
--show	Displays the port number the RPC server listens to, the maximum number of connections, the current number of active connections, and the host IP that the server listens to.
--show-client-names	Displays the names of the connected clients. Can be used for extracting the LEG_NAME key (see *LEG-Domains Association Section* (on page A-5)).
--show-statistics	Displays the RPC server statistics. They contain information about the number of current RPC sessions, as well as statistics for RPC server actions such as invocations and errors.

Operation	Description
--reset-statistics	Clears the RPC server statistics.

Examples of using the p3rpc utility

- To display the port number the RPC server listens to, the maximum number of connections, the current number of active connections, the host IP that the server listens to, and the name of the configuration file used by the server:

```
p3rpc ––show
```

- To display the statistics of the RPC server:

```
p3rpc ––show-statistics
```

- To clear the statistics of the RPC server:

```
p3rpc ––reset-statistics
```

p3sm Utility

The **p3sm** utility is used for general configuration and management of the SM.

The command format is: `p3sm <OPERATION> [OPTIONS]`

The following tables list the **p3sm** operations and options.

Table B-44 p3sm Operations

Operation	Description	Notes
--show	Displays the current SM configuration and statistics.	
--load-config	Reloads the SM configuration file. If the **–f** option is not used, file *`p3sm.cfg`* is loaded.	
--resync	Resynchronizes subscribers of specified SCE with SM's database. The SCE is specified using the option `--ne-name=SCE_NAME`.	Progress indicator
--start [--wait]	Starts the server. The option `--wait` causes the CLU to return only after the SM is up.	Default: started
--stop	Stops the server. **Note:** When using fail-over, a simple shut-down of the SM does *not* work (the Veritas Cluster Server identifies that the SM is down and attempts to restart it). The correct procedure is: 1. Perform the manual fail-over, as described in Manual Fail-Over. 2. Use the Veritas Cluster Manager Application to stop the monitoring (probing) of the SM. 3. Use the SM's CLU (the `p3sm` command) to stop the SM.	
--restart [--wait]	Stops the server operation and then restarts it. The option `--wait` causes the CLU to return only after the SM is up.	
--sm-version	Displays the currently installed SM version.	
--sm-status [--detail]	Displays the SM operational status: whether the SM is running or not, and whether it is Active or Standby. If errors have occurred, their summary is displayed. To receive a detailed description, use the option `--detail`.	

Operation	Description	Notes
--extract-support-file	Retrieves the support file from the agent. This command extracts the SM support information to a defined file, which is defined using the option `--output=FILE`. SM support information should be extracted and sent to Cisco's customer support with each support request.	
--reset-sm-status	Clears errors and warning that were displayed to the user.	
--logging=[on/off]	Enables/disables the logging of user logon to the UserLog. **Note:** When this flag is enabled, it may affect performance.	
--show-stats	Displays statistics information regarding logon operations. The rate results are updated once every 10 seconds.	
--reset-stats	Resets the statistics information.	

Table B-45 p3sm SM Options

SM Option	Abbreviation	Description
--ne-name=NAME	-n NAME	Specifies logical NAME of the SCE Platform to resynchronize.

Table B-46 p3sm File Options

File Option	Abbreviation	Description
--output=FILE	-o FILE	Where to extract the support information file to, relative to the SM root directory
--file=FILE	-f FILE	File to load the configuration from, relative to the SM configuration directory.

Table B-47 p3sm Miscellaneous Options

File Option	Abbreviation	Description
--ignore-warnings	-i	Ignore configuration validation warnings while loading the configuration file.
--remote=IP[:port]	-r	Used with **--load-config** to load the local configuration file to both the local SM and the remote SM.
--detail		Display a detailed view of the SM-status.
--wait		Used with **--start** or **--restart** to signal the CLU to return only when the SM is up.

Examples of using the p3sm utility

- To start the server:

```
p3sm --start
```

- To stop the server:

```
p3sm --stop
```

Note When using fail-over, a simple shut-down of the SM does **not** work (the Veritas Cluster Server identifies that the SM is down and attempts to restart it). The correct procedure is:
1. Perform the manual fail-over, as described in Manual Fail-Over;
2. Use the Veritas Cluster Manager Application to stop the monitoring (probing) of the SM;
3. Use the SM's CLU (p3sm --stop) to stop the SM.

- To display the SM configuration:

```
p3sm --show
```

- To resynchronize the subscribers of the specified SCE with the SM’s database:

```
p3sm --resync --ne-name=my_SCE_100
```

- To stop the server operation and then restart it:

```
p3sm --restart
```

- To reload the SM configuration file, *p3sm.cfg*:

```
p3sm --load-config
```

- To display the SM operational status (active or inactive):

```
p3sm --sm-status
```

- To extract the SM support information to the specified file.

```
p3sm --extract-support-file --output=support.zip
```

p3subs Utility

The **p3subs** utility is used for manipulating individual subscriber information in the SM database. Regarding properties, the property names depend on the application running on your system. To find descriptions of the application property names and values, see the documentation provided with the application installed on your system.

The command format is: `p3subs <OPERATION> [SUBSCRIBER-OPTIONS]`

The following tables list the **p3subs** operations and options.

Table B-48   p3subs Operations

Operation	Description
--add	Adds/updates a subscriber. The operation fails if the subscriber exists, unless the **--overwrite** option is used.
--set	Adds/updates mappings and/or properties for the specified subscriber. A new mapping overwrite all existing mappings, unless the **--additive-mapping** option is used. A property is overwritten only when a new value is assigned to it, but not when a *different* property has a new value assigned to it. For example, specifying a new value for one property, rdr.transaction.generate, does **not** cause some other property, such as rdr.subscriber.generate, to be overwritten.
--show	Displays information for specified subscriber.

Operation	Description
--remove	Removes the specified subscriber.
--show-all-mappings	Displays all the mappings for specified subscriber.
--remove-mappings	Removes specified mapping of specified subscriber.
--remove-all-mappings	Removes all the mappings of specified subscriber.
--show-property	Displays value of specified property of specified subscriber.
--show-all-properties	Displays the values of all the properties of specified subscriber.
--show-all-property-names	Displays all the property names and descriptions.
--reset-property	Resets specified property of specified subscriber to its default value.
--remove-properties	Removes all properties and custom properties from the subscriber record.
--clear-state	Clears applicative state of specified subscriber. This command clears **only** the backup copy at the SM; it does **not** clear the applicative state record in the SCE Platform.

Table B-49 p3subs Subscriber Options

Subscriber Option	Abbreviation	Description
--overwrite		Used in add operations to replace the existing subscriber configuration, instead of failing.
--subscriber=NAME	-s NAME	Performs operation using specified subscriber NAME.
--additive-mappings		Adds the new mapping(s) to any existing ones. (Without this option, any existing mappings are overwritten.)
--ip= IP1[/RANGE][,…]		Performs operation using specified IP mapping(s). IP is in dotted notation. "/RANGE" is used for specifying several consecutive mappings, by specifying the number of consecutive set bits in the mask. For example, **1.1.1.0/30** means **1.1.1.0 to 1.1.1.3**, or **1.1.1.0 with mask 255.255.255.252**.
--vlan-id=VLAN1[,…]		Performs operation using specified VLAN mapping(s).
--property= KEY1[=VAL1][;...]	-p KEY1[=VAL1][;...]	Performs operation using the specified KEY=VAL property/properties. These properties are defined by the application and influence the subscriber service in the SCE.
--custom-property= KEY1[=VAL1][;...]		Performs operation using the specified KEY=VAL custom property/properties. These properties are user defined and have no influence on the service the subscriber receives.
--domain=DOMAIN	-d DOMAIN	Performs operation on specified DOMAIN. If DOMAIN is **none**, the operation refers to subscribers who have no domain specified.

Examples of using the p3subs utility

- To add a subscriber with the specified IP address:

```
p3subs --add --subscriber=jerry --ip=96.142.12.7
```

- To overwrite subscriber information (because the subscriber **jerry** already exists, this operation would fail, but the **overwrite** option allows the IP address to be overwritten):

```
p3subs --add --subscriber=jerry --ip=96.128.128.42 --overwrite
```

- To set a property value for the specified subscriber:

```
p3subs --set --subscriber=jerry --property=rdr.transaction.generate=true
```

- To add new mappings for the specified subscriber; any existing ones are overwritten:

```
p3subs --set --subscriber=jerry  -–vlan-id=1
```

- To add new mappings to the existing ones for the specified subscriber:

```
p3subs --set --subscriber=jerry -–vlan-id=4,2 --additive-mappings
```

- To display information for the specified subscriber:

```
p3subs --show --subscriber=jerry
```

- To remove the specified subscriber:

```
p3subs --remove --subscriber=jerry
```

- To display all the mappings for the specified subscriber:

```
p3subs --show-all-mappings --subscriber=jerry
```

- To remove the specified mappings for the specified subscriber:

```
p3subs --remove-mappings --subscriber=jerry --ip=96.142.12.7,96.128.128.42
```

- To remove a range of consecutive mappings for the specified subscriber:

```
p3subs --remove-mappings --subscriber=jerry --ip=1.1.1.0/30
```

- To remove all the mappings for the specified subscriber:

```
p3subs --remove-all-mappings --subscriber=jerry
```

- To display the value of the specified property for the specified subscriber:

```
p3subs --show-property --subscriber=jerry –-property=reporting
```

- To display the values of all the properties for the specified subscriber:

```
p3subs --show-all-properties --subscriber=jerry
```

- To display all the property names and descriptions:

```
p3subs --show-all-property-names
```

- To reset specified property of specified subscriber to its default value:

```
p3subs –-reset-property --subscriber=jerry
--property=rdr.transaction.generate
```

- To clear the applicative state of the specified subscriber: This command clears only the backup copy at the SM, it does **not** clear the applicative state record in the SCE Platform.

```
p3subs –-clear-state --subscriber=jerry
```

p3subsdb Utility

The **p3subsdb** utility is used for managing the subscriber database, and for operations on groups of subscribers.

The command format is: `p3subsdb <OPERATION> [OPTIONS] [FILE-OPTIONS]`

The following tables list the **p3subsdb** operations and options.

Table B-50 p3subsdb Operations

Operation	Description	Notes
--clear-all	Removes all subscriber records from the SM database.	Progress indicator
--clear-domain	Removes all subscriber records from the specified domain.	Progress indicator
--show-num	Displays number of subscribers in database for the specified domain.	
--show-all	Lists all the subscriber names.	
--show-domain	Lists all the subscriber names in the specified domain.	
--import	Imports subscribers to the database from a specified CSV file. The filename that is to be imported is specified using the format "--file=import-filename". The results go to a results file, *`import-results.txt`*, which is created in the same directories as the CSV file.	Progress indicator
--export	Exports subscribers from the database to a specified CSV file. The filename that is to be exported is specified using the format "--output=export-filename". The results go to a results file, *`export-results.txt`*, which is created in the same directory as the CSV file.	Progress indicator
--clear-all-states	Clears the state of all subscribers in the SM database.	

Table B-51 p3subsdb Options

Option	Abbreviation	Description
--prefix=NAME		Used in the export operation for filtering the export.
--domain=DOMAIN	-d DOMAIN	Performs the operation on the specified DOMAIN. If DOMAIN is **none**, the operation refers to the subscribers who have no domain specified.

Table B-52 p3subsdb File Options

File Option	Abbreviation	Description
--file=FILE	-f FILE	Specifies the subscriber CSV FILEs to import from.
--output=FILE	-o FILE	Specifies the subscriber CSV FILE to export to.

Examples of using the p3subsdb utility

- To import subscribers from a specified CSV file:

```
p3subsdb --import --file=mySubscriberFile.csv
```

- To export subscribers to a specified CSV file:

```
p3subsdb --export -o mySubscriberFile.csv
```

- To export subscribers to a specified CSV file, using filtering options:

```
p3subsdb --export --prefix=a -–output=mySubscriberFile.csv
```

- To export subscribers to a specified CSV file, using filtering options:

```
p3subsdb --export --prefix=a –o a.csv
```

- To remove all subscriber records from the SM database:

```
p3subsdb –-clear-all
```

- To remove all subscriber records from the specified domain:

```
p3subsdb –-clear-domain --domain=myDomain
```

- To list all the subscribers:

```
p3subsdb --show-all
```

- To list all subscribers in a specified domain:

```
p3subsdb --show-domain --domain=myDomain
```

- To show the number of subscribers in a specified domain:

```
p3subsdb --show-num --domain=myDomain
```

- To list all subscribers who have no domain specified:

```
p3subsdb –-show-domain --domain=none
```

- To clear the state of all subscribers in the SM database:

```
p3subsdb –-clear-all-state
```

APPENDIX C

CPE as Subscriber in Cable Environment

The cable market presents special issues in terms of subscribers, in addition to the normal subscriber management issues that exist in other markets, such as DSL and Wireless.

This appendix deals with the special case when the CPE is considered as the subscriber in Cisco Service Control's solution for a cable environment. *Note that this appendix is* **not** *relevant for the more common case where the cable modem with all the CPEs behind it is considered the subscriber.*

This appendix makes use of the following cable terms:

- Customer Premises Equipment (CPE): The CPE is an individual computer. The cable support module considers the CPE as the subscriber.
- Cable Modem (CM): There may be multiple CPEs connected through a single CM. Although the CPE is the subscriber, the policy is assigned to the CM, and the CPEs inherit the policy from the CM. All CPEs connected through the same CM always have the same policy.
- Cable Modem Termination System (CMTS): The CMTS groups several thousand CMs and connects them to the network. Several CMTS devices can be served by the same SCE Platform and thus belong to the same domain.
- DHCP Server: Assigns IP addresses and provides the login/logout event information to the system.

Cable Support Module

The Subscriber Manager (SM) includes a special cable support module (**p3cable**) for dealing with the special case where the CPE is considered as a subscriber in a cable environment. The cable support module functions as a translator between the cable world (DHCP events) and the Cisco SM, for this special case. It provides an API on top of the basic SM API functionality. This API is accessible using the Java/C/C++ APIs by calling the `cableLogin` and `cableLogout` methods.

To ensure the correct behavior of the cable support module, certain configuration steps are necessary, such as the correct domain configuration and the static/dynamic CM configuration.

The cable support module, which translates between the SM and the DHCP events in the cable world, performs the following functions:

- associates between CPEs and CMs
- makes CPEs inherit application policy from their CM
- allows/denies the introduction of CPEs whose CM is unfamiliar to the SM

(For additional information regarding the functions of the cable support module, see *p3cable Utility* (on page B-4).)

The cable support module uses the hardware (MAC) addresses of the CM as its subscriber name. The subscriber name of the CPE is the hardware address of its CM followed by the hardware address of the CPE.

CM and CPE in the SM

In the special case when CPEs are considered as subscribers, cable modems are not delegated to the SCE in any way, and are not considered as subscribers in Cisco Service Control's solution. However, for ease of integration and for the sake of simplicity, CMs are saved as subscribers in the SM only (but are **never** introduced to the SCE).

Cable modem SM subscribers are saved in special hidden subscriber domains called *CM domains*. These CM domains do not contain any SCE and are created automatically upon an insertion of a CM. For a CPE in a given subscriber domain, its CM will reside in a CM domain having the same name as the CPE domain but with the prefix **CM_**.

Because CM domains are hidden, they cannot be configured by the configuration file. However, it is possible to run subscriber-related commands (**p3subs** and **p3subsdb**) on these domains.

A CM subscriber's name has the following form: <CM MAC> (the MAC of the CM as sent in the DHCP protocol).

A CPE subscriber's name (for such a CM) has the following form: <CM MAC>__<CPE MAC> (the MAC of the CM, followed by **two** underscore signs, followed by the MAC of the CPE).

Importing and exporting cable modems is performed using the **p3cable** command, similar to importing and exporting subscribers, except that it is unnecessary to import the CM with an IP address.

Note When importing cable modems, the **full** CM domain name (**CM_** plus the domain name of its CPEs) **must** be provided.

Example:

> In the configuration of this example, the SM has a domain called DomainA. We want CPEs arriving from CMTS with IP 1.2.3.4 to reach this domain, therefore we have configured 1.2.3.4 as an alias of DomainA.
>
> During operation, as a result of a DHCP request-response, the DHCP LEG event sends a login event of a cable modem with MAC 0X0Y0Z from CMTS 1.2.3.4.
>
> In the login event, the alias sent was 1.2.3.4 (the alias of domain DomainA), so the cable modem subscriber will be entered into domain CM_DomainA with the name 0X0Y0Z.
>
> When a login event of its CPE with MAC 0A0B0C is sent with the same alias (as the CPE that arrived from the same CMTS), the CPE subscriber will be entered into domain DomainA with the name 0X0Y0Z__0A0B0C.

Static and Dynamic CMs

Login/logout events of CPEs whose CM does not exist in the subscriber database will be ignored, since no subscriber will be create in the SM and aggregated to the SCE. This CPE traffic will be treated as default subscriber.

The SM supports two modes of integrating with cable modems. These modes are controlled by editing and loading the *p3sm.cfg* configuration file. (Configuring dynamic CM support is described in *Configuration File Options* (on page A-1).) The current status can be viewed by using the CLU **p3cable**.

- **Deny dynamic CM:** In this mode, login /logout events of cable modems that were **not** imported using the **p3cable** command will be ignored. Consequently, the CPE traffic of these CMs will be treated as default subscriber.
- **Allow Dynamic CM:** In this mode, login/logout events of cable modems that were **not** imported using the **p3cable** command will result in automatic addition of the cable modem to the subscriber data base. These cable modems will receive the tuneables that were defined in the domain tunable template section of the configuration file.

APPENDIX D

Troubleshooting

This appendix describes how to troubleshoot the SM’s installation and daily operation.

Using Troubleshooting Chapter

Each entry in this appendix consists of an error message, probable cause(s), and solution. Note that the same error message may appear in more than one section of this appendix.

When an unexpected error occurs during the system’s installation or daily operation, search for the error message throughout this appendix (the message may appear in more than one place). When you find the error message, read the section below the message and try the recommended solution. If the message appears more than once, try to correct the most probable cause first.

General Errors

SM Not Running

Error message

The following sequence of output appears (in response to the command `p3sm --sm-status`):

```
> p3sm –-sm-status
Could not connect to SM.
```

Probable cause

The SM server has not been started.

Solution

Use the following command to start/restart the SM server:

```
> p3sm –-start
```

SM in Failure Mode

Error message

The following sequence of output appears (in response to the command `p3sm --sm-status`):

```
> p3sm –-sm-status
SM is running.
SM operational state is Failure
Command terminated successfully
```

Probable cause

The SM server restarted three times in 30 minutes due to an internal error.

Solution

This error can happen only in a cluster setup. Check the pcube user log and the Veritas Cluster Server log for the reason for the failure that caused the reboots. Act according to the problem in the logs.

Additional operations that can be taken are:

- Use the following command to extract a support file:

  ```
  > p3sm –-extract-support-file –f ../support.zip
  ```

  Send the support file to Cisco's customer support (see *Technical Support* (on page x))

- Use the following command to start/restart the SM server and get out of the Failure state:

  ```
  > p3sm –-restart
  ```

General Setup Errors

Install-sm.sh Script – user Is Not root

Error message

The following sequence of output appears (in response to the command `./install–sm.sh`):

```
> ./install-sm.sh
install-sm.sh: Starting SM installation sequence
install-sm.sh: Error - this script must be run by root - exiting.
```

Probable cause

You started the installation sequence as **user** and not as **superuser**.

Solution

Run the *`install-sm.sh`* script as **superuser**.

Install-sm.sh Script – user pcube Exists

Error message

The following sequence of output appears (in response to the command `./install-sm.sh`):

```
# ./install-sm.sh
install-sm.sh: Starting SM installation sequence
install-sm.sh: Error - pcube user exists and has home /export/home/pcube,
not /opt/pcube - remove it or use -o - exiting.
#
```

Probable cause

Your machine already has the user **pcube**.

Solution

Run the installation using the **–o** option (overwrite), as follows:

```
# ./install-sm.sh -o
```

Install-tt.sh Script

Note A minimum of 1.5 GB of free hard disk space is needed to install the TimesTen database.

Error message

The following sequence of output appears (in response to the command `install-tt.sh/export/home/pcube/lib/tt`):

```
> install-tt.sh /export/home/pcube/lib/tt
install-tt.sh: Starting TimesTen P-Cube installation sequence
install-tt.sh: Error - This script must be run by root - exiting.
```

Probable cause

You started the installation sequence as **user** and not as **superuser**.

Solution

Run the *`install-tt.sh`* script as **superuser**.

Note that the TimesTen directory name given, *`/lib/tt/`*, **must** be relative to the (pcube) user directory. For example, if the user directory is pcube, install TimesTen in *`/export/home/pcube/lib/tt/`*.

Install-dsn.sh Script

Note that the TimesTen directory name given, *`/var/tt/`*, **must** be relative to the (pcube) user directory. For example, if the user directory is pcube, install TimesTen in *`/export/home/pcube/var/tt/`*.

TimesTen Database Setup Errors

Introduction

The TimesTen configuration consists of several configuration files. This section explains the purpose and scope of each of these files. When troubleshooting TimesTen you will be requested to edit these configuration files, and then to reboot the machine or restart the SM. Note that in most cases the defaults applied by the SM installation procedure are satisfactory.

Note Changing the TimesTen configuration files should be done with **extreme care**, and it is best to **consult Cisco technical support** prior to making the change.

System (Kernel) Configuration File

The kernel configuration file is a system configuration file, which affects system-wide configuration parameters:

- For Solaris, it is file *`/etc/system`*.
- For RedHat, it is file *`/etc/sysctl.conf`*.

The Subscriber Manager’s installation procedure configures this file to add extra semaphores and shared memory to the system. After editing this file, you have to reboot the machine for the changes to take effect.

If you are running other applications that require changes in this file’s semaphore and shared memory values, **take care** that the TimesTen configuration does not override the other application’s configuration, or vice versa. **In particular**, when installing Cisco's Collection Manager (CM) and SM on the same machine, Cisco's technical support should be consulted for the proper values to use for the file’s configuration parameters.

Configuration File /var/TimesTen/sys.odbc.ini

The file *`/var/TimesTen/sys.odbc.ini`* is a TimesTen configuration file that configures system DSNs. Any user on the machine on which the system DSN is defined can use this file. The SM DSNs are system DSNs that are named PCube`_SM_Repository` and `PCube_SM_Local_Repository`, and which have the following system DSN configuration parameters:

- LogFileSize: The size of the TimesTen log file, in megabytes.
- PermSize: The size of the permanent memory region for the data store, in megabytes. You may increase PermSize but not decrease it.

  The data stored in the permanent memory region includes tables and indexes that make up a TimesTen data store. The permanent data partition is written to the disk periodically.
- TempSize: The size of the memory allocated to the temporary region, in megabytes.

  Temporary data includes locks, cursors, compiled commands, and other structures needed for command execution and query evaluation. The temporary data partition is created when a data store is loaded into memory and is destroyed when the data store is unloaded.

Note For additional information, see the recommended database parameter configuration table in RAM and Configuration Parameters Versus Number of Subscribers.

- SMPOptLevel: Optimizes the database operation on multi-processor machines. If the machine is a multi-processor platform, set parameter SMPOptLevel to 1 (default is 0).

Configuration File ~pcube/.odbc.ini

The file *~pcube/.odbc.ini* is a TimesTen configuration file that configures user DSNs.

TimesTen DSN Configuration - Cannot Find Requested DSN

Error message

The following sequence of output appears (in response to the command `p3sm –-sm-status`):

```
> p3sm –-sm-status
SM is running.
SM operational state is XXX
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 CLIENT]Cannot find the requested DSN
(PCube_SM_Repository_CS) in ODBCINI
```

Probable cause

The TimesTen Client DSN is not configured correctly in file ~pcube/.odbc.ini.

Solution

Ensure that file *~pcube/.odbc.ini* contains the following:

```
[ODBC Data Sources]
PCube_SM_Repository_CS=TimesTen 5.0 Client Driver
```

TimesTen DSN Configuration - Data Source Name Not Found

Error message

The following sequence of output appears (in response to the command `p3sm ––sm-status`):

```
> p3sm ––sm-status
SM is running.
SM operational state is XXX
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 ODBC Driver]Data source name not found and no
default driver specified
```

Probable cause

The TimesTen Client DSN is not configured correctly in file ~pcube/.odbc.ini.

Solution

Ensure that file *~pcube/.odbc.ini* contains the following:

```
[PCube_SM_Repository_CS]
TTC_SERVER_DSN=PCube_SM_Repository
```

Ensure that file */var/TimesTen/sys.odbc.ini* contains the following:

```
[ODBC Data Sources]
PCube_SM_Repository=TimesTen 5.0 Driver
[PCube_SM_Repository]
Driver=__TTDIR__/TimesTen/pcubesm22/lib/libtten.so
DataStore=__VARDIR__/pcube_SM_Repository
```

TimesTen Database Settings - Cannot Connect to Data Source

Error message

The following sequence of output appears (in response to the command `p3sm --sm-status`):

```
> p3sm -–sm-status
SM is running.
SM operational state is XXX
Error - Times-Ten DB is not set up correctly:
[TimesTen][TimesTen 5.0.35 CLIENT]Unable to connect to data source (DSN:
pcube_SM_Repository_CS; Network Address: X.X.X.X; Port Number: XXX): This
operation has Timed Out. Try increasing your ODBC timeout attribute or check
to make sure the target TimesTen Server is running
```

Probable cause

The following causes are possible:

- The address of the Server DS is incorrect.
- The port of the Server DS is incorrect.
- TimesTen is not active.

Solution

The solutions for the above causes are:

- (The address of the Server DS is incorrect.) Ensure that file ~pcube/.odbc.ini *contains the fol*lowing:

```
TTC_SERVER=127.0.0.1
```

- (The port of the Server DS is incorrect.) On a default installation, ensure that file *~pcube/.odbc.ini* does *not* contain *TTC_SERVER_PORT*.
  On a non-default installation, ensure that file *~pcube/.odbc.ini* *does* contain *TTC_SERVER_PORT=Non-default-port*.
- (TimesTen is not active.) Run the following command:

  ```
  ~pcube/lib/tt/TimesTen/pcubesm22/bin/ttStatus
  ```

  If TimesTen is not working, re-install TimesTen.

If the above solutions do not work, please refer to the TimesTen manual.

TimesTen Configuration Error - Not Enough Memory

Error message

The following sequence of output appears (in response to the command `p3sm --sm-status`):

```
> p3sm --sm-status
SM is running.
SM operational state is XXX
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 ODBC Driver][TimesTen]TT0836: Cannot create data
store shared-memory segment, error 1455 -- file "db.c", lineno 6289,
procedure "sbDbConnect()"
```

Probable cause

There is not enough memory for creating TimesTen's in-memory database.

Solution

Do **all** of the following:

- Ensure that the Unix machine has at least 1024 MB of memory installed.
- Ensure that the configured memory size parameters (*PermSize and TempSize*) specified in file */var/TimesTen/sys.odbc.ini* are less than the total amount of memory installed in the machine.
- For **Solaris**, ensure that the maximum shared memory (parameter *shmsys:shminfo_shmmax*) specified in file */etc/system* is less than the total amount of memory installed in the machine.
- For **Red Hat**, ensure that the maximum shared memory (parameter *kernel.shmmax*) specified in file */etc/sysctl.conf* is less than the total amount of memory installed in the Linux Machine.

TimesTen Configuration Error - Incorrect Memory Definitions

Error message

The following sequence of output appears (in response to the command `p3sm --sm-status`):

```
> p3sm -–sm-status
SM is running.
SM operational state is XXX.
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 ODBC Driver]Overflow in converting data store or
log file size from megabytes to bytes, or in converting log buffer size from
kilobytes to bytes
```

Probable cause

The memory definitions of DSN are incorrect.

Solution

Ensure that the configured permanent memory size and log file size (parameters *PermSize and LogSize*) are less than the total amount of memory and of disk space specified in file *`/var/TimesTen/sys.odbc.ini`*.

TimesTen Configuration Error - Cannot Create Semaphores

Error message

The following sequence of output appears (in response to the command `p3sm ––sm-status`):

```
> p3sm ––sm-status
SM is running.
SM operational state is XXX
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 ODBC Driver][TimesTen]TT0925: Cannot create data
store semaphores (Invalid argument) –
file "db.c", lineno 5124, procedure "sbDbCreate()", sqlState: 08001,
errorCode: 925
```

Probable cause

TimesTen was unable to create the data store semaphores that are defined in the kernel configuration file (**Solaris**: *`/etc/system`* for Solaris; for **Red Hat**: *`/etc/sysctl.conf`* for Red Hat).

Solution

Do **all** of the following:

- Ensure that the machine has at least 1024 MB of memory installed.
- Reboot the machine after the first time that TimesTen is installed.
- Verify the contents of the system (kernel) configuration file:
  - For **Solaris**, ensure that file *`/etc/system`* contains the following:

    ```
    semsys:seminfo_semmni = 20
    semsys:seminfo_semmsl = 100
    semsys:seminfo_semmns = 2000
    semsys:seminfo_semmnu = 2000
    ```

  - For **Red Hat**, ensure that file *`/etc/sysctl.conf`* contains the following:

```
*---- Begin settings for TimesTen
kernel.sem = “SEMMSL_250 SEMMNS_32000 SEMOPM_100 SEMMNI_100
*---- End of settings for TimesTen
```

TimesTen Configuration Error - Cannot Read Data Store File

Error message

The following sequence of output appears (in response to the command `p3sm ––sm-status`):

```
> p3sm ––sm-status
SM is running.
SM operational state is XXX.
Error - Times-Ten DB is not setup correctly:
[TimesTen][TimesTen 5.0.35 ODBC Driver][TimesTen]TT0845: Cannot read data
store file. OS-detected error: Error 0 -- file "db.c", lineno 6320,
procedure "sbDbConnect()"
```

Probable cause

TimesTen was unable to read the data store file, probably due to an error during the installation. This error occurs when installing a TimesTen application on top of an existing TimesTen, without having first uninstalled the old TimesTen database.

Solution: Perform the following procedure:

Step 1 Remove the database, by using the SM p3db CLU with the following commands:

```
>p3db –destroy-rep-db
>p3db –destroy-local-db
```

Step 2 Uninstall TimesTen with the following commands:

```
> su
Password:
# ~pcube/lib/tt/TimesTen4.5/32/bin/setup.sh -uninstall
```

Step 3 Re-install TimesTen, either by running the SM *`install-tt.sh`* script or by using the installation files supplied by TimesTen.

TimesTen Configuration Error - Data Store Space Exhausted

Error message

The following sequence of output appears in the SM log (while using the SM APIs):

```
java.io.IOException: Failure in putting subscriber
45977166__00:50:bf:97:c1:b2 : [TimesTen][TimesTen 5.0.35 ODBC
Driver][TimesTen]TT0802: Data store space exhausted -- file "blk.c", lineno
1571, procedure "sbBlkAlloc"
```

Probable cause

The TimesTen database has already reached its maximum capacity, which caused the operation of adding a new subscriber to the database to fail.

Solution

Usually, doing just one of the following is sufficient:

- Reduce the number of the subscribers handled by the SM (of course, this solution is not always possible).
- Configure the system to support a larger number of subscribers. Note that this solution may require editing one or more of the TimesTen configuration files discussed in *Introduction* (on page D-4), as well as rebooting the machine.
- Move the SM to a more powerful machine; this could be a faster CPU (or more CPUs), a larger disk, more RAM, etc.

For help and guidance in implementing the last two solutions, please contact Cisco *technical support* (on page x).

Network Management Command Line Utility (p3net) Errors

First Connection - Operation Timed Out

Error message

The following sequence of output appears (in response to the command `p3net --connect`):

```
> p3net --connect --ne-name=YYYY
Error - failed to connect to element 'YYYY'
Operation timed out: connect
```

Probable cause

The following causes are possible:

- The IP address is incorrect.
- The element YYYY is down.

Solution

The solutions for the above causes are:

- Ensure that the IP address is correct.
- Ensure that the element YYYY is online and is connected via its management port.

Status Error – Connection Down

Error message

The following sequence of output appears (in response to the command `p3net --show-ne`):

```
>p3net --show --ne-name=se0
Network Element Information:
============================
Name:        YYY
Description: testing element
Host:        X.X.X.X
Ip:          X.X.X.X
Port:        14374
Status:      Connection down (Failure in connecting to agent on host,
Connection refused: connect)
Type:        SCE1000
Domain:        smartNET.policy.unitTestSubscribers
Subscriber Management: Not Active
```

Probable cause

The following causes are possible:

- The IP address is incorrect.
- The element YYY is down.

Solution

The solutions for the above causes are:

- Ensure that the IP address is correct.
- Ensure that the element YYY is online and is connected via its management port.
- Ensure that the RPC adaptor is online on the port that the status indicates.

Status Error – Subscriber Management Down

Error message

The following sequence of output appears (in response to the command `p3net --show-ne`):

```
>p3net --show --ne-name=se0
Network Element Information:
============================
Name:        YYY
Description: testing element
Host:        X.X.X.X
Ip:          X.X.X.X
Port:        14374
Status:      Connection ready
Type:        SCE1000
Domain:        smartNET.policy.unitTestSubscribers
Subscriber Management: Not Active
```

Probable cause

The Subscriber Management field indicates whether the SM successfully performed SM-SCE subscriber synchronization. If the value of the field is *Not Active*, it is possible that the SM failed to synchronize the SCE.

Solution

One possible solution is to force SM-SCE resynchronization by using the CLU command `p3sm --resync`.

Subscriber Database Command Line Utility (p3subsdb) Errors

CSV File Validation Error

Error message

The following sequence of output appears (in response to the command p3subsdb `--import`):

```
> p3subsdb --import --file=/export/home/pcube/XXX.csv
Error - Failed to validate the file XXX.csv

See import-results.txt for detailed errors description.

> cat import-results.txt
x.csv:1: expected 2 items but got 4 items.
1 subscribers, 1 errors.
NO APPLICATION INSTALLED, MAKE SURE TO INSTALL PQI BEFORE IMPORTING CM WITH
TUNEABLES.
```

Probable cause

You tried to import a four-field *csv* file to the SM, but no application (*SCAS BB* or SCAS M) was installed.

For example, the following *csv* file for a *SCAS BB* application contains four fields:

```
# CSV line format: subscriber-id, domain, mappings, package-id
JerryS,subscribers,80.179.152.159,0
ElainB,subscribers,194.90.12.2,3
```

However, the default definition file that defines *csv* file parsing rules contains only two fields: **name** and **ip mapping**.

Solution

Do one of the following:

- Install an application (*SCAS BB* or *SCAS M pqi* file) on the SM (for details, see the *SCAS BB User Guide* or the *SCAS M User Guide*).
- Import a *csv* file that has just two fields.

Cable Support Command Line Utility (p3cable) Errors

CSV File Import Error

Error message

The following sequence of output appears (in response to command `p3cable –-import-cm`):

```
> p3cable –-import-cm --file=/export/home/pcube/XXX.csv
Importing cable modems ... 0%
Importing cable modems ... 100%
Error - Errors during import from
'H:\work\Mng\dev\install\ems\bin\win32\x.csv':
Imported 1 CM(s).  1 Error(s).
See cm-import-results.txt for detailed errors description.

> cat cm-import-results.txt
x.csv:1: expected 2 items but got 4 items.
1 cable modem(s); 1 error(s).
NO APPLICATION INSTALLED, MAKE SURE TO INSTALL PQI BEFORE IMPORTING CM WITH
TUNEABLES.
```

Probable cause

You tried to import a four-field *csv* file to the SM, but no application (*SCAS BB* or SCAS M) was installed.

For example, the following *csv* file for a *SCAS BB* application contains four fields:

```
# CSV line format: subscriber-id, domain, mappings, package-id
JerryS,subscribers,80.179.152.159,0
ElainB,subscribers,194.90.12.2,3
```

However, the default definition file that defines *csv* file parsing rules contains only two fields: **name** and **ip mapping**.

Solution

Do one of the following:

- Install an application (*SCAS BB* or *SCAS M pqi* file) on the SM (for details, see the *SCAS BB User Guide* or the *SCAS M User Guide*).
- Import a *csv* file that has just two fields.

Configuration Errors

The errors in the following sections may appear after the CLU command `p3sm –-load-config` or in a user log during the SM startup procedure.

Network Management Errors

Error message (1)

```
Error section  <section name>: cannot contain white spaces.
```

Probable cause

- [SE.XXX] section cannot contain white spaces (SCE name cannot contain white space).

Solution

- Remove the white spaces.

Error message (2)

```
Error in section <section name>: host  <ip address> already exists in
section <section name>
```

Probable cause

- Configuration cannot contain two SCEs with the same IP address.

Solution

- Change the IP address of one of the SCEs.

Error message (3)

```
Unknown NE <name> found in domain <domain name> section: it does not have
[SE.<name>] section
```

Probable cause

- The section <domain name> includes, under the `elements` property, an SE that is not defined in an [SE.xxx] section.

Solution

- Add the missing [SE.xxx] section to the file.

Error message (4)

```
Duplicate NE <name> found in domain <domain name> section: it already
appears in < domain name > domain section.
```

Probable cause

- Same SCE cannot belong to more than one domain.

Solution

- Remove the SCE from all but one of the domains.

Domain Errors

Error message (1)

```
Error section <section name>: cannot contain white spaces.
```

Probable cause

- [Domain.XXX] section cannot contain white spaces (domain name cannot contain white spaces).

Solution

- Remove the white spaces.

Error message (2)

```
Error  <alias name> value - alias name should not start with ‘CM_’.
```

Probable cause

- The alias name cannot start with `CM_` because this is the prefix of hidden domain generated by the SM when working with CMs (see *CPE as Subscriber in Cable Environment* (on page C-1)).

Solution

- Use a different prefix for the alias.

Error message (3)

```
Alias <alias name> already exists in [<domain name>] section
```

Probable cause

- Each alias can appear in only one [Domain.XXX] section.

Solution

- Alias mentioned in the error message should be removed from all but one [Domain.XXX] section.

Error message (4)

```
Unknown domain <domain name> found in [LEG-Domains Association]. It does not
appear as a section.
```

Probable cause

- Domain mentioned in given section does not have a [Domain.XXX] section.

Solution

- Domain mentioned in error message should be given a [Domain.XXX] section.

Error message (5)

```
Invalid non-integer value: <value> for property '<property name>' in section
[section name].
```

Probable cause

- Properties in [Domain.XXX] sections do not have integer values.

Solution

- Properties mentioned in error message should be given integer values.

Error message (6)

```
Error in section <domain name>: Property - <name> not found: <list of
application properties>
```

Probable cause

- Property defined in [Domain.XXX] section is not found in properties list defined by installed application.

Solution

- Delete the properties mentioned in the error message (or define them in the installed application).

Error message (7)

```
New configuration was not applied due to the following warnings:
Warning - Cannot remove domain <domain name> with <num of SCEs> SCEs. Note
that all subscribers will be removed from domain db.
Please use '--ignore-warnings' option to complete the operation.
```

Probable cause

- This is actually a warning: its purpose is to warn that the user removed from the p3sm.cfg file a domain that contained SCEs (which probably contained subscribers, etc.), and that the user will lose all subscriber data relevant for that domain. This warning appears only after the CLU command `p3sm ––load-config` is activated.

Solution

- To avoid this warning, use the `—-ignore-warnings` option.

RPC Errors

Error message

```
New configuration was not applied due to the following warnings:
Warning - RPC configuration was changed. Note: Reloading may take up to 5
seconds.
Please use '--ignore-warnings' option to complete the operation.
```

Probable cause

- This is actually a warning: it is displayed after the CLU command `p3sm –load-config` is activated when the RPC configuration in the p3sm.cfg file has been changed.

Solution

- Use the `—-ignore-warnings` option to complete the operation.

Radius Listener Errors

Error message

```
Duplicate NAS identifier <nasID> found in section [NAS name] : already
exists in <other NAS name>
```

Probable cause

- <nasID> is not unique.

Solution

- Change one of the <nasID> so that both are unique.

Common Validation Errors

The following configuration errors are relevant for all sections/parameters of the *p3sm.cfg* file.

Error message (1)

```
Unknown property <property name> found in section [<section name>] in
configuration file <file name>.
```

Probable cause

- Property written in the *p3sm.cfg* file is unknown to the SM. Maybe the name is misspelled or the property belongs in a different section.

Solution

- Ensure that the name is spelled correctly and that the property resides in the correct section.

Error message (2)

```
Unknown section [<section name>] found in configuration file {2}.
```

Probable cause

- The section written in the p3sm.cfg file is unknown to the SM. Maybe the name is misspelled.

Solution

- Ensure that the section name is spelled correctly.

Error message (3)

```
Error value <value> for property <property name> in section [<section
name>]. Optional values: [<values range>]
```

Probable cause

- Value of the property is invalid. The <values range> field contains the valid values.

Solution

- Specify any valid value for the property.

Error message (4)

```
Missing mandatory property <property name> in section [<section name>].
```

Probable cause

- The property <property name> is **mandatory** and must appear in the section <section name>.

Solution

- Set a value for the requested property in the specified section.

Error message (5)

```
Error value <property value> for property <property name> in [<section
name>] section. Valid format: [0..255].[0..255].[0..255].[0..255]
```

Probable cause

- The value is an invalid IP address.

Solution

- Specify a valid IP address.

Error message (6)

```
Error empty value for <property name> property in [<section name>] section -
must have at least one character
```

Probable cause

- Value of the <property name> is empty; for example, `prop=`

Solution

- Specify a non-empty value for the property.

Error message (7)

```
Section <section name> added when already exists
```

Probable cause

- The section with <section name> appears more than once. This error is most likely to occur for the [SE.XXX] and [Domain.XXX] sections.

Solution

- Use the specified section name only once.

APPENDIX E

Veritas Cluster Server Requirements and Configuration

This appendix provides basic guidelines for the Veritas Cluster Server (hereafter VCS) configuration in an SM cluster installation. It assumes basic knowledge of the VCS environment; it does not come in place of the VCS user guide. This appendix does not cover installation of the cluster or of SM's cluster agents.

This appendix lists the software and hardware system requirements for the VCS. It also gives a step-by-step explanation how to configure the SM cluster using the VCS configuration tools. Most of the examples are taken from the use of the Java Veritas Manager GUI, although the operations can also be done via the Veritas command-line utilities.

SM 2.5 supports Veritas Cluster Server version 3.5 on Solaris machines and Veritas Cluster Server version 2.2 on Linux machines.

Veritas Cluster Server System Requirements

For your convenience, the following Veritas Cluster Server System Requirements have been taken from the Veritas site:
`http://www.veritas.com`

- Supported Platforms:

  Sun Solaris 8, 9

  Red Hat Linux 3.0

- Networking:

  Public Network: 10 MB/100 MB/Gigabit Ethernet

  Private Network: 10 MB/100 MB/Gigabit Ethernet

- Ethernet Controllers:

  Requires at least three independent Ethernet connections per system

- Memory:

  Each VERITAS Cluster Server system requires at least 128 MB of RAM (256 MB of RAM is recommended)

- Supported Server Hardware:

Please refer to http://support.veritas.com or contact your VERITAS sales representative for the latest list of certified server hardware.

  - Sun Solaris 8, 9
  - Red Hat Linux 2.1, 3.0

- Supported Storage Hardware:

  Please refer to http://support.veritas.com or contact your VERITAS sales representative for the latest list of certified storage hardware.

  - Sun
  - Red Hat Linux

Veritas Cluster Server Configuration Guidelines

The sections below assume that the following operations were performed before starting the VCS configuration:

- Installation of the VCS on both machines. As part of this installation:
  - Each machine was given a hostname, which is used as the *system* name for the VCS configuration.
  - The machines are configured to recognize each other’s hostname.
- An IP address was allocated for the cluster (hereafter, *the cluster’s IP*).
- The SM and the TimesTen database were installed on both machines.
- The SM’s VCS agents were installed on each machine.
- The VCS manager Java console was installed on the administrator PC.

Note that in an SM cluster, two SM machines are connected to each other in a fully redundant way. The connection uses four cables: two for the VCS heartbeat mechanism, and two for the TimesTen database replication mechanism. Each machine is connected to the network via one of two redundant NICs. To access the cluster, you should use the *cluster IP address*, which is a virtual IP managed by the VCS. For management operations, you should use the local IP address of each machine.

SM Cluster Configuration

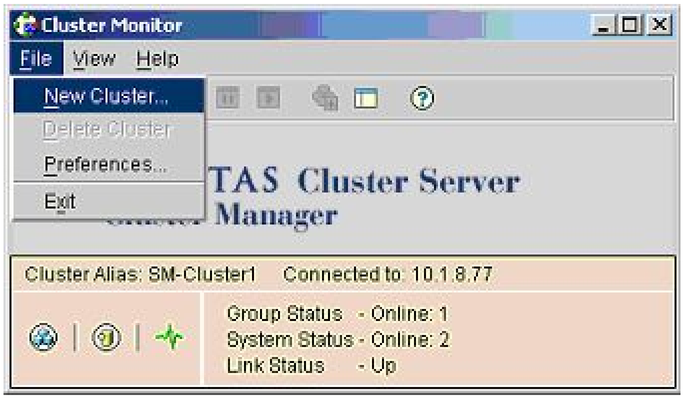
To configure the VCS with the SM cluster resources, perform the procedures described in the following sections

Adding Clusters

To add a cluster:

Step 1 Open the VCS cluster manager Java console by using START | PROGRAMS | VERITAS CLUSTER MANAGER | CLUSTER MANAGER (JAVA CONSOLE).

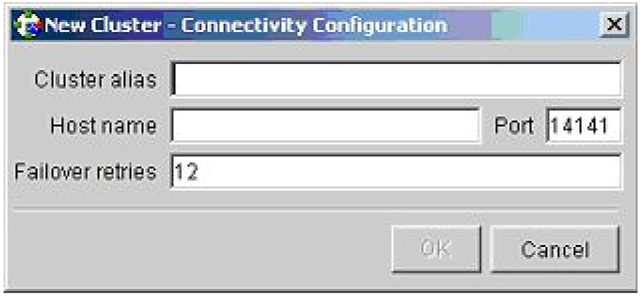
Step 2 Add a new cluster by selecting from the menu FILE | NEW CLUSTER.



Cluster Alias: SM-Cluster1 Connected to: 10.1.8.77

Group Status - Online: 1
System Status - Online: 2
Link Status - Up

Step 3 Configure the cluster (see following figure).



Cluster alias:
Host name: Port: 14141
Failover retries: 12

- Cluster Alias: cluster name.
- Host Name: one of the machine’s IP addresses or hostname.

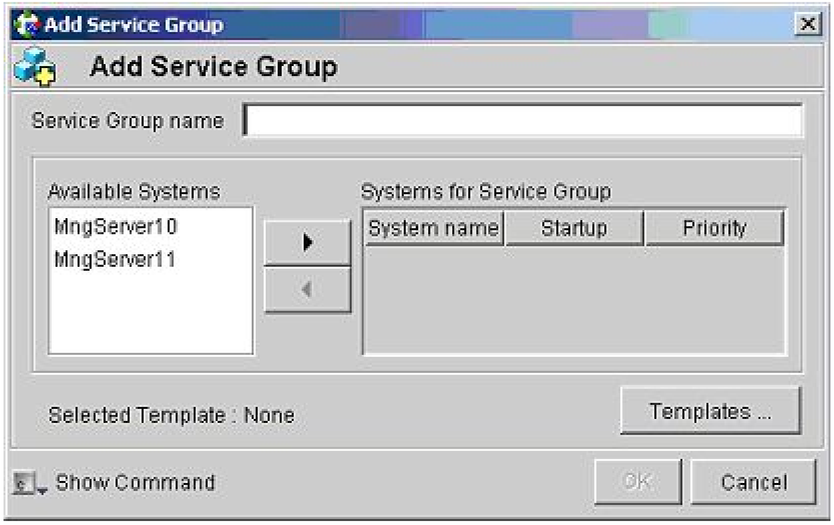
Step 4 Log in to the cluster.

Adding Services Groups

To add a service group to the cluster:

Step 1 In the cluster explorer, in the service group tab, right-click the cluster, and select ADD SERVICE GROUP.

Step 2 The following window appears:



Step 3 Enter a name for the service group.

Step 4 Add the two machines as part of the service group, and define their priority in the cluster.

Step 5 Click OK.

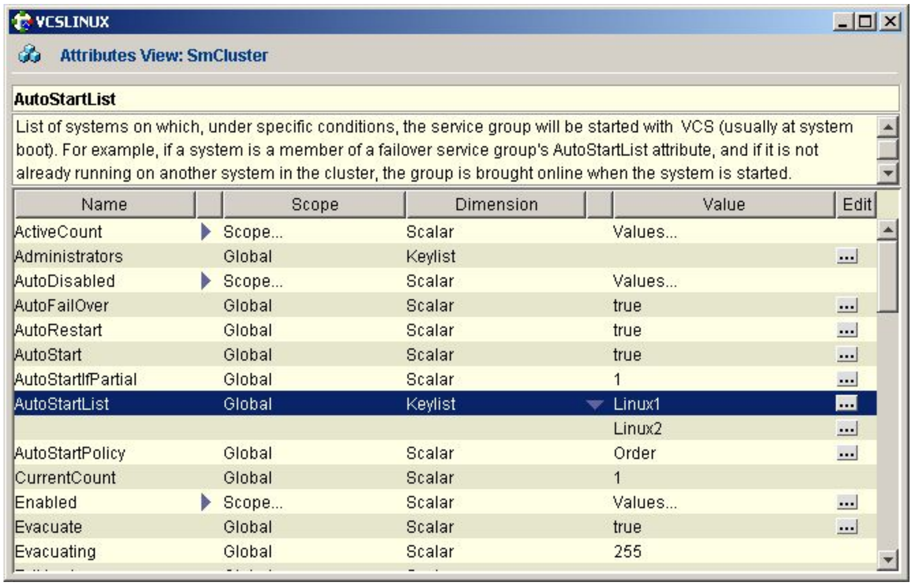
Setting the Auto-start

This section describes how to set the auto-start parameters that define which machine will start after a boot of both nodes. If these parameters are not set, then at boot of both nodes the cluster will stay offline.

To set the auto-start parameters:

Step 1 From the service group display, click SHOW ALL ATTRIBUTES.

Step 2 Make sure that both nodes are defined in the AutoStartList (see the following figure).



AutoStartList

List of systems on which, under specific conditions, the service group will be started with VCS (usually at system boot). For example, if a system is a member of a failover service group's AutoStartList attribute, and if it is not already running on another system in the cluster, the group is brought online when the system is started.

Name		Scope	Dimension		Value	Edit
ActiveCount	▶	Scope...	Scalar		Values...	
Administrators		Global	Keylist			...
AutoDisabled	▶	Scope...	Scalar		Values...	
AutoFailOver		Global	Scalar		true	...
AutoRestart		Global	Scalar		true	...
AutoStart		Global	Scalar		true	...
AutoStartIfPartial		Global	Scalar		1	...
AutoStartList		Global	Keylist	▼	Linux1	...
					Linux2	...
AutoStartPolicy		Global	Scalar		Order	...
CurrentCount		Global	Scalar		1	
Enabled	▶	Scope...	Scalar		Values...	...
Evacuate		Global	Scalar		true	...
Evacuating		Global	Scalar		255	

Step 3 Specify which node will start by defining the *AutoStartPolicy* parameter; select one of the following:

- Priority: the node with higher priority (as defined in SystemList).
- Order: the node that comes up first (if the two nodes are rebooted together, the former standby SM is usually the one that comes up first).
- Load: the node that was active before the reboot.

Step 4 Make sure that the *AutoStart* parameter is set to `true` (if `false`, both nodes will come up as standby).

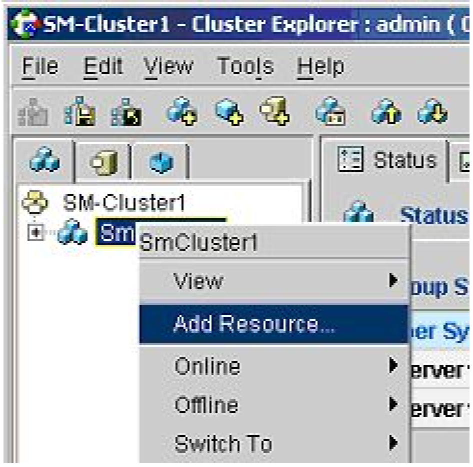
Adding SM Cluster Resources

This section describes how to add the various SM cluster resources.

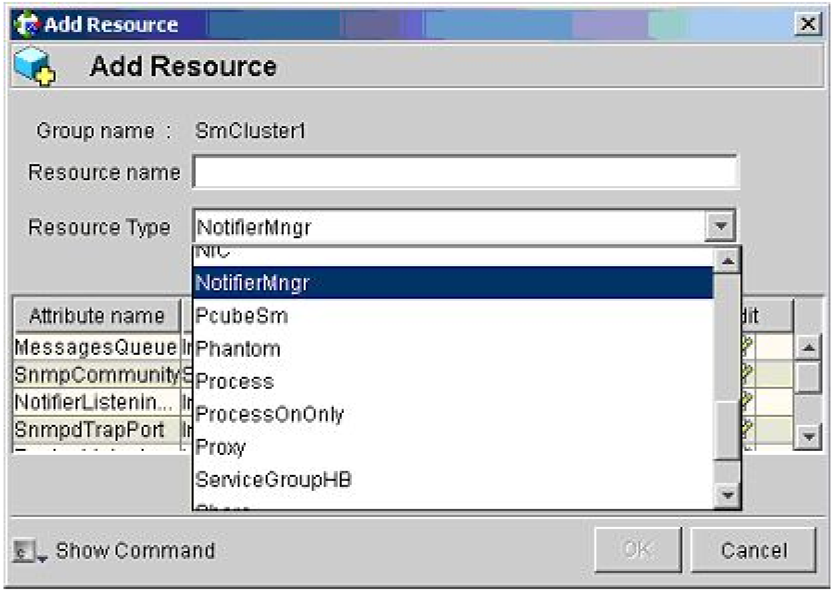
Adding Resources - General Guidelines

To add a resource (general guidelines):

Step 1 From the right-click menu of the service group, select ADD RESOURCE.



Step 2 When the following screen is displayed, select the RESOURCE TYPE, and give the resource a name.



Group name : SmCluster1

Resource name

Resource Type NotifierMngr

Step 3 Scroll through the attributes, and configure the ones that you need.

Step 4 When you are done, click OK.

Adding Network NICs

To add a Network NIC:

Step 1 Decide which are the two network interfaces that will be used for the network connection.

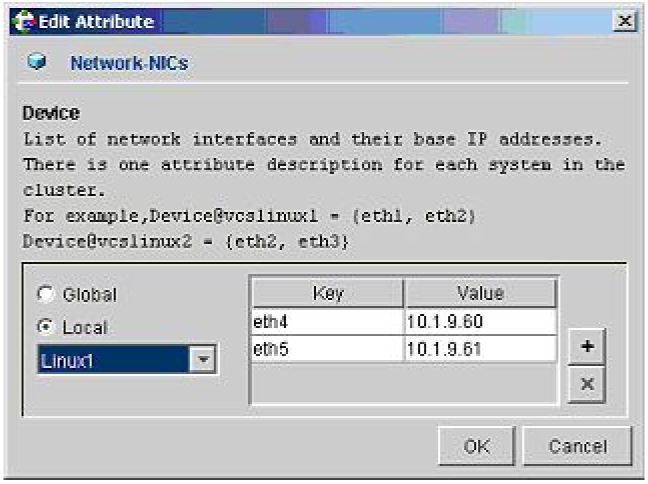
Step 2 Add a MultiNICA resource called *Network-NICs* to the service group.

Step 3 Define the following parameter:

- Device: Write the names of the Network NICs in the KEY column and their corresponding IP addresses in the VALUE column.

  Note: Use the LOCAL option and configure each machine separately, because the IP addresses are different in each machine.

  In the following example, `eth4` and `eth5` are the network NICs.



Device

List of network interfaces and their base IP addresses. There is one attribute description for each system in the cluster.

For example,Device@vcslinux1 = {eth1, eth2}

Device@vcslinux2 = {eth2, eth3}

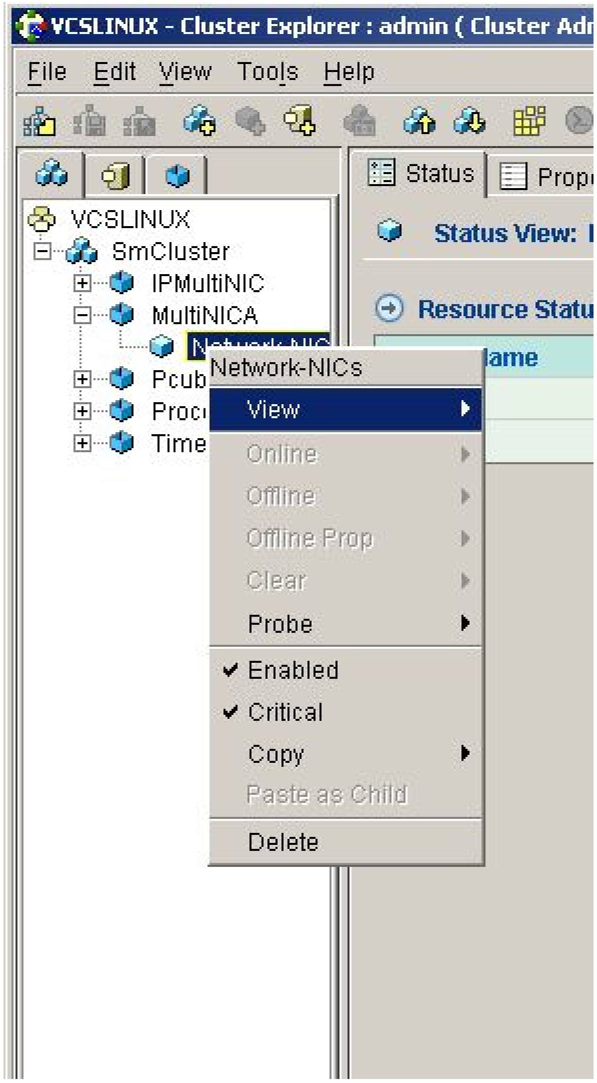
○ Global

● Local

Linux1

Key	Value
eth4	10.1.9.60
eth5	10.1.9.61

Step 4 Check the ENABLED and CRITICAL attributes of the resource (see the following figure):



Adding Network VIPs

To add a Network VIP:

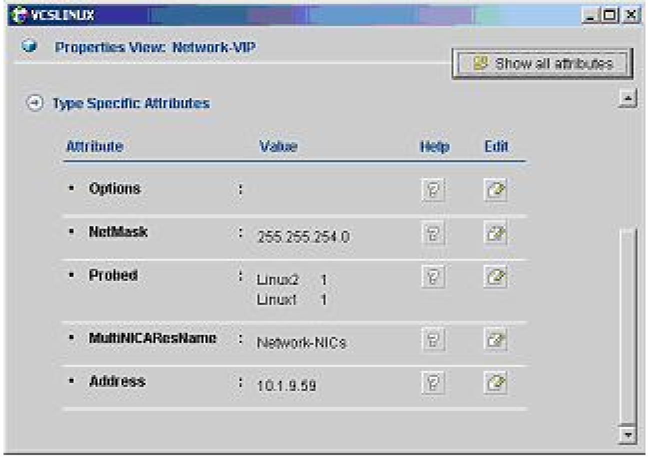
Step 1 Decide on the IP address of the cluster.

Step 2 Add to the service group an IPMultiNIC resource called *Network-VIP*.

Step 3 Define the following parameters:

- Address: Type the Cluster IP address.
- Net-mask: Type the network-mask you want to use for this IP.

- MultiNICAResName: Type *Network-NICs* to specify the relevant NICs.



Attribute		Value	Help	Edit
Options	:			
NetMask	:	255.255.254.0		
Probed	:	Linux2 1 Linux1 1		
MultiNICAResName	:	Network-NICs		
Address	:	10.1.9.59		

Step 4 Set the resource to be ENABLED and CRITICAL (see *Adding Network NICs* (on page E-7)).

Adding SM Resources

To add an SM resource:

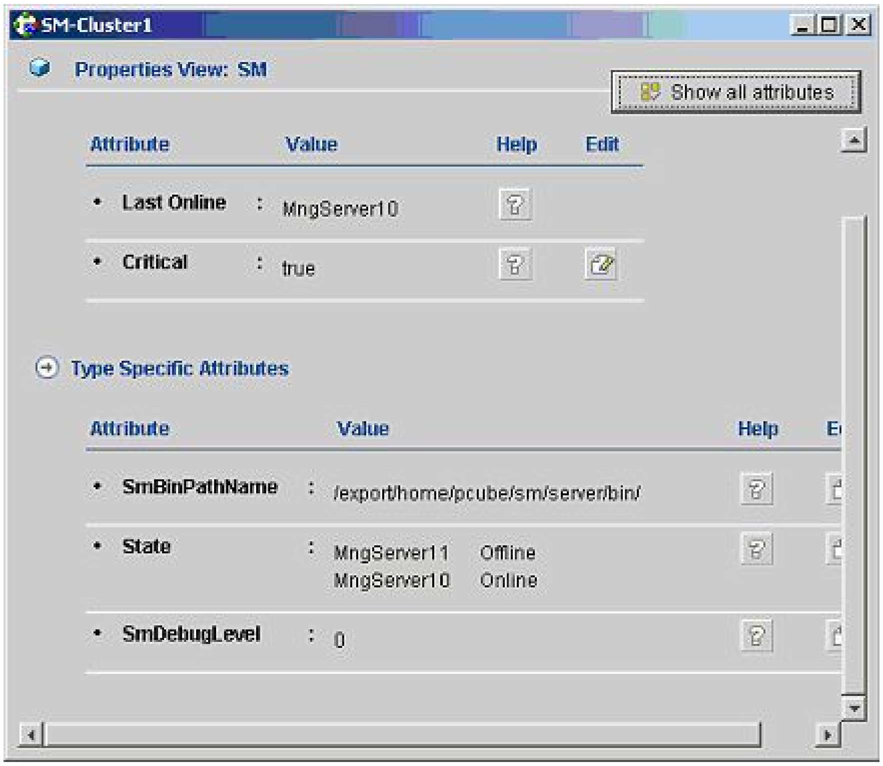
Step 1 Import the PcubeSm agent’s type from file */opt/VRTSvcs/bin/PcubeSm/PcubeSm.cf*.

Step 2 Add to the service group a PcubeSm resource called *SM*.

Step 3 Define the following parameters:

- SmBinPathName: Type the path to the *bin* directory under the SM installation directory; for example, */opt/pcube/sm/server/bin/*.

- SmDebugLevel: Type 1 to view debug messages, type 0 to disable debug messages.



Attribute	Value	Help	Edit
Last Online	MngServer10	?	
Critical	true	?	✎

Attribute	Value	Help	E
SmBinPathName	/export/home/pcube/sm/server/bin/	?	
State	MngServer11 Offline MngServer10 Online	?	
SmDebugLevel	0	?	

Adding TimesTen Daemon Resources

To add a TimesTen Daemon Resource:

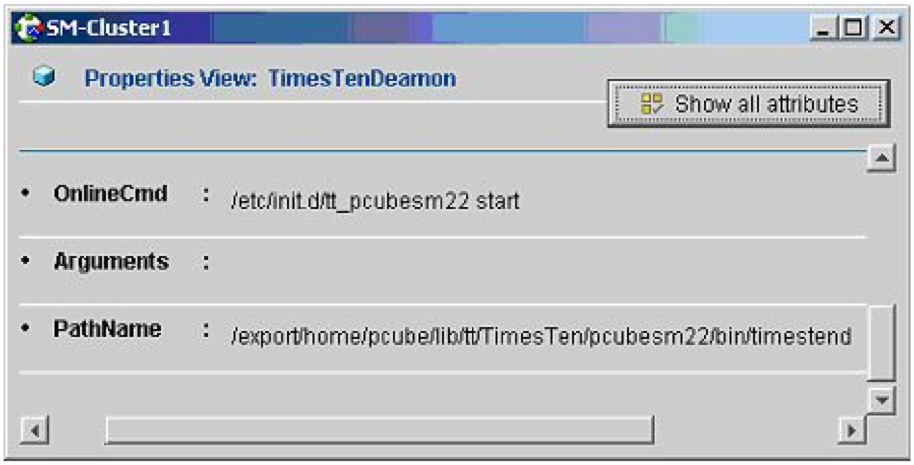
Step 1 Import the ProcessOnOnly agent's type from file `/opt/VRTSvcs/bin/ProcessOnOnly/ProcessOnOnly.cf`.

Step 2 Add to the service group a ProcessOnOnly resource called "*TimesTenDaemon*".

Step 3 Define the following parameters (see following figure):

- OnlineCmd: Type the TimesTen Daemon start command: `/etc/init.d/tt_pcubesm22 start`.

- PathName: Type the TimesTen Daemon process path; for example, `/opt/pcube/lib/tt/TimesTen/pcubesm22/bin/timestend`.



- OnlineCmd : /etc/init.d/tt_pcubesm22 start
- Arguments :
- PathName : /export/home/pcube/lib/tt/TimesTen/pcubesm22/bin/timestend

Adding TimesTen Replication Agent Resources

To add a TimesTen replication agent resource:

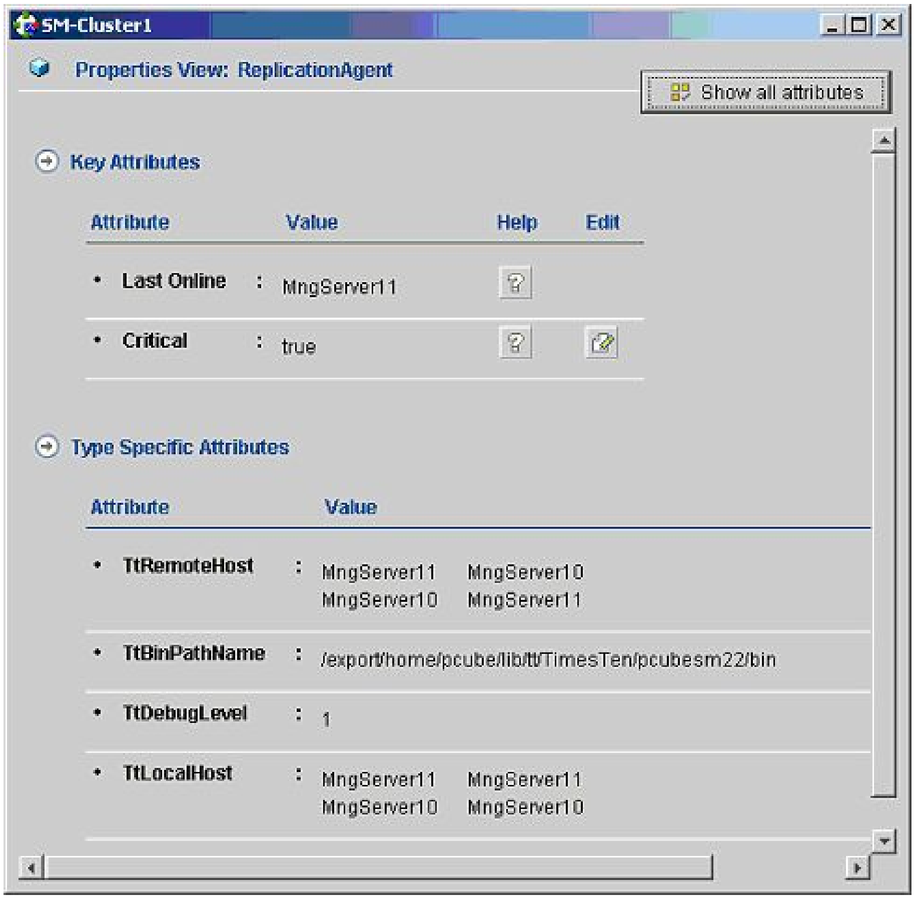
Step 1 Import the TimesTenRep agent's type from file `/opt/VRTSvcs/bin/TimesTenRep/TimesTenRep.cf`.

Step 2 Add to the service group a TimesTenRep resource called *ReplicationAgent*.

Step 3 Define the following parameters (see following figure):

- TtRemoteHost: For each system, type the hostname of the other system. For additional information, see *Replication Network Configuration* (on page E-12).
- TtLocalHost: For each system, type the hostname of this system. For additional information, see *Replication Network Configuration* (on page E-12).
- TtBinPathName: Type the TimesTen bin directory path; for example, `/opt/pcube/lib/tt/TimesTen/pcubesm22/bin`.

- TtDebugLevel: Type 1 for viewing debug messages, enter 0 for disabling the debug messages.



Key Attributes

Attribute		Value	Help	Edit
• Last Online	:	MngServer11	?	
• Critical	:	true	?	✎

Type Specific Attributes

Attribute		Value
• TtRemoteHost	:	MngServer11 MngServer10 MngServer10 MngServer11
• TtBinPathName	:	/export/home/pcube/lib/tt/TimesTen/pcubesm22/bin
• TtDebugLevel	:	1
• TtLocalHost	:	MngServer11 MngServer11 MngServer10 MngServer10

Replication Network Configuration

The configuration of the replication private network between the two SMs of the cluster must be carefully planning. This section discusses some guidelines for doing the configuration.

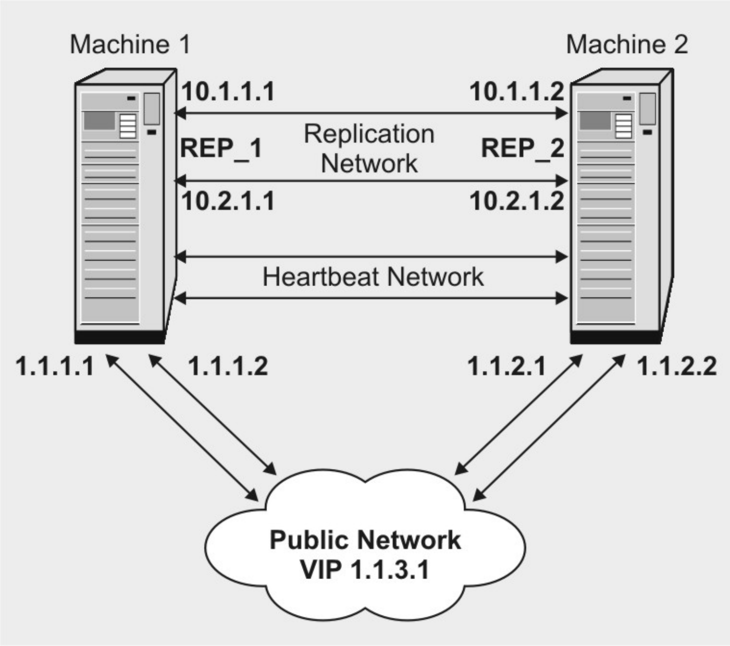
The TimesTen replication agent uses hostnames to implement fail-over between the two replication NICs. The agent uses the first IP address of the hostname supplied to the agent to connect to the other agent. If the connection fails and cannot be reconstructed on the first IP, the replication agent tries the next IP addresses assigned to this hostname, and so on.

The assignment of hostnames to IP addresses should be done by editing the `/etc/hosts` file.

Note Reboot of the machine is required for the configuration to take place.

EXAMPLE:

The following figure shows an example of a replication network.



To configure the replication network shown by the above figure, do the following:

- Configure the IP addresses of each of the replication NICs, every one in a different network. In this example, the IP addresses of the Machine1 replication NICs are 10.1.1.1 and 10.2.1.1.
- Assign a hostname to both of the local replication NIC IP addresses. (This hostname is supplied to the VCS replication agent as the local hostname.) In this example, the hostname REP_1 is assigned to the IP addresses of the replication NICs. In the `/etc/hosts` file, be sure to assign the local hostname (Machine1) to the local replication NICs too. Make sure there are **no empty lines** between the lines containing the local hostname.
- Assign a hostname to both of the remote replication NIC IP addresses. (This hostname is supplied to the VCS replication agent as the remote hostname.) In this example, the hostname REP_2 is assigned to the IP addresses of the replication NICs of Machine2.
- The `/etc/hosts` file on Machine1 should look as follows:

```
127.0.0.1   localhost
1.1.1.1     Machine1    loghost
10.1.1.1    Machine1    REP_1    REP_1_NIC_1
10.1.2.1    Machine1    REP_1    REP_1_NIC_2
10.1.1.2    REP_2       REP_2_NIC_1
10.1.2.2    REP_2       REP_2_NIC_2
```

- On Machine1, use the following hostnames as the Replication agent parameters:
  - REP_1 as parameter TtLocalHost
  - REP_2 as parameter TtRemoteHost

Linking the Resources

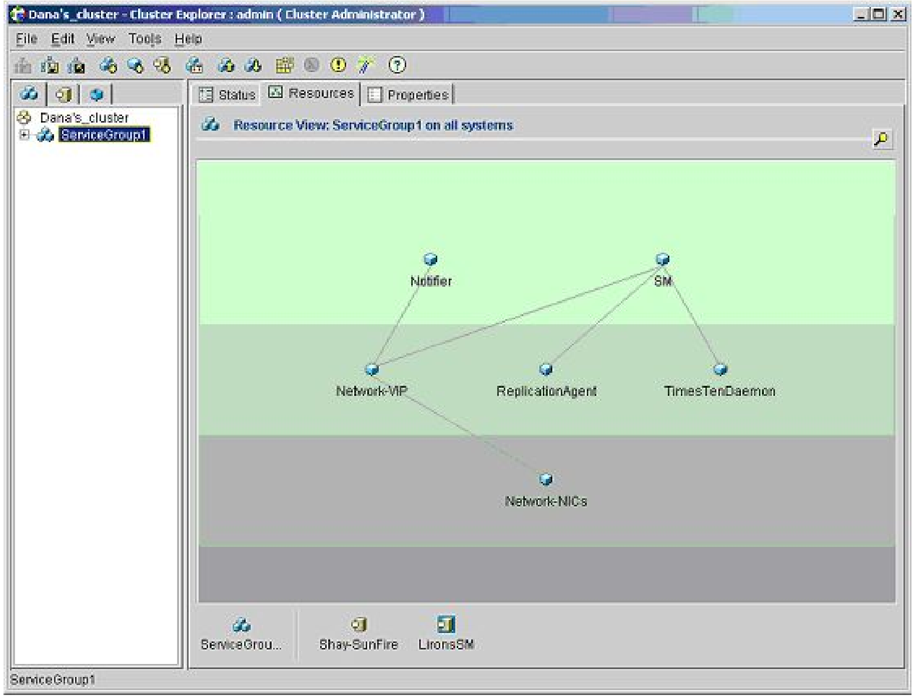
Linking the resources defines the order of becoming online and going offline.

To link the resources:

Step 1 Select the service group and enter the Resources tab.

Step 2 To link two resources, click once on one resource, pull the line to the second resource, and click once over the icon of the second resource.

The final links should look like the ones in the following figure:



Saving and Closing Your Cluster Configuration

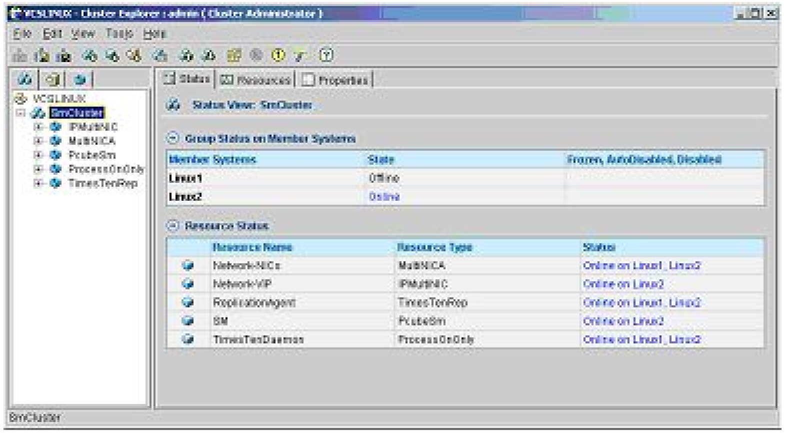
Remember to save your work at the end of the configuration process (see *Saving the Configuration* (on page E-15) and *Closing the Configuration* (on page E-15)).

Verifying That Service Group is Online

Check that all of the resources are online/offline, according to the system and the resource type.

Note The TimesTen Daemon, the NICs, and the TimesTen Replication Agent should all be online on all of the systems.

The state should be similar to the following figure:



Status View: SmCluster

Group Status on Member Systems

Member Systems	State	Frozen, AutoDisabled, Disabled
Linux1	Offline	
Linux2	Online	

Resource Status

	Resource Name	Resource Type	Status
	Network-NICs	MultiNICA	Online on Linux1, Linux2
	Network-VIP	IPMultiNIC	Online on Linux2
	ReplicationAgent	TimesTenRep	Online on Linux1, Linux2
	SM	PcubeSm	Online on Linux2
	TimesTenDaemon	ProcessOnOnly	Online on Linux1, Linux2

Useful Operations

Following are some useful operations for management of the VCS.

Logging into the Cluster

After you have added the cluster, click the icon: a login window appears.

Use the initial user and password (user: *admin*; password: *password*) for logging in.

Saving the Configuration

Click the icon, or from the menu select FILE | SAVE CONFIGURATION.

Before exiting the VCS make sure your configuration is saved ; otherwise, your configuration will be lost.

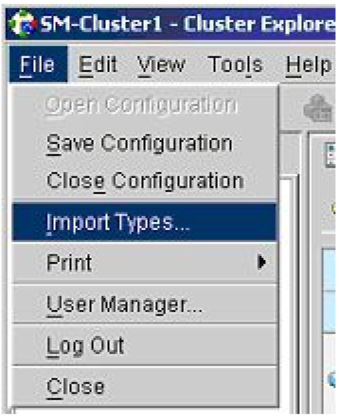
Closing the Configuration

Click the icon, or from the menu select FILE | CLOSE CONFIGURATION.

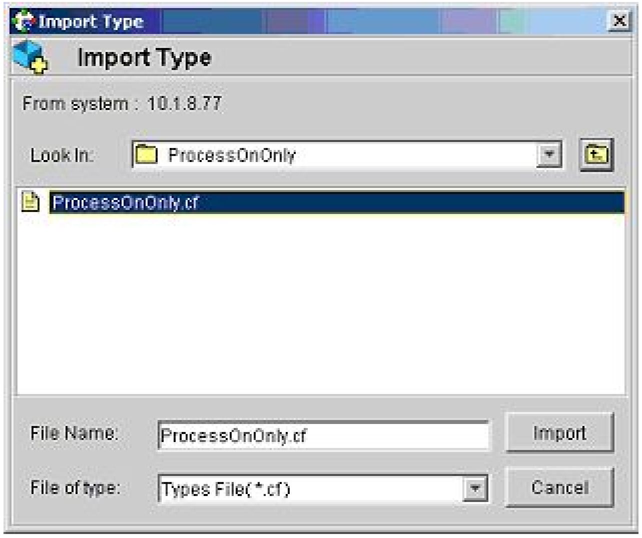
Before exiting the VCS, make sure your configuration is closed. Some operations (like rebooting the system) could fail or cause a configuration conflict if performed while the configuration is in read/write mode.

Importing Types

To be able to configure the SM Veritas agents, you first have to import the type file of these agents. This operation is performed by selecting from the menu FILE | IMPORT TYPES. A navigation window appears:



The window navigates through one of the cluster-system's file system. Go to the agent directory under `/opt/VRTSvcs/bin/<agent-dir>`, where there is a file whose name extension is `.cf`. Select this file (see following figure).



From system : 10.1.8.77

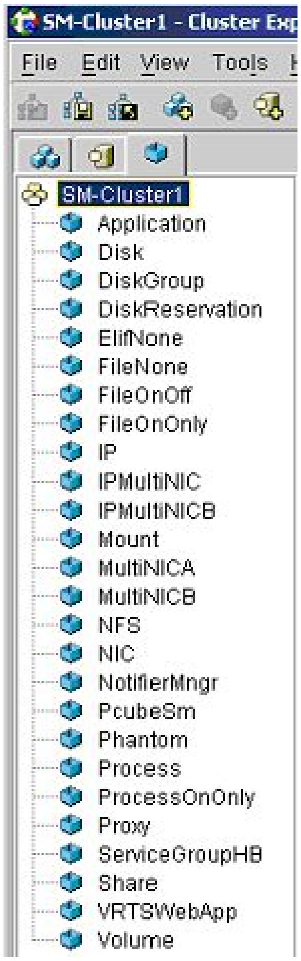
Look In: ProcessOnOnly

ProcessOnOnly.cf

File Name: ProcessOnOnly.cf

File of type: Types File(*.cf)

You can see the resource's parameters in the following window:



SM-Cluster1
- Application
- Disk
- DiskGroup
- DiskReservation
- ElifNone
- FileNone
- FileOnOff
- FileOnOnly
- IP
- IPMultiNIC
- IPMultiNICB
- Mount
- MultiNICA
- MultiNICB
- NFS
- NIC
- NotifierMngr
- PcubeSm
- Phantom
- Process
- ProcessOnOnly
- Proxy
- ServiceGroupHB
- Share
- VRTSWebApp
- Volume

SNMP Support

VCS provides a method for notifying the user of important events such as a resource or system fault. For this purpose, VCS supplies a NotifierMngr agent that enables the reception of messages from VCS and the delivery of those messages to SNMP consoles. This section describes configuring NotifierMngr in order to enable SNMP support.

Configuring NotifierMngr

NotifierMngr can be added and configured using either the command line or the Cluster Manager Java Console.

When started from the command line, Notifier is a process that VCS does not control.

For best results, therefore, use the NotifierMngr agent bundled with VCS to configure Notifier as part of a highly available service group, which can then be monitored, brought online, and taken offline.

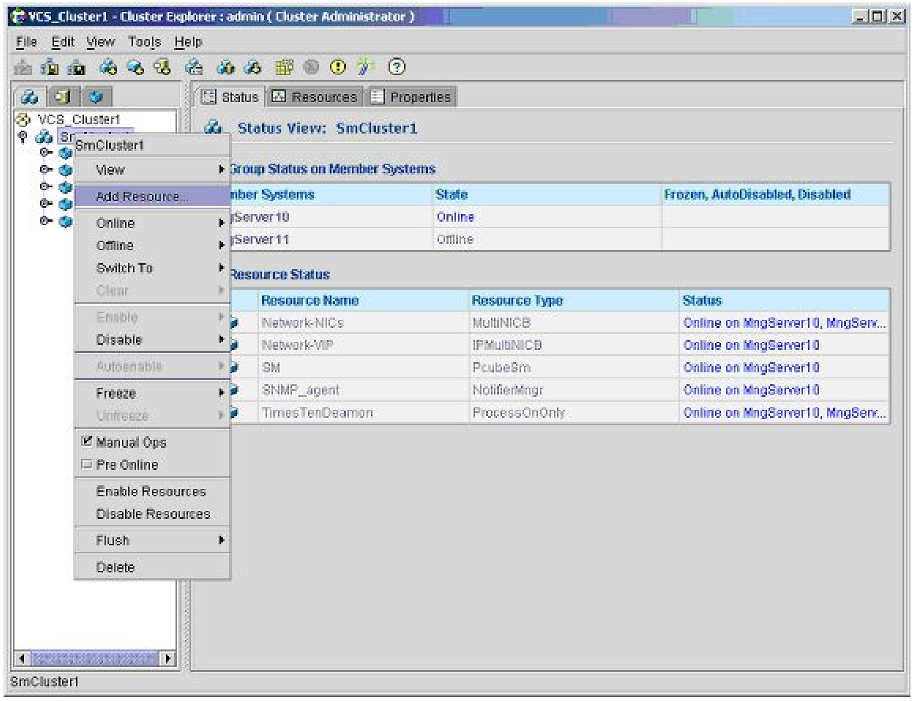
The following sections describe the configuration process using the Cluster Manager Java Console.

Adding NotifierMngr Resource

NotifierMngr should be added as a resource to the SM Cluster service group.

To add the NotifierMngr Resource:

Step 1 Add to the service group a NotifierMngr resource called Notifier.



Status View: SmCluster1

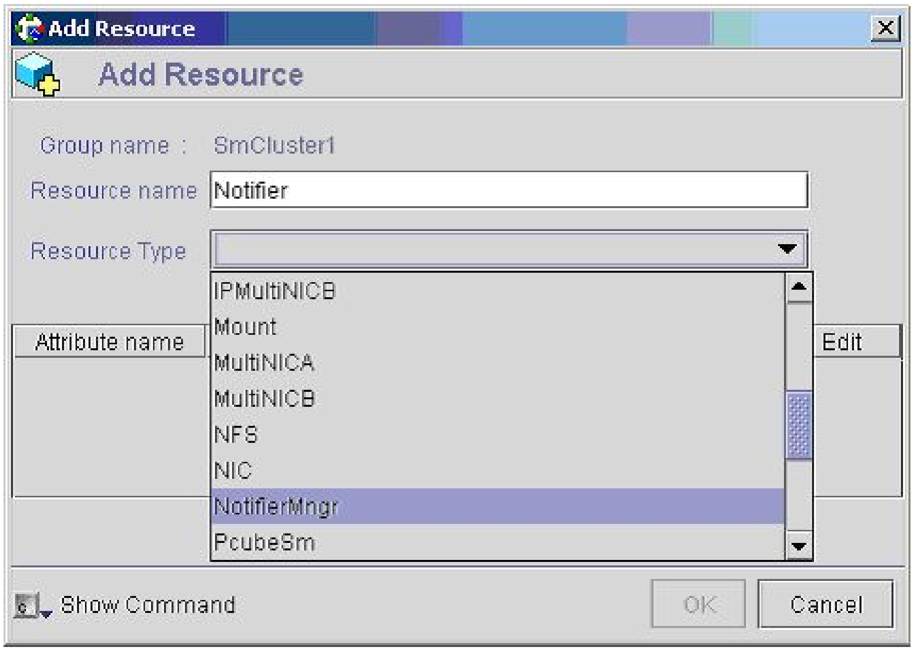
Group Status on Member Systems

nber Systems	State	Frozen, AutoDisabled, Disabled
Server10	Online	
Server11	Offline	

Resource Status

Resource Name	Resource Type	Status
Network-NICs	MultiNICB	Online on MngServer10, MngServ...
Network-VIP	IPMultiNICB	Online on MngServer10
SM	PcubeSm	Online on MngServer10
SNMP_agent	NotifierMngr	Online on MngServer10
TimesTenDeamon	ProcessOnOnly	Online on MngServer10, MngServ...

Step 2 When the following screen is displayed, choose NotifierMngr as a ResourceType.



Group name : SmCluster1

Resource name: Notifier

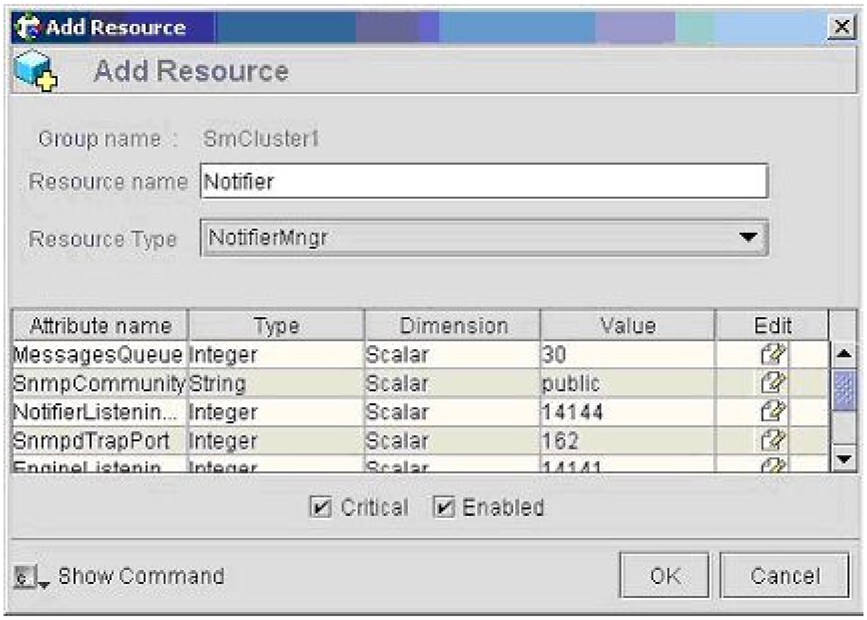
Resource Type: NotifierMngr

Configuring the NotifierMngr Attributes

After the NotifierMngr resource is added its attributes should be configured.

To configure the NotifierMngr attributes:

Step 1 Select NotifierMngr as a Resource Type; the following screen appears:



Group name : SmCluster1

Resource name Notifier

Resource Type NotifierMngr

Attribute name	Type	Dimension	Value	Edit
MessagesQueue	Integer	Scalar	30	✎
SnmpCommunity	String	Scalar	public	✎
NotifierListenin...	Integer	Scalar	14144	✎
SnmpdTrapPort	Integer	Scalar	162	✎
EngineListenin	Integer	Scalar	14141	✎

☑ Critical ☑ Enabled

Figure E-3: Configuring NotifierMngr Attributes

Step 2 Define the following parameters (see above figure):

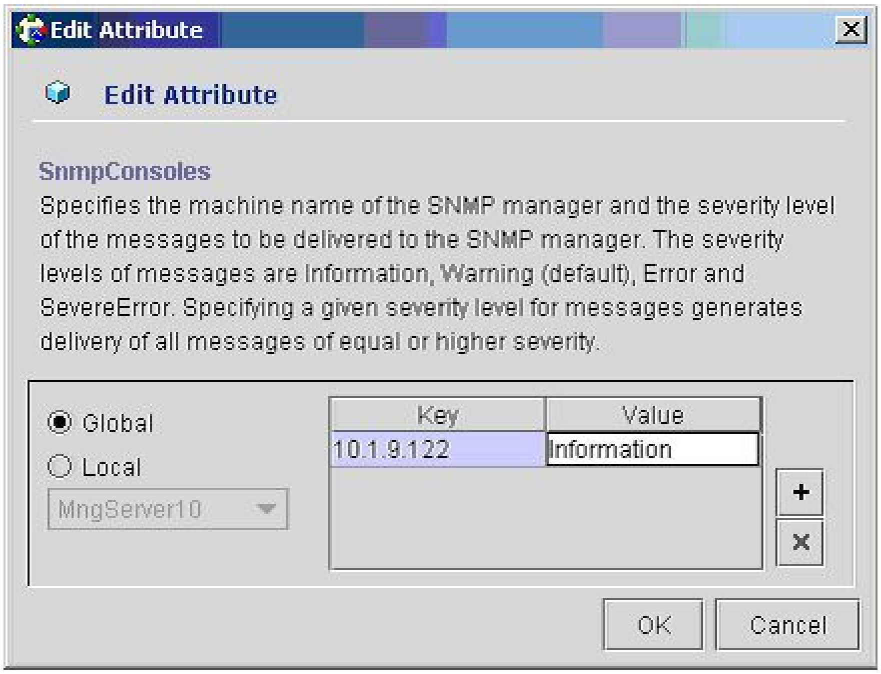
SnmpConsoles: Specify the machine name of the SNMP manager and the severity level of messages to be delivered to the SNMP manager. The severity levels of messages are Information, Warning, Error, and SevereError. Specifying a given severity level for messages generates delivery of all messages of equal or higher severity.

SnmpdTrapPort: Specify the port to which the SNMP traps are sent. The value specified for this attribute is used for all consoles if more than one SNMP console is specified. The default is `162`.

SnmpCommunity: Specify the community ID (a string scalar) for the SNMP manager. The default is `public`.

Configuring the SnmpConsole Attribute

The SnmpConsole attribute specifies the IP addresses that you want the SNMP traps to be sent to. You can specify different trap severity for each IP address:



Edit Attribute

SnmpConsoles

Specifies the machine name of the SNMP manager and the severity level of the messages to be delivered to the SNMP manager. The severity levels of messages are Information, Warning (default), Error and SevereError. Specifying a given severity level for messages generates delivery of all messages of equal or higher severity.

● Global
○ Local
MngServer10

Key	Value
10.1.9.122	Information

Linking to IPMultiNIC

To link to IPMultiNIC:

Step 1 Using the resources viewer, connect the *Notifier* to the *Network-VIP* resource so that it will be online after the VIP.

For more information, see the figure in *Adding TimesTen Replication Agent Resources* (on page E-11).

Viewing Traps

After NotifierMngr is added and configured, it will send traps according to the configured severity to the destinations configured by the SnmpConsole Attribute.

These traps can be viewed using SNMP trap viewer/MIB Browser (for example, AdventNet MibBrowser).

For a complete list of traps/severities, please see Chapter 10 of the VERITAS Cluster Server User Guide.

Glossary of Terms

A

Active subscriber

An online subscriber who is actually generating IP traffic.

Anonymous subscriber mode

A mode in which entities defined as an IP address(es) are treated as subscribers. The correlation to actual subscriber IDs is not performed by the system, but can be done offline externally.

Automatic integration

Subscribers are introduced into the Subscriber Manager (SM) automatically upon login, by the LEG.

B

Bump-in-the-wire topology

The SCE Platform physically resides on the data link between the subscriber side and the network side.

This topology can both monitor and control traffic.

C

Cable support module

An SM module for cable environment. When working in a mode of CPE as subscriber, this module functions as a translator between the cable world and the SM.

Cisco Application

An SML program that determines how the SCE Platform operates.

D

Data Collector (DC)

A software application that is responsible for collecting RDRs from the SCE Platforms, processing them, and preparing them for reports.

Domains

A group of SCE Platforms that share a group of subscribers. The subscriber traffic can pass through any SCE Platform in the domain.

A subscriber and an SCE can belong to only one domain.

E

External splitter topology (SCE1000 only)

The SCE1000 does not reside physically on the data link. Data is forwarded to the SCE1000 via an optical splitter, which splits the traffic on the link, sending all information to the SCE1000 in parallel with its transmission through the splitter. The optical splitter is connected physically on the Gigabit Ethernet link and only the receive inputs of the two GBE interfaces in the SCE1000 are connected to the splitter.

This topology permits monitoring of the traffic flow only.

I

Inline connection mode

The SCE Platform physically resides on the data link between the subscriber side and the network side, and can both receive and transmit traffic, permitting traffic control as well as monitoring.

L

Login Event Generators (LEGs)

Software components that use the SM API to generate subscriber-record update messages such as login/logout, and send them to the SM.

LEGs are usually installed on AAA/OSS platforms, and serve as translators from events generated by the customer system to Cisco subscriber update events.

M

Manual integration

Subscribers are introduced into the SM by importing them in the form of a `csv` file, or by using the appropriate SM CLU.

O

Online subscriber

A subscriber who is currently online (logged in). At any given time, a certain number of online subscribers will be idle.

P

PQI (Cisco Installation) File

An application package file that is installed on the SCE Platform or associated software modules.

Pull mode

Subscriber information is not distributes automatically. Only when an SCE Platform cannot associate the IP traffic with a subscriber, will it request (pull) information from the SM.

Push mode

The SM automatically distributes (pushes) subscriber information to the relevant SCE Platform(s) immediately after the record is added or changed. Therefore, when the subscriber starts producing traffic through the SCE Platform, it is ready with the required subscriber information.

Q

Quota

A (subscriber's) limit for a specific metric, such as bandwidth or volume.

R

RDR (Raw Data Record)

A data record produced by the SCE Platform that reports on events in the traffic. RDRs produced by the SCE Platform are sent to the Data Collector and then stored in the Data Collector database or forwarded to third-party systems. The RDR typically contains quota request or reports service usage.

Receive-only connection mode

The SCE Platform does not reside physically on the data link, and therefore can only receive data and not transmit.

This mode has traffic monitoring capability only.

S

SCE Platform

The Cisco Service Control purpose-built hardware service component. This hardware device, an SCE1000, or SCE2000, is capable of performing smart analysis of the packets at wire speed. It monitors the traffic on the line, producing raw data to be provided to the loaded application, which processes the data for functions such as reporting, policy management, subscription management, and implementation of tiered service subscriber aware traffic policies.

SLI (SML Loadable Image) File

An SLI file is a software package (part of an Engage solution) that contains the SML application that is loaded onto a SE Platform. The SML application determines the behavior of the SE Platform. Different SE Platforms can have different SML applications, even when they are within the same POP. (Operators do not need to access the SLI file.)

Subscriber

A Service Provider’s client. This can refer to either an individual (single IP address) or a company (range of IP addresses).

Subscriber aging

The automatic removal of a subscriber when no traffic sessions assigned to it have been detected for a defined amount of time

Subscriber aware mode

A mode in which actual subscribers are defined in the system, thus requiring no external correlation to actual subscriber IDs.

Index